

EN_Internet Booking Engine

API documentation

Document version	v2.3.85 2.3 - recommended API version 85 – sprint number where recent changes were introduced
Supported API versions	v2.2, v2.3
Deprecated versions	v1.1, v2.0, v2.1

Table of Contents

Change history	4
Glossary	27
General provision.....	28
General information about API methods	29
API Versions.....	29
General information about API v2.x methods	29
HTTP(S)-headers v2.x	29
IBE API services query	30
IBE API services response codes.....	30
Booking flow description.....	32
General booking flow	32
Description of partners services upsale process after payment	33
API methods description.....	34
Obtain session service.....	34
Obtain minimal prices service	36
Obtain flights service	44
Booking management service	70
Obtain partner's services information service: api/reservation/partners	168
PNR Tariff confirmation service	174
Service for obtaining a seatmap without PNR.....	179
Auxiliary services management service.....	188
Auxiliary partner's services management service	207
Retrieve aircraft seatmap service	217
Booking payment service.....	222
Authorise in CRM: api/user	237
Retrieve information about linked users service (CRM)	244
Promo code management service (CRM).....	247
Discount retrieving service	254
Itinerary receipt receive and send to email service.....	258
Voluntary tickets and services refund service	261

Involuntary refund of tickets service. Simplified scheme.	276
Change passenger's data service	282
Online check-in service.....	288
Cancellation service	293
Service for obtaining a catalog of services without PNR	297
Certificate management service	310
Service for getting information about the weather	317
Service for checking participation of passengers in the loyalty program	321
Services for working with queues (for ADP environment).	323
Service for void airtickets	335
Service for add remark to reservation /api/version/Reservation/AddRemark	337
Service for add SSR to reservation /api/version/Reservation/AddSSR	348
Service for prolong reservation.....	360
Service for retrieving SSR and remarks information /api/version/Reservation/Advanced	364
Service for removing segment from the reservation.....	375
Obtain tickets for rebook /api/version/Exchange/Search.....	377
Service of tickets rebook /api/version/Exchange	384
Service API methods description.....	394
Prolong Amadeus session service	394
Retrieve airports list service.....	396
IBE health check service.....	398
Send error report service.....	402
Get exchange rates service.....	404
Get Cyrillic transliteration rules service	407
Service for monitoring the status of the backup engine contour	410
Attachments	412
Attachment 1. IBE error codes.....	412
Attachment 2. Dictionaries	424
Attachment 3. Remarks/service SSR	447
Attachment 4. Partner's services.	453
Attachment X. Request and response examples	456

Change history

Date (dd.mm.yy)	Version	Changelog	Author
04.04.2022	2.3.85	<p>Added:</p> <ul style="list-style-type: none"> • POST api/Exchange/Search method for searching tickets for rebook; • POST api/Exchange method for proceeding rebook of tickets; • a new services DBAG is added to EN_Attachment 2. Dictionaries; • a new parameter currency is added methods get/catalogue and post/services and get/seatmap; • a new partner's service Business rooms is added to EN_Attachment 4. Partner's services; • pnrNumbersPassengerCount parameter for POST api/reservation/prolong method. <p>Changed:</p> <ul style="list-style-type: none"> • description of information about methods POST/GET api/annul. 	Savin A.A.

09.02.2022	2.3.84	<p>Added:</p> <ul style="list-style-type: none"> • POST api/InvoluntaryRefund/Calculate method to get information about involuntary refund; • autoRefundCompleted parameter to the POST api/InvoluntaryRefund response; • isInfantProduct parameter to the POST api/partners request; • isCancelledByAirline parameter to the GET api/reservation response; • issuedByThirdPartyOffice parameter to the GET api/reservation response; • isCompanyGroupReservation parameter to the GET api/reservation response (v2.3); • 9003, 9004 error codes (IBE) to EN_Attachment 1. IBE error codes; • 1007, 5006, 6007 error codes (services) to EN_Attachment 1. IBE error codes; • IssuedOffice field to the remark for partner's service to EN_Attachment 3. Remarks/service SSR; • Alfamedicine service added to EN_Attachment 4. Partner's services. <p>Changed:</p> <ul style="list-style-type: none"> • isGroupReservation parameter assignment in GET api/reservation response. 	Savin A.A.
------------	--------	--	------------

27.12.2021	2.3.83	<p>Added:</p> <ul style="list-style-type: none"> • POST api/certificate method to create certificate; • POST api/payment/applePay method to request payment with ApplePay token; • POST api/involuntaryRefund/moneyless method to process involuntary refund without money refund; • to requests and responses of GET/POST api/reservation, POST api/refund/calculate and POST api/refund/partial methods luggageSeats array parameter with information about luggage with separate air ticket; • to request POST api/reservation endorsements array parameter with additional information about documents to buy a ticket with subsidized fares or to apply discount on partner loyalty cards; • to PUT api/Passengers method lang, redirectLang, returnUrl parameters; • notes for GET reservation/Advanced, POST Reservation/AddSSR, POST Reservation/AddRemark methods; • in Attachment 2. Dictionaries: new offices. <p>Changed:</p> <ul style="list-style-type: none"> • description of the logic of changing passengers data and format and parameters of PUT api/Passenger response. 	Gudumak S. E.
------------	--------	---	------------------

13.12.2021	2.3.82	<p>Added:</p> <ul style="list-style-type: none"> • in Attachment 2. Dictionaries: new offices • to requests to add remarks and SSRs POST api/reservation/AddRemark /api/Reservation/AddSSR; • to get information about remarks and SSRs GET api/reservation/Advanced; • request params api/flights: cabin, SearchAllBrands • to responses api/catalogue and api/catalogueStandalone param onlyForRussianCitizens • to response payment/googlePay param ACS, mdOrder • parameters to requests PUT api/passenger,POST payment : callbackUrl, engine • annul segment method: POST api/reservation/CancelSegments; • request params POST api/reservation :accountCode, tourCode <p>Changed:</p> <ul style="list-style-type: none"> • in response payment/googlePay parameter returnUrl changed to redirectUrl 	Kulish K.V.
28.10.2021	2.3.81	<p>Added:</p> <ul style="list-style-type: none"> • in Attachment 2. Dictionaries: new office • in Attachment 2. Dictionaries: NYML meal service • to request and response of api/reservation method added "currency" parameter • to requests POST api/SeatmapStandalone и POST api/CatalogueStandalone added "currency" parameter • to requests POST api/SeatmapStandalone и POST api/CatalogueStandalone added "nationality" parameter 	Kulish K.V.

15.10.2021	2.3.80	<p>Added:</p> <ul style="list-style-type: none"> • POST api/payment/googlePay method for performing payment with GooglePay parameter; • to the response of api/reservation method: <ul style="list-style-type: none"> ○ information about taxes taxDetailsInformation; ○ information about statuses of flight coupons ticketCouponStatuses; ○ date and time of issuing tickets and services issueDate; • to GET api/flights request optional parameter of currency of recommendations; • to GET api/calendar request and response optional parameter of currency of recommendations; • in Attachment 1. IBE error codes: new error codes; • in Attachment 2. Dictionaries: new offices; • in Attachment 2. Dictionaries: BLTW service which isn't sold in IBE. <p>Changed:</p> <ul style="list-style-type: none"> • in Attachment 2. Dictionaries: types of passengers GCF, JCB and JNN is expanded for using in family trips. 	Gudumak S. E.
14.09.2021	2.3.79	<p>Added:</p> <ul style="list-style-type: none"> • in Attachment 2: dictionary "Types of Amadeus offices" • to GET api/reservation method added new parameter fareBasisCodesFlightRefs; • to requests GET api/flights , GET api/calendar added new parameter - meta; • added voiding ticket method DELETE api/ticket • added prolong method POST reservation/prolong • to GET api/reservation response added statuses parameter 	Kulish K.V.

18.08.2021	2.3.78	<p>Added:</p> <ul style="list-style-type: none"> • to the booking payment service was added a description of a new method GET api/payment/info to receive information that should be displayed on the payment page before making a payment; • service for checking the participation of passengers in the loyalty program POST api/services/searchContact; • to the response of the api/reservation method were added parameters of a sign of errors in passengers data hasPassengersDataErrors and array with detailed information about these errors passengerDataErrors; • services for working with queues on the ADP environment: POST api/queue/placePNR, POST api/queue/list, POST api/queue/counttotal, POST api/queue/removeItem; • in Attachment 2: dictionary "Types of Amadeus offices" - new offices. 	Gudumak S. E.
22.07.2021	2.3.77	<p>Added:</p> <ul style="list-style-type: none"> • to the response of the GET api/reservation method added new parameter checkInHasStarted; • to flight obtain service added description of new methods GET api/flights/allFlightStatusesForDay and POST api/flights/particularFlightsStatuses for obtain information about the flight schedule; • description of the new get/weather method for obtain information about the weather; • to the auxiliary services management service added description of the new POST api/services/favorite method for obtain favorite services for passengers; • in Attachment 1: IBE error codes added a new error code 6100; • in Attachment 2: dictionary "Types of passengers" added new types SNN, GCF, JCB. 	Maslyanny I. V.

25.06.2021	2.3.76	<p>Added:</p> <ul style="list-style-type: none"> to GET api/Flights response, values totalBaseFare, totalDiscount to Attachment 2. Dictionaries "Passenger types": types YCD,INS to Attachment 4. Partner's services "Aeroexpress service codes": STANDARTRT code <p>Changed:</p> <ul style="list-style-type: none"> in Attachment 2. Dictionaries "Auxiliary services types": Hindu -> Indian Cuisine; Sea food meal -> Hot seafood; Hot meal ->Hot meat dish. in Attachment 1. IBE error codes Amadeus errors: 1130 Unknown error appeared during refund process. -> Voluntary refunds are not available as one of the passengers has successfully checked in for the flight. in Attachment 2. Dictionaries "Passenger types": CHR->JNN 	Kulich K.V.
26.05.2021	2.3.75	<p>Added:</p> <ul style="list-style-type: none"> a description of the new POST payment/email method to the booking payment service for sending a payment letter to the mail; in the GET api/reservation method response a new isPayForbiddenForAgency parameter; in the GET api/flights and GET api/reservation methods responses a new parameter containsInternationalFlights; in the Attachment 1: IBE error codes - new error codes 1140, 1141, 5010; in the Attachment 2: dictionary "Types of Amadeus offices" - new offices; in the Attachment 2: dictionary "Types of passengers" - new types of passengers CHR, MIL, DIS, MDP; in the Attachment 2: dictionary "Types of supplementary services" - new services WCHR, WCHS, WCHC, CKIN. 	Kirillova A.N.

27.04.2021	2.3.74	<p>Added:</p> <ul style="list-style-type: none"> • the AccountCode request parameter has been added to the Service for obtaining the minimum prices: api/calendar and the Service for receiving flights: api/flights • in the Reservation, v2.3 added the isAgentReservation parameter; • in the /api/v2.3/Checkin/GetPassengerChekinStatus response the FlightReferenceNumbers parameter an array of strings; • the POST /api/v2.2/Payment/Email request with lang and redirectLang string variables. Generates a link to payment to be sent to a letter to be sent to the client. <p>Changed:</p> <ul style="list-style-type: none"> • in the Terms of Reference: for the test environment, the endpoint has been changed, from which get information about nutrition in the tariff options from https://loyalty-tst.u6.ru/WebTicket/foodinfo/getjson to https://webticket-tst.uralairlines.ru/foodinfo/getjson 	Kirillova A.N.
------------	--------	--	-------------------

25.03.2021	2.3.73	<p>Added:</p> <ul style="list-style-type: none"> • in the Attachment 2: dictionary - new types of passengers: UNN, STR; • in the Attachment 2: dictionary - new service packages: PAKR and PAKZ; • in the POST /Refund/Calculate service response, the parameter refundWithTicketOnlyServices is an array of strings of the data subcategory; • in POST /Reservation and /Flights/Search service requests, the integer parameters StrAdults, ChrChildren and UnaccompaniedMinors; • in the response of the /Flights/Search request, the ruleKey string along the path of the data.recommendations.outboundFareAndAmountDetails <p>Changed:</p> <ul style="list-style-type: none"> • in the Terms of Reference: for the test environment, the endpoint has been changed, from which get information about nutrition in the tariff options from https://webticket.uralairlines.ru/test/foodinfo/getjson to https://loyalty-tst.u6.ru/WebTicket/foodinfo/getjson 	Kirillova A.N.
------------	--------	--	-------------------

25.02.2021	2.3.72	<p>Added:</p> <ul style="list-style-type: none"> • in the GET api/Catalog service response the subCategory parameter of the additional subcategory services; • in the POST api/Refund/Partial service request an array services with information about the returned services of AK; • in the general technical information on refund, new statuses of refund sessions AutoRefundCompleted and ServicesCanceled; • for the service of voluntary return of tickets and services, a method for registering an order for auto return in BPC - bmmRefund; • in the GET api/Calendar service request for searching through SIG the twp parameter, which is an indicator whether the calendar will be requested for +-7 days from the current date; • in the GET api/v2.3/Reservation response the bookingTimeLimit parameter of the time and date until which the unpaid booking will exist; • in the remark on the added markup, the ChargeType parameter indicating the type of the added markup (markup from DMS or dynamic markup from getContactMarkup); • in Attachment 4: Partner services in the table Service codes Alpha insurance new AVIADYNAMICBUS tariff returned for Business tariffs; <p>in Attachment 2: dictionary of types of additional services CVID service.</p> <p>Changed:</p> <ul style="list-style-type: none"> • the return scenario, due to the addition of the auto return option; • in Attachment 2: dictionary of additional XBLL and XBLS services were moved from the dictionary of Types of additional services supported by the IBE (not sold in the IBE) to the dictionary of Types of additional services, the category and description have been changed. 	Kirillova A.N.
------------	--------	---	-------------------

01.02.2021	2.3.71	<p>Added:</p> <ul style="list-style-type: none"> • In the GET api/Flights response a block alternativeFlights with alternative directions from CRM; • the POST api/refund/partial request, in the request the parameter isTicketRefund a sign to return a ticket or only services and the parameter partnerProducts a block of returned partner services; • the examples of requests for a full refund of a ticket with services and refund of partner services; • in the api/reservation response the availability parameters and the reasons for the cancellation unavailability, respectively, isAnnulAvailable and annulUnavailabilityReason, as well as the isRefundAvailable flag for availability of refunds; • a description of two methods for adding and removing services from an order: api/services/add, api/services/remove; • in the Attachment 2: dictionary of types of additional services added meals codes: ECML, FCML, ACML, RCML, CSML, RUML, ORML, MEML, GEML, ASML; • in the Attachment 2: dictionary of tariff options added noMealsOnBoard option "No meals". 	Kirillova A.N.
23.12.2020	2.3.70	<p>Added:</p> <ul style="list-style-type: none"> • in the GET api/Partners/{id} response, an array of compatibleProductCodes of partner service rate codes that can be added to the booking together with the current one; • in the GET api/Reservation response, in the appliedDiscounts block, the serviceReference parameter indicating the identifier of the service for which the discount is applied within the current booking, and the promotionType parameter indicating the type of promo code (for a flight or a service); • in the Attachment 2: dictionary of types of additional services, service codes BLOU - business lounge and VLOU - Vip lounge; • in the Attachment 3 the information on adding the SSR for partner services. 	Kirillova A.N.

30.11.2020	2.3.69	<p>Added:</p> <ul style="list-style-type: none"> • the excludedPeriods block with information about the period for excluding the validity of the promo code and the loyaltyCardIsRequired parameter - a flag indicating the promo code for DCL in the POST api/Promotions response; • the parameters arrivalTerminal and departureTerminal with information about the terminal number at the point of arrival and departure, respectively, and the hash parameter with a unique flight hash in the GET api/Reservation response; • the parameters loyaltyCardValidTo, indicating the expiration date of the loyalty card, and serviceClassUpOrdered, indicating the upgrade for the owner of the "Wings" gold card, in the POST api/Reservation request; • the parameters of the GET api/Flights request affecting the search via SIG: mode - search mode, ofn and rfn - arrays of outbound and return flights respectively; • the GET api/Flights response parameters: outboundFlightGroupHash and returnFlightGroupHash - hashes of the outbound and return flight groups respectively; outboundFareFamilyCode and returnFareFamilyCode - Fare family codes for outbound and outbound directions respectively. 	Kirillova A.N.
03.11.2020	2.3.68	<p>Added:</p> <ul style="list-style-type: none"> • the preselectedOption block with information about preselection for partner services in the GET method api/Partners/{id} response; • the example of a response to a Telemedicine request. 	Kirillova A.N.
09.10.2020	2.3.67	<p>Added:</p> <ul style="list-style-type: none"> • the passengerReference parameter for POST to api/Partners request to add telemedicine; • the orderId parameter of the order number of the partner service in the POST api/Partners service response. 	Kirillova A.N.

15.09.2020	2.2.66	<p>Added:</p> <ul style="list-style-type: none"> • in the GET api/reservation method response the appliedCertificates parameter containing information about the applied certificates; • a new method GET api/Checkin/GetPassengerChekinStatus which allows to get information about the registration availability, taking into account the registration availability in the context of each segment. 	Kirillova A.N.
27.07.2020	2.2.64	<p>Added:</p> <ul style="list-style-type: none"> • for the POST api/payment request the returnUrl parameter of return address. <p>Changed:</p> <ul style="list-style-type: none"> • for the GET api/catalogue v2.3 response in the services, packs and seatService arrays the referenceNumber reference to the flight segment has been changed to an array of references to the flight segments referenceNumbers. <p>Removed:</p> <ul style="list-style-type: none"> • for the POST api/services v2.3 request the insurance and partnerProducts blocks; • in the GET api/catalog v2.3 response block insuranceProducts. 	Kirillova A.N.

<p>26.06.2020</p>	<p>2.2.63</p>	<p>Removed:</p> <ul style="list-style-type: none"> • the information about an outdated GET api/discounts request; • in the Attachment 4 outdated Aeroexpress service codes: ADVSTANDART - Preliminary Standard, STANDARTRT - Round Trip, PAIROW - Paired, PAIRRT - Paired Round Trip, GROUPOW - Group, GROUPRT - Group Round Trip. <p>Added:</p> <ul style="list-style-type: none"> • in the GET api/catalogue method response the crmEmdPromoCodes parameter containing information about promotional codes for additional services received from the CoreApi service; • in the GET api/flights method response the aircraftIataCode parameter with the value of the aircraft IATA code; • in the GET, POST api/reservation methods responses the aircraftIataCode parameter with the value of the aircraft IATA code; • a new service POST api/discounts, which allows you to get an applicable discount campaign for the specified search parameters; • a new service for managing certificates GET, DELETE api/certificates, which allows to receive information about certificates and delete them for the current number; • a new service POST api/CatalogStandalone to get a catalog of services without creating a PNR. <p>Changed:</p> <ul style="list-style-type: none"> • the service api/SeatmapStateless renamed into api/SeatmapStandalone. 	<p>Kirillova A.N.</p>
-------------------	---------------	--	---------------------------

01.06.2020	2.2.62	<p>Removed information about deprecated version api 2.1</p> <p>Added:</p> <ul style="list-style-type: none"> • in the GET api/catalogue method response a block with packs of services and an array of seat characteristics - seatCharacteristics; • in the GET api/seatmap method response the isSeatUnavailable parameter and an array of objects with information about the cost of seats in terms of passengers passengersTotalData; • a new service POST api/SeatmapStateless to get a seatmap without creating a PNR; • in the GET api/reservation method response the packReferenceNumber parameter of the reference to the package of services added to the booking; • in the Attachment 2 a dictionary of service packages; • in the Attachment X an example of a successful request and response for the POST api/SeatmapStateless service. 	Kirillova A.N.
06.05.2020	2.2.61	<p>Added:</p> <ul style="list-style-type: none"> • the GET and POST api/annul methods; • the GET api/reservation method response parameters: <ul style="list-style-type: none"> ○ reservationRetrieveStatus - booking receipt status; ○ engineType - the system in which the reservation was created; ○ isGroupReservation - flag defining group booking; ○ outboundBaggageAllowances и returnBaggageAllowances - arrays defining baggage allowances; • in the GET api/flights v2.2 response a parameter defining the sale rate - isSale; • in the Attachment 2 new types of meals and services supported by the IBE. 	Kirillova A.N.
03.04.2020	2.2.60	Added a promotion block with applied promo campaign in api/flights v2.2 response	Kirillova A.N.

02.03.2020	2.2.59	<p>Added:</p> <ul style="list-style-type: none"> • the MoneyWallCoefficient parameter in the response of api/flights and api/catalogue requests, for calculating the amount of the monthly payment when paying in installments; • the BaggageAllowance parameter in the recommendations block of the api/flights response, to determine the baggage allowance; • the NextNearestFlightDate parameter in api/flights request response; • the GET /api/system/Availability method to track the state of the backup IBE contour in IT Grad. 	Kirillova A.N.
27.02.2020	2.2.58	<p>Added:</p> <ul style="list-style-type: none"> • the description of the text parameter in the services block of the api/services request; • an example of a POST services request to add an animal carriage; • an animal reference book in the Attachment 2. 	Kirillova A.N.
03.02.2020	2.2.58	<p>Added:</p> <ul style="list-style-type: none"> • serviceCode, promoNumber parameters in appliedDiscounts block of api/reservation response; • text, petType parameters in services block of api/reservation response; • redirectLang parameter in POST api/payment request. <p>Changed:</p> <ul style="list-style-type: none"> • body and description of parameters for POST api/payment request 	Bodrova S.E.

25.12.2019	2.2.57	<p>Added:</p> <ul style="list-style-type: none"> the information about partners methods, which allow to request information about partners services separately from the api / catalog; the new types of meals: CAML (dessert) и KSML (kosher); <p>Changed:</p> <ul style="list-style-type: none"> the services SLSPORT and SLTRAVEL for partner Sealine are available for booking; the structure of the api/flights request, translations in the foodOnBoard block have been moved to the translation object. 	Kirillova A.N.
09.12.2019	2.2.56	<p>Added:</p> <ul style="list-style-type: none"> startFlightDate, endFlightDate, startDate, endDate parameters in POST api/promotions response <p>Removed:</p> <ul style="list-style-type: none"> serviceType parameter in GET api/calendar and GET api/flights requests 	Bodrova S.E.
20.11.2019	2.2.55	<p>Added:</p> <ul style="list-style-type: none"> showInBookingCart field in GET api/reservation response options. originOfficeId field in GET api/reservation response with original booking office id information; SearchPrice field in POST api/reservation response with information about price received in search query adults, children, infants, pensioners, youths, subsidizedChildren fields in POST api/reservation request with each type of passengers information; origin field - IATA source airport/hub in POST api/reservation request destination field - IATA destination airport/hub in POST api/reservation request clarifications in Attachment 2 for Amadeus office types dictionary; 	Vershinin A. V. Krutyakova E.S

08.11.2019	2.2.54	Added partnerProducts block in api/refund/calculate response	Bodrova S.E.
14.10.2019	2.2.53	<p>Added:</p> <ul style="list-style-type: none"> • list of partners in Attachment 4 and Sealine/Alfastrah partners services codes. • partners insurance purchase sample in IBE API v.2.2 api/services; • v. 2.3 changes in /api/services section; • partner's insurance information retrieve sample via api/reservation/partners; • field fareFamilies in prices → discount object in api/calendar response; • partners block in api/refund/calculate response; 	Vershinin A. V. Berezka O. M.
08.10.2019	2.2.52	<p>Attachment with samples is renamed and placed in the end of documentation. Updated other Attachment's numbers.</p> <p>Added:</p> <ul style="list-style-type: none"> • ticketFares and ticketFaresWithoutZapOff technical fields in POST api/reservation response. Used for information message display regarding SIG/Amadeus prices difference. • discountConstraints array in api/flights response. Contains promo action's application settings; • excludedNationalities array in api/flights response. Contains nationality codes restricted for transportation on specified flight. • error codes returned by /api/promotions service and corresponding description texts; 	Vershinin A. V. Voronina D. A.
25.09.2019	2.2.51	<p>Added:</p> <ul style="list-style-type: none"> • priceHidden field in GET api/reservation response. Allows to control IT-tariff price display. 	Vershinin A. V.

11.09.2019	2.2.50	<p>Added :</p> <ul style="list-style-type: none"> • serviceType parameter in api/calendar and api/flights queries. Allows to select preferred data source (Amadeus/SIG); • airShopSessionId field in api/flights response; • expectedPrice and airShopSessionId fields in POST api/reservation request. Used in booking preceded by SIG search flow. <p>General changes in documentation:</p> <ul style="list-style-type: none"> • removed information regarding obsolete api v 1.1 and 2.0 versions; • added v 2.3 in the list; • clarified booking flow process; • added information regarding 3rd party services exploiting IBE API. 	<p>Bodrova S.E.</p> <p>Vershinin A. V.</p>
22.08.2019	2.2.49	<p>Added:</p> <ul style="list-style-type: none"> • necessary ticket refund service description; 	<p>Berezka O. M.</p>
07.08.2019	2.2.48	<p>Changed version enumeration approach. Now version contains recommended API version and sprint number where latest changes were introduced.</p> <p>Added:</p> <ul style="list-style-type: none"> • GET settings/transliteration service description. Allows to retrieve Cyrillic transliteration rules; • foodOnBoards array in api/flights response. Contains information about meals available on flights; • foodOnBoards array in GET api/reservation. Contains information about meals available on flights. 	<p>Smirnova S.A.</p> <p>Vershinin A. V.</p>
22.07.2019	1.4.47	<p>Added :</p> <ul style="list-style-type: none"> • api/reservation/partners service descriptions. Allows to retrieve information about partners services in booking and perform ticket/policy re-send for issued services. 	<p>Vershinin A. V.</p>

04.07.2019	1.4.46	<p>Added:</p> <ul style="list-style-type: none"> • bookingStatus field in GET api/catalogue response; • serviceOptions field clarification in api/catalogue response; • api/refund and api/refund/calculate method's description. Allows to perform voluntary refund of bookings. • api/passenger method description. Allows to make changes in passenger data in booking. • error codes in Attachment 1. 	Vershinin A. V.
10.06.2019	1.4.45	<p>Added:</p> <ul style="list-style-type: none"> • sessionMarker field description in GET api/reservation response 	Smirnova S.A.
29.05.2019	1.4.44	<p>Added:</p> <ul style="list-style-type: none"> • serviceOptions field in GET api/catalogue response; • api/reservation v2.2 changes (booking for subsidised passenger types) • api/services v2.2 changes (auxiliary services upsale after booking payment) • description of auxiliary services upsale process for payed booking. • information about possibility to authorise with token for api/user method. 	Berezka O. M. Vershinin A. V.
20.05.2019	1.4.43	<p>Recommended IBE API version upgraded from 1.1 to 2.1</p> <p>API v. 1.1 now considered as deprecated.</p> <p>Updated information about IBE errors in Attachment 1.</p>	Vershinin A. V.

28.04.2019	1.4.42	<p>Added:</p> <p>Auxiliary services.</p> <ul style="list-style-type: none"> • error codes in add auxiliary services in booking flow (Attachment 1) • api/services v2.1 service description. Added error structure format in response. <p>Aeroexpress.</p> <ul style="list-style-type: none"> • partnerProducts in api/catalogue response. (partners services information) • partners services information in api/services request • partners services information in api/reservation response • Attachment 5 with partners and partners services codes; • clarification regarding sending information to BPC for partners services in api/payment method <p>Misc:</p> <ul style="list-style-type: none"> • added information about remarks written in PNR (Attachment 4). • updated api/session service description 	Vershinin A. V.
12.04.2019	1.4.39	Added marketingCarrier field in api/reservation and api/flights responses.	Voronina D. A.
11.03.2019	1.4.39	<p>Changes introduced in 38-39 sprints.</p> <p>Added:</p> <ul style="list-style-type: none"> • promo-light, one bag, no bags tariffs description. Removed business-light from list of tariffs. • CNN passenger type (subsidised child); • subsidised tariff description in tariff dictionary. • CNN parameter description in api/calendar v1.1 request. • CNN parameter description and isSubsidized flag in api/flights v1.1 response. • validation rules description for creating bookings with subsidised tariffs. Method POST api/reservation v1.1. 	Vershinin A. V.

11.01.2019	1.4.37	<p>Added API v. 2.0 section and description</p> <ul style="list-style-type: none"> • introduced POST api/session method instead of api/token; • removed X-Api-Key and X-Session headers; • removed Authorization header; 	Smirnova S.A.
19.12.2018	1.4.35	<p>Added:</p> <ol style="list-style-type: none"> 1. Changes in GET api/reservation response <ol style="list-style-type: none"> 1. tickets array with information about tickets. 2. added spentBonuses and spentBonusesByPassengers fields 3. added originOfficeType field; 4. added hasSpecialCondition field; 2. Added information about token expiration time 3. Clarified POST api/token services work. 4. Added ID field in api/token response 5. Added X-Stamp header in api/flights method 	Vershinin A. V.

01.12.2018	1.4	<p>Added:</p> <ul style="list-style-type: none"> • booking flow description; • new token type, type=Skyscanner; • v 1.1 POST api/reservation method description. Added booking creation samples. • zapOff description (information about discounts) in api/pricing response; • format clarifications for api/catalogue method; • totalPrice field in api/flights response → recommendations which determines full ticket price subtracted by discounts applied. Added samples. • is Empty field in api/seatmap response → data. Availability of seats for selection within a segment flag; • api/discounts service description; • fields in api/user response, identifying country where document was issued; • ticketNumber parameter in GET api/reservation method. Allows to request retrieve booking information by ticket number and surname; • disc parameter in api/calendar service. Controls request of discounts. • api/ problemNotification service description; • api/receipt service description; • removed isValid field from POST api/promotions response; • update of cache parameter description in api/calendar service; • api/exchanges method description 	Vershinin A. V.
29.08.2018	1.3	<p>Upgraded API version to 1.1 Added</p> <ul style="list-style-type: none"> • api/airports service descriptions; • api/payment v 1.1 service description; • cache parameter description for api/calendar services • api/health service description 	Vershinin A. V.

19.06.2018	1.2	<ol style="list-style-type: none"> 1. Clarified description of parameters in POST api/payment method 2. Added tariff options and categories description. Attachment 2. 3. api/services service is split into GET api/catalogue for retrieving of auxiliary services and POST api/services for adding auxiliary services into booking. 4. Changed api/seatmap service type from POST to GET. Added passengerRelatedPrice structure description. 5. Added samples of flightDurationUntilNextStop field usage in api/flights response. 	Vershinin A. V.
04.06.2018	1.1	Added description of basic API services needed for booking creation.	Vershinin A. V.
23.05.2018	1.0	Initial version of document.	Vershinin A. V.

Glossary

U6	Ural Airlines company
IBE, booking engine	Internet Booking Engine, U6 new ticket booking system. IBE API is described in this document.
Amadeus	Global ticket booking system (GDS)
SIG	Integrator(vendor) of searching solutions from different GDS.
CRM	Customer management system integrated with IBE
BPC	Booking payment system integrated with IBE
WSAP	Web Service Access Point
PNR	Passenger name record on Amadeus side. Contains information regarding flights, passenger data, auxiliary services, discounts etc.
SSR	Special Service Request. In current documentation almost always treated as "separate record in PNR". Record might contain information about passenger, services and other.
TST	Transitional Stored Ticket, part of Amadeus document with information regarding tariff options in PNR.
TSM	Transitional Stored Miscellaneous, part of Amadeus document with information regarding auxiliary services, added in PNR.

General provision

Purpose of this document is to provide IBE API methods description and its usage samples for 3rd party developers.

IBE API represents REST API which allows to search tickets and prices, book tickets on different flight directions, add auxiliary services and perform payment.

IBE API appears to be an interlayer between Amadeus/SIG/CRM/BPC systems, other partners services and end-user client applications.

General information about API methods

API Versions

Recommended API version: v2.3

Recommended to use for all kind of requests.

Queries without specifying of API version is prohibited (exception: api/health service);

Supported versions: v2.2, v2.3

Deprecated versions: v1.1, v2.0

General information about API v2.x methods

Main differences between v 2.x and v 1.x:

Added possibility of using unique API-key instead of token. It allows to request flights and calendar information without necessity to create a session.

If this information needs to be stored in booking context then you should create a booking session via api/v2.x/session method and pass session ID in X-Session header. X-API-Key header should be also be passed.

At present following keys created for:

Arcadia (booking engine website),

Axmor (mobile application),

CRM (obtaining prices for mailing purposes),

BPC (test booking creating),

Skyscanner (obtaining prices for meta-search),

Aviasales (obtaining prices for meta-search)

API keys are provided by request.

HTTP(S)-headers v2.x

For IBE API v2.x queries following HTTP(S) headers should be used:

Header name	Value	Purpose	Exceptions
X-API-Key	xApiKey - key value. Depends on app type (arcadia, axmor, skyscanner)	authorisation API-key	GET Airports method GET ExchangeRates method POST ProblemNotification method GET Settings/Transliteration method
X-Session	sessionKey — value, retrieved from API Session query	Session management via HTTP header	POST session method GET flights method GET calendar method POST promotions method GET reservation method Services methods except renew and problemNotification.
Content-Type	application/json	General	No

IBE API services query

IBE API query string in common case looks like: <http://host/api/apiVersion/serviceName>

where host – server, where booking engine is deployed

api – constant,

apiVersion – necessary IBE API version in «v2.x» format

serviceName – IBE API service name, for ex. «session» или «flights»

IBE API services response codes

IBE API handles exception situations thrown during service query execution. Following responses might be returned:

200 OK – both success and failed. In case of failed response, success field in response will be set to false. Error codes array will contain error code and description.

400 Bad Request – in case if query string was incorrect. Response model might differ from one used for correct service response.

401 Unauthorized – in case if API-key was not passed

Also codes specific for particular service might be returned.

Booking flow description

General booking flow

Booking flow consists of following API v2.x queries:

№	API Method	Description
1	POST api/session	Retrieving of unique booking session. Booking context is stored in session.
2	GET api/flights	Retrieving information about flights, tariffs and prices.
3	POST api/reservation	Booking creation: <ol style="list-style-type: none">1. Adding flights information2. Adding passenger's data3. Adding tariff's data Can be done in two ways <ol style="list-style-type: none">1. Single query passing all data2. Two queries passing flights data first, passenger and tariff data second IBE frontend uses 2nd way.
4	GET api/catalogue	Optional. Obtaining list of auxiliary services (U6 and partners) and its prices
5	GET api/seatmap	Optional. Obtaining plane's seatmap and seat prices.
6	POST api/services	Optional Adding auxiliary services (u6's or partner's) to booking Available services can be retrieved on steps 4 and 5 (see above)
7	GET api/partners/types	Optional Retrieve list of available partners.

8	GET api/partners/[id]	Optional Retrieve list of services available for specific partner. Where id is partner's number (presented in attachment)
9	POST api/partners	Optional. Adding of partner's service to booking.
10	POST api/payment	Obtaining link for redirection to payment page
11	GET api/reservation	Optional Retrieving information about paid booking (issued tickets etc)

Description of partners services upsale process after payment

Starting from API 2.2 version IBE supports possibility to upsale auxiliary services in context of new session

№	API Method	Description
1	-	Booking creation and payment according process described above.
2	POST api/session	Obtaining of new session which will be used for services upsale
3	GET api/reservation	X-Session header is mandatory Loading of payed booking context into session
4	GET api/catalogue	Obtaining list of auxiliary services (U6 and partners) and its prices
5	GET api/seatmap	Optional. Obtaining plane's seatmap and seat prices.
6	POST api/services	Adding auxiliary services (u6's or partner's) to booking
7	POST api/payment	Obtaining link for redirection to payment page

Notes:

1. If booking was not paid, a link for payment of full order will be formed (for all unpaid booking parts)
2. Upsale for paid booking is temporary not available (api/catalogue will not return information in case booking was paid)

API methods description

Obtain session service

API service	session
Short description	Allows to obtain session for storing booking context
Supported HTTP(S)-methods	POST
Conterminous systems methods used	no

POST session method

Description: new session creation

At present session expires in 60 minutes each API query automatically prolongates it.

Request format /api/apiVersion/session

Request header

X-API-Key - unique GUID of system using IBE API.

Request paramaters

No parameters

Response format

```
{  
  "sessionKey": "string",  
  "sessionId": "string"  
}
```

Response parameters

Parameter	Description
sessionKey	Session key Used for storing of booking context

sessionId	Id session service field
-----------	-----------------------------

Response sample

```
{  
  "sessionKey": "ad712a62-3d18-4d3a-bd3b-6d883d111aaf",  
  "sessionId": "654cc0ec-c857-955f-af82-792a239675b6"  
}
```

Obtain minimal prices service

API service	calendar
Short description	Obtaining of minimal ticket prices by dates. Mostly used for price display in calendar
Supported HTTP(S)-methods	GET
Conterminous systems methods used	Amadeus: Fare_InstantFlexPricer и Fare_FlexPricerCalendar SIG: SIG_AirShop, SIG_AirMinPrice CRM: CalculateDiscountList2

GET calendar method

Request format: /api/apiVersion/calendar

Request samples:

One way flight, search through Amadeus:

/api/apiVersion/calendar

?orig=LED&dest=DME&dep=20180524&ret=&adt=2&chd=1&inf=0&src=0&yth=0&cnn=0&mth=false

Two way flight, search through SIG:

/api/apiVersion/calendar

?orig=LED&dest=DME&dep=20180610&ret=20180620&adt=1&chd=0&inf=0&src=0&yth=0&mth=false

Request parameters

IMPORTANT: either subsidised (SRC / YTH / CNN) or non-subsidised (ADT / CHD / INF) set of passengers types should be used in request.

Parameter	Type	Description	Value samples
orig	string	Original airport IATA-code	LED
dest	string	Destination airport IATA-code	DME
dep	date	Departure date	Format yyyyMMdd Sample: 20180801
ret	date	Return date	Формат yyyyMMdd Пример: 20180905
adt	integer	number of Adult passengers	Integer number in interval [0; 9]. Overall number of passengers must not exceed 9
chd	integer	number of Child passengers	
inf	integer	number of Infant passengers	
src	integer	number of Senior citizen passengers	
yth	integer	number of Youth passengers	
cnn	integer	number of passengers of «subsidised child» type	
mth	boolean	obtain values grouped by months false – standard request, true – request by months	false
twp	boolean	Request a value for a period of +-7 days from the current date or not (only works for search through SIG) false - standard request, true - request for values +-7 days from the current date	false

cache	string	<p>Determines whether cache data should.</p> <p>Actual only results returned from Amadeus search data provider. In case of SIG data provider parameter should be ignored.</p> <p>Default (0): request Fare_InstantFlexPricer Amadeus method first. If error occurs - request Fare_FlexPricerCalendar</p> <p>OnlyCache (1): request only Fare_InstantFlexPricer method in Amadeus.</p> <p>SkipCache (2) – request only Fare_FlexPricerCalendar method in Amadeus.</p> <p>Default value - 0</p> <p>This parameter should not be used in case mth parameter is set to true</p> <p>Changed Amadeus services querying:</p> <ol style="list-style-type: none"> 1. If requested date of flight <= 2 weeks from current data, Fare_FlexPricerCalendar queried straight away 2. FlexPricerCalendar also requested for periods 15.12 - 15.01 and 22.04 - 15.05 (all dates included) 	0
-------	--------	--	---

disc	boolean	<p>Service parameter, allows to control discount querying.</p> <p>Default value - true.</p> <p>disc = false used in querying information for swiper interface elements.</p> <p>Only Fare_FlexPricerCalendar Amadeus method gets queried.</p> <p>disc = true used for querying of information for calendar with discounts available by months.</p> <p>In this case Amadeus Fare_FlexPricerCalendar method in Amadeus gets queried and also CRM method CalculateDiscountList2.</p>	-
AccountCode	string	Optional parameter. Used in SIG to display prices for specific account codes.	BFZO4
meta	boolean	Sign of metasearch.	
currency	string	Currency of recommendations.	EUR

Response format

```

{
  "data": {
    "prices": [{
      "price": number,
      "fareAmounts": [{
        "paxType": string,
        "paxCount": integer,
        "taxAmount": number,
        "totalAmount": number
      }
    ]
  },
  "flightsNumbers": [string],
  "date": datetime,
  "discount": {
    "id": string,
    "name": string,
    "amount": integer,
    "discountType": string,
    "usageType": string,
  }
}

```

```

        "startDate": datetime,
        "endDate": datetime,
        "fareFamilies": [string]
    },
    "returnPrices": [{
        "price": number,
        "fareAmounts": [{
            "paxType": string,
            "paxCount": integer,
            "taxAmount": number,
            "totalAmount": number
        }
    ],
    "flightsNumbers": [string],
    "date": datetime
}
]
}
],
"range": integer,
"currency": string
},
"errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
]
}
],
"success": boolean
}

```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	range	Date interval for which data was returned
	currency	Currency of recommendations.
	prices	Array of objects with price information
	price	Minimal price for one way flight for date specified in prices.date field

		date	<p>One way flight date with minimal price returned by service</p> <p>Format: yyyy-MM-ddTHH:mm:ss</p> <p>Example: 2018-06-03T00:00:00</p>
		flightsNumbers	<p>Array of strings with flights number's information for specific flight direction.</p> <p>Contains several values if flight contains transfers.</p>
		fareAmounts	Array of objects with ticket price information for specific passenger
		paxType	<p>Passenger type.</p> <p>Possible values: <u>Attachments, passenger types</u></p>
		paxCount	Amount of passengers of specific type
		taxAmount	Tariffs and fees
		totalAmount	Full ticket price including tariffs and fees
		discount	<p>Object with information about discount available for present flight</p> <p>Discount may absent for certain transfers if flights are in the list of exceptions. Exceptions stored on CRM and IBE side.</p>
		id	Discount campaign ID in CRM
		name	Discount campaign title
		amount	Discount amount
		discountType	<p>Discount type, possible values</p> <p>0 – Percent</p> <p>1 – Roubles</p> <p>It is recommended to check discount type against its value. For ex for type 0 amount should be within [0;100] interval.</p>

			usageType	Discount application type, possible values: 0 – Fare 1 – Order
			startDate	Discount interval start date
			endDate	Discount interval end date
			fareFamilies	Strings array with information about fareFamilies for which discount may be applied.
		returnPrices		Array of objects, which contains minimal prices for return flight dates which matches with dates specified in prices.date. Data is retrieved from corresponding Amadeus recommendations.
			price	Return flight price for returnPrices.date date
			date	Date of return flight with minimal price returned by service
			flightsNumbers	Array of strings with flights number's information for specific flight direction. Contains several values if flight contains transfers.
			fareAmounts	Array of objects with ticket price information for specific passenger Analogue of data.prices.fareAmounts. For one way flight array will be empty.
			paxType	Passenger type. Possible values: <u>Attachments, passenger types</u>
			paxCount	Amount of passengers of specific type
			taxAmount	Tariffs and fees
			totalAmount	Full ticket price including tariffs and fees
errors				Information about errors appeared during query execution.
			ibeCode	Error code, see <u>EN Attachment 1. IBE error codes</u>

	description	Code description, see. EN Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Amadeus error code
	text	Amadeus error text

Error request and response samples

Request:

api/apiVersion/calendar?orig=BAX&dest=VVO&dep=20190821&ret=&adt=1&chd=0&inf=0&src=0&yth=0&cnm=0&mth=false&disc=false&cache=0

Parameter dep contains date in the past.

Response:

```
{
  "errors": [
    {
      "ibeCode": 1008,
      "description": "Data validation error.",
      "messages": [
        {
          "code": "DepartureDate",
          "text": "The specified condition was not met for
'Departure Date'."
        }
      ]
    }
  ],
  "success": false
}
```

Obtain flights service

Obtain flights service v2.2

API service	flights
Short description	Obtain list of available flights for specified dates and recommendations with ticket prices
Supported HTTP(S)-methods	GET
Conterminous systems methods used	Amadeus: Fare_FlexPricerUpsell SIG: SIG_AirShop CRM: CalculateDiscount

GET flights method

Description: obtain list of flights and flight's prices

Request format: /api/apiVersion/flights

Request samples

One way flight:

/api/apiVersion/flights

?orig=LED&dest=SVX&dep=20180610&ret=&adt=1&chd=1&inf=0&src=0&yth=0&cnn=0

Two way flight:

/api/apiVersion/flights

?orig=LED&dest=PRG&dep=20180526&ret=20180531&adt=1&chd=0&inf=0&src=0&yth=0&cnn=0

Request parameters

IMPORTANT: either subsidised (SRC / YTH / CNN) or non-subsidised (ADT / CHD / INF) set of passengers types should be used in request.

Parameter	Type	Description	Value samples
-----------	------	-------------	---------------

orig	string	Original airport IATA-code	LED
dest	string	Destination airport IATA-code	DME
dep	date	Departure date	Format yyyyMMdd Sample: 20180801
ret	date	Return date	Формат yyyyMMdd Пример: 20180905
adt	integer	number of Adult passengers	Integer number in interval [0; 9]. Overall number of passengers must not exceed 9
chd	integer	number of Child passengers	
inf	integer	number of Infant passengers	
src	integer	number of Senior citizen passengers	
yth	integer	number of Youth passengers	
cnn	integer	number of passengers of «subsidised child» type	
mode	string	Search mode (optional parameter, affects search via SIG)	mono - only mono-recommendations are requested, for example, PR there and PR back; full - all recommendations are requested.
ofn	string	Array of one-way flight numbers (optional parameter, affects search via SIG)	60

rftn	string	Array of return flight numbers (optional, affects search via SIG)	371,392
AccountCode	string	Personal account client code (optional parameter, affects search via SIG)	BFZO4
meta	boolean	Sign of metasearch.	
currency	string	Currency of recommendations.	EUR
cabin	string	Class of service	Economy, Business
SearchAllBrands	bool	Search for all brands	

Response format

```

{
  "data": {
    "airShopSessionId": string,
    "currency": string,
    "outboundFlights": [flightGroup],
    "returnFlights": [flightGroup],
    "fareFamilies": [{
      "fareFamilyId": integer,
      "code": string,
      "shortCode": string,
      "category": string,
      "isSubsidized": boolean,
      "order": integer,
      "commercialFamily": string,
      "canRefund": boolean,
      "hasUpsale": boolean "options": [{
        "code": string,
        "category": string,
        "order": integer,
        "showInPopup": boolean,
        "showInSummary": boolean,
        "showInUpsell": boolean,
        "showInFareIncluded": boolean,
        "showForAdult": boolean,
        "showForInfant": boolean
      }
    ]
  }
},
  "foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string "commercialFamily": string,

```



```

    }
  ],
  "outboundBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer
  }
],
"returnBaggageAllowances": [{
  "quantity": integer,
  "weight": integer,
  "passengerType": integer
}
],
"discount": {
  "id": string,
  "name": string,
  "amount": integer,
  "discountType": integer,
  "usageType": integer,
  "startDate": datetime,
  "endDate": datetime
},
"promotion": {
  "number": string,
  "amount": integer,
  "promotionStatus": string,
  "promotionType": string,
  "description": string,
  "startFlightDate": datetime,
  "endFlightDate": datetime,
  "startDate": datetime,
  "endDate": datetime,
  "isUnique": boolean,
  "campaignId": string,
  "directions": [{
    "id": string,
    "origin": string,
    "destination": string
  }
],
"services": [{
  "code": string
}
],
"families": [{
  "code": string
}
],
"hasConflict": boolean,
"excludedFlights": [{
  "flightNumber": string,
  "departureAirport": string,
  "arrivalAirport": string
}
]
}

```



```

    }
  ],
  "moneyWallCoefficient": integer,
  "containsInternationalFlights": bool
},
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }
]
}
],
"success": boolean
}

```

flightGroup object

```

{
  "flightGroupId": integer,
  "flights": [{
    "flightId": integer,
    "departureAirport": string,
    "departureTerminal": string,
    "arrivalAirport": string,
    "arrivalTerminal": string,
    "departureDateTime": datetime,
    "excludedNationalities": [
      string
    ]
  ],
  "arrivalDateTime": datetime,
  "aircraft": string,
  "aircraftIataCode": string,
  "techStops": [{
    "arrivalDateTime": datetime,
    "departureDateTime": datetime,
    "airportCode": string,
    "stopDuration": string,
    "flightDurationUntilNextStop": string
  }
],
  "flightDuration": string,
  "flightDurationUntilNextStop": string,
  "stopDuration": string,
  "flightNumber": string,
  "marketingCarrier": string,
  "operatingCarrier": string,
  "codeshareName": string,
  "hash": string
}
],
"flightGroupDuration": string,

```

```
"hash": string
}
```

Response format if there are no flights for the requested dates (round trip)

```
{
  "data": {
    "outboundFlights": [],
    "returnFlights": [],
    "fareFamilies": [],
    "recommendations": [],
    "outboundNextNearestFlightDate": datetime,
    "returnNextNearestFlightDate": datetime,
    "foodOnBoards": [],
    "currency": string,
    "airShopSessionId": string,
    "moneyWallCoefficient": integer
  },
  "errors": [],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
airShopSessionId	SIG search session ID
currency	currency in which prices are represented
moneyWallCoefficient	Coefficient for calculating the monthly installment payment
containsInternationalFlights	Sign that one of the airports is located outside the territory of the RF
outboundFlights	Array of flightGroup objects with information about outbound flights

	returnFlights	Array of flightGroup objects with information about return flights. For one way flight array will be empty
	fareFamilies	Array of objects with information about fare families
	fareFamilyId	Fare family ID
	code	Fare family code, for ex: ECOLIGHTRU/ECOFLEXRU/ECOPREMURU etc.
	shortCode	Shortened code for fare family identification for example for fare families mentioned above: PR/EC/FL etc.
	category	Fare family category, for ex: ECOLIGHT/ECOFLEX/ECOPREM Possible values: see Attachment 2, Main fare family categories
	order	fare families display order
	commercialFamily	fare family type for ex: U6ECONOMY or U6BUSINESS
	isSubsidized	Sign of subsidised tariff
	canRefund	Sign of refundable tariff
	hasUpsale	Sign indentifying whether present tariff can be upgraded to next one in price category. Used for management of Upsale block display in frontend.
	options	Array of fareFamiliesOptions objects with information about tariff options. Possible values: see Attachment 2, tariff options
	outboundNextNearestFlightDate	The nearest departure date for one-way flight, if there are no flights on the requested date

	returnNextNearestFlightDate	The closest date of departure back, if the flight is RT and there are no flights for the requested dates
	foodOnBoards	Array of objects with information about meals available on the flight
	departureAirport	Airport of departure
	arrivalAirport	Airport of arrival
	flightNumber	Flight number
	commercialFamily	Fare family type (different type of meals for economy and business can be available for one flight)
	translation	Object with translation options
	ru	Options translation to russian
	en	Options translation to english
	zh	Option translation chinese
	alternativeFlights	Alternative destinations (returned from CRM if no flights were found on the selected date)
	departure	Departure airport of alternative direction
	arrival	Arrival airport of alternative destination
	discountConstraints	Array of objects with information about exceptions in discounts
	fareFamilies	Array of fare families for which discount should be applied
	flights	Array of flight numbers for which discount should be applied If at least one <u>of transfers</u> of the flight is not present in array, discount should not be applied
	excludedFlights	Array of objects with information about flight number exceptions
	flightNumber	Flight number
	departureAirport	Airport of departure
	arrivalAirport	Airport of arrival
	recommendations	Array of recommendations objects for flights

	outboundFlightGroupId	outbound flight number, matches one of outboundFlights.flightGroupId numbers
	outboundFareFamilyId	outbound fare family number, matches one of fareFamilies.fareFamilyId numbers
	returnFlightGroupId	return flight number, matches one of returnFlights.flightGroupId numbers
	returnFareFamilyId	return fare family number, matches one of fareFamilies.fareFamilyId numbers
	outboundFlightGroupHash	Flight group hash there
	returnFlightGroupHash	Flight group hash back
	outboundFareFamilyCode	Fare family code for outbound travel Matches fareFamilies.code
	returnFareFamilyCode	Return Fare Family Code Matches fareFamilies.code
	outboundPrice	outbound ticket's price
	outboundTaxes	outbound ticket's tariffs and fees
	returnPrice	return ticket's price
	returnTaxes	return ticket's tariffs and fees
	outboundSeatsLeft	Amount of remaining seats for present outbound recommendation
	returnSeatsLeft	Amount of remaining seats for present return recommendation
	totalPrice	Total ticket price including all applied discounts
	outboundFareAndAmountDetails	An array of fareDetails objects containing detailed information about fares for the "there" direction
	returnFareAndAmountDetails	An array of fareDetails objects containing detailed information about fares for the "return" route
	outboundBaggageAllowances	An array of objects containing detailed information about baggage allowances for the "there" route
	returnBaggageAllowances	An array of objects containing detailed information about baggage allowances for the "return" route

		discount	<p>Information about discount available for present flight</p> <p>Discount may absent for certain flight they are in the list of exceptions. Exceptions stored on CRM and IBE side.</p>
		id	Discount campaign id in CRM
		name	Discount campaign title
		amount	Discount amount
		discountType	<p>Discount type, possible values</p> <p>0 – Percent</p> <p>1 – Roubles</p> <p>It is recommended to check discount type against its value. For ex for type 0 amount should be within [0;100] interval.</p>
		usageType	<p>Discount application type, possible values:</p> <p>0 – Fare</p> <p>1 – Order</p>
		startDate	Discount interval start date
		endDate	Discount interval end date
		promotion	Information about the promo code applied for this flight.
		number	Promo code
		amount	Discount amount by promo code
		promotionStatus	<p>Promo code status</p> <p>Possible values: dictionary of promo codes statuses</p>
		promotionType	<p>Promo code type</p> <p>0 - promotional code for the flight</p> <p>1 - promotional code for an additional service</p>

		description	Description of the promo code campaign
		startFlightDate	Start date of the period of flights with a promo code
		endFlightDate	End date of the period of flights with a promo code
		startDate	Start date of the promo code validity period
		endDate	End date of the promo code validity period
		isUnique	A sign of the uniqueness of a promo code
		campaignId	Promo code campaign ID
		directions	Directions of exclusions / inclusions
		origin	Departure airport
		destination	Arrival airport
		services	An array of objects with information about the services that the promo code applies to
		code	Service code Possible values: dictionary of additional services (except insurance)
		excludedFlights	Common exclusions - flights for which discounts are not indicated
		flightNumber	Flight number
		departureAirport	Departure airport
		arrivalAirport	Arrival airport
		hasConflict	Sign of success in adding a promotional code to a booking False if errors occurred while adding the promo code
errors			Information about errors appeared during query execution.
		ibeCode	Error code, see EN_Attachment 1. IBE error codes
		description	Code description, see. EN_Attachment 1. IBE error codes

messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by <code>ibeCode</code> .
<code>code</code>	Amadeus error code
<code>text</code>	Amadeus error text

Object `flightGroup`

Contains information about flights

data.outboundFlights or data.returnFlights	
<code>flightGroupId</code>	index number used in recommendations
<code>flightGroupDuration</code>	Total flight time: summed up of duration values of each <u>transfer</u> and technical stop.
<code>hash</code>	Service field
flights	Array of objects with information <u>about transfers</u>
<code>flightId</code>	Flight id
<code>departureAirport</code>	Airport of departure
<code>departureTerminal</code>	Terminal in airport of departure
<code>arrivalAirport</code>	Airport of arrival
<code>arrivalTerminal</code>	Terminal in airport of arrival
<code>departureDateTime</code>	Departure date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T17:25:00
<code>excludedNationalities</code>	Array with information about passenger nationalities prohibited for transportation on present flight Example: ISR (for flights 3837 and 3838)
<code>arrivalDateTime</code>	Arrival date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T20:50:00

aircraft	Full aircraft title For ex.: Airbus A320
aircraftIataCode	IATA aircraft code
flightDuration	Duration of flight Длительность <u>перелёта для текущего сегмента??</u>
flightDurationUntilNextStop	Flight duration until <u>first</u> technical stop See <u>Attachment 3, service response samples</u>
stopDuration	Technical stop duration
flightNumber	flight number
marketingCarrier	Marketing carrier Example: U6, S7
operatingCarrier	Operating carrier (one actually performed flight) Example: U6, OK
codeshareName	Full aviacompany title performing code-share flight Example: Czech Airlines for operatingCarrier = OK For operatingCarrier = U6, present field is empty
hash	service field
techStops	Array of objects with information about technical stops
arrivalDateTime	Date and time of arrival to airport where technical stop is performed Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-26T05:30:00
departureDateTime	Date and time of departure from airport where technical stop is performed Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-26T08:00:00
airportCode	Airport code

	stopDuration	Technical stop duration
	flightDurationUntilNextStop	Flight duration until next stop <u>Attachment 3, service response samples</u>

Object fareFamiliesOptions

Contains information about tariff options. Used for tariff option display in IBE website.

data.fareFamilies.options	
code	Tariff option code Example: upTo5bonuses, refundExchangeProhibited, mealsOnBoard etc.
category	Option category code Example: bonus, ffServices etc.
order	Index number of option display
showInPopup	flag determining whether option should be displayed in "your options" pop-up
showInSummary	flag determining whether option should be displayed on Summary page.
showInUpsell	flag determining whether option should be displayed in tariff upgrade information block
showInFareIncluded	flag determining whether option should be displayed when filter is selected
showForAdult	flag determining whether option should be displayed for adult passenger.
showForInfant	flag determining whether option should be displayed for infant.

Object fareDetails

Contains information about tariff details

data.recommendations.outboundFareDetails or data.recommendations.returnFareDetails	
bookingCode	In general case it is first symbol in fareBasis

fareBasis	Base tariff title Example: WPROW или WECOW
isBreakpoint	Flag identifying whether tariff ends on present segment.
paxType	Passenger type. Possible values: Attachments, passenger types
paxCount	Amount of passengers of specific type
taxAmount	Tariffs and fees
totalAmount	Full ticket price including tariffs and fees
isSale	A parameter that defines the sales rate

Object BaggageAllowances

Contains information about baggage allowances

data.recommendations.outboundBaggageAllowances or data.recommendations.returnBaggageAllowances	
quantity	count of baggage
weight	weight of baggage
passengerType	passenger type

totalPrice field value examples

One way flight, no discounts

totalPrice is equal to value outboundPrice

```

▼ recommendations: [{outboundFlightGroupId: 0, outboundFareFamilyId: 0, outboundPrice: 2217, outboundTaxes: 1922,...}]
  ▼ 0: {outboundFlightGroupId: 0, outboundFareFamilyId: 0, outboundPrice: 2217, outboundTaxes: 1922,...}
    ► outboundFareDetails: [{bookingCode: "W", fareBasis: "WPROW", isBreakpoint: true,...}]
      outboundFareFamilyId: 0
      outboundFlightGroupId: 0
      outboundPrice: 2217
      outboundSeatsLeft: 9
      outboundTaxes: 1922
      totalPrice: 2217

```

One way flight with discount

totalPrice is calculated in the following way:

$$\text{totalPrice} = \text{outboundPrice} - \text{Floor}((\text{totalAmount} - \text{taxAmount}) * \text{discount})$$

[totalAmount – taxAmount] – need to calculate for all subtariffs and passenger types

Floor – round down operation.

$$\text{totalPrice} = 8051 - \text{Floor}((4415 - 2390) * 0,04) - \text{Floor}((3636 - 2321) * 0,04) = 8051 - \text{Floor}(81) - \text{Floor}(52,6) = 8051 - 81 - 52 = 7918$$

```
▼16: {outboundFlightGroupId: 5, outboundFareFamilyId: 0, outboundPrice: 8051, outboundTaxes: 4711,-}
  ▶ discount: {id: "1-1PSJTS1", amount: 4, discountType: 0, usageType: 0}
  ▼ outboundFareDetails: [{bookingCode: "W", fareBasis: "WPROW", isBreakpoint: true,-},-]
    ▼0: {bookingCode: "W", fareBasis: "WPROW", isBreakpoint: true,-}
      bookingCode: "W"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2390, totalAmount: 4415}]
        fareBasis: "WPROW"
        isBreakpoint: true
    ▼1: {bookingCode: "O", fareBasis: "OPROW", isBreakpoint: true,-}
      bookingCode: "O"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2321, totalAmount: 3636}]
        fareBasis: "OPROW"
        isBreakpoint: true
    outboundFareFamilyId: 0
    outboundFlightGroupId: 5
    outboundPrice: 8051
    outboundSeatsLeft: 9
    outboundTaxes: 4711
    totalPrice: 7918
```

Two way flight, no discounts:

$$\text{totalPrice} = \text{outboundPrice} + \text{returnPrice}$$

$$\text{totalPrice} = 6727 + 8051 = 14778$$

```
▼21: {outboundFlightGroupId: 1, outboundFareFamilyId: 1, returnFlightGroupId: 2, returnFareFamilyId: 0,-}
  ▶ outboundFareDetails: [{bookingCode: "W", fareBasis: "WECRT", isBreakpoint: true,-},-]
    outboundFareFamilyId: 1
    outboundFlightGroupId: 1
    outboundPrice: 6727
    outboundSeatsLeft: 9
    outboundTaxes: 4827
  ▶ returnFareDetails: [{bookingCode: "O", fareBasis: "OPROW", isBreakpoint: true,-},-]
    returnFareFamilyId: 0
    returnFlightGroupId: 2
    returnPrice: 8051
    returnSeatsLeft: 4
    returnTaxes: 4711
    totalPrice: 14778
```

Two way flight with discount:

totalPrice is calculated in the following way:

$$\text{totalPrice} = \text{outboundPrice} - \text{Floor}((\text{totalAmount} - \text{taxAmount}) * \text{discount}) + \text{returnPrice} - \text{Floor}((\text{totalAmount} - \text{taxAmount}) * \text{discount})$$

[totalAmount – taxAmount] – need to calculate for all subtariffs and passenger types for outbound and return directions.

$$\text{totalPrice} = 34472 - \text{Floor}((15308 - 2423) * 0,04) - \text{Floor}((19164 - 2404) * 0,04) + 7392 - \text{Floor}((4084 - 2404) * 0,04) - \text{Floor}((3308 - 2423) * 0,04) = 34472 - \text{Floor}(515,4) - \text{Floor}(670,4) + 7392 - \text{Floor}(67,2) - \text{Floor}(35,4) = 34472 - 515 - 670 + 7392 - 67 - 35 = 40577.$$

```
▼ 47: {outboundFlightGroupId: 1, outboundFareFamilyId: 2, returnFlightGroupId: 1, returnFareFamilyId: 1,-}
  ▶ discount: {id: "1-1P5J7S1", amount: 4, discountType: 0, usageType: 0}
  ▼ outboundFareDetails: [{bookingCode: "I", fareBasis: "IBLOW", isBreakpoint: true,-},-]
    ▼ 0: {bookingCode: "I", fareBasis: "IBLOW", isBreakpoint: true,-}
      bookingCode: "I"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2423, totalAmount: 15308}]
        fareBasis: "IBLOW"
        isBreakpoint: true
    ▼ 1: {bookingCode: "Q", fareBasis: "QFLOW", isBreakpoint: true,-}
      bookingCode: "Q"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2404, totalAmount: 19164}]
        fareBasis: "QFLOW"
        isBreakpoint: true
      outboundFareFamilyId: 2
      outboundFlightGroupId: 1
      outboundPrice: 34472
      outboundSeatsLeft: 2
      outboundTaxes: 4827
  ▼ returnFareDetails: [{bookingCode: "A", fareBasis: "AECRT", isBreakpoint: true,-},-]
    ▼ 0: {bookingCode: "A", fareBasis: "AECRT", isBreakpoint: true,-}
      bookingCode: "A"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2404, totalAmount: 4084}]
        fareBasis: "AECRT"
        isBreakpoint: true
    ▼ 1: {bookingCode: "W", fareBasis: "WECRT", isBreakpoint: true,-}
      bookingCode: "W"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2423, totalAmount: 3308}]
        fareBasis: "WECRT"
        isBreakpoint: true
      returnFareFamilyId: 1
      returnFlightGroupId: 1
      returnPrice: 7392
      returnSeatsLeft: 9
      returnTaxes: 4827
      totalPrice: 40576
```

Methods available for version 2.3

Method POST flights/conditions

Description: Allows to get the TFC (Tariff Application Conditions).

Request format /api/apiVersion/flights/conditions

Request body

```

{
  "rulesKeys": [
    {
      "key": string
    }
  ]
}

```

Request parameters

Parameter		Description
rulesKeys		Array of objects with information about the keys of the TFC
	key	Key TFC

Request body example

```

{
  "rulesKeys": [
    {
      "key": "ATPCO/000/00/Y6/-
550576457/1111/NPRMW/0074/ЕКВ/КЛД/0//14/20210708/@%#NPRMW@%#1A@%#@%#23ТЦ1"
    },
    {
      "key": "ATPCO/000/00/Y6/-
550576046/1148/NECMW/0074/ЕКВ/КЛД/0//14/20210702/@%#NECMW@%#1A@%#@%#23ТЦ1"
    }
  ]
}

```

Response body

```

{
  "data": {
    "faresRules": [
      {
        "key": string,
        "fareConditions": [
          {
            "number": integer,
            "title": string,
            "description": string
          }
        ]
      }
    ]
  }
},
"errors": [

```

```

    {
      "ibeCode": string,
      "description": string,
      "messages": [
        {
          "code": string,
          "text": string
        }
      ]
    }
  ],
  "success": boolean
}

```

Response parameters

Parameter		Description
data		The data returned by the request
faresRules		Array of objects with information about rules
	key	Key TFC
	fareConditions	Array of objects with rules for the key
	number	Number of the rule
	title	Title of the rule
	description	Description of the rule
errors		Information about errors appeared during query execution
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode
	code	Amadeus error code
	text	Amadeus error text
success		Sign of the success of the request If the errors array is empty, then the field value will be true

Method GET flights/allFlightStatusesForDay

Description: allows to get all flights for directions on specific date

Request format: /api/apiVersion/flights/allFlightStatusesForDay

Request parameters

Parameter	Description
date	Flight date
departureAirport	Departure airport
arrivalAirport	Arrival airport

Request example

```
/api/v2.3/flights/allFlightStatusesForDay?departureAirport=DME&arrivalAirport=SVX&date=2021-10-27
```

Response body

```
{
  "data": {
    "flightStatuses": [
      {
        "fnNumber": integer,
        "fnSuffix": string,
        "fnCarrier": string,
        "dayOfOrigin": datetime,
        "departureAirportActual": string,
        "arrivalAirportActual": string,
        "legState": string,
        "legType": string,
        "aircraftRegistration": string,
        "aircraftType": string,
        "departureAirportSched": string,
        "arrivalAirportSched": string,
        "departureDTSchedUTC": datetime,
        "arrivalDTSchedUTC": datetime,
        "departureDTSchedLocal": datetime,
        "arrivalDTSchedLocal": datetime,
      }
    ]
  }
}
```



```

    "estimatedArrivalDate": datetime,
    "offBlockDT": datetime,
    "airborneDT": datetime,
    "landingDT": datetime,
    "onBlockDT": datetime,
    "updatedDT": datetime,
    "checkinSinceDTUTC": datetime,
    "checkinUntilDTUTC": datetime,
    "checkinSinceDTLocal": datetime,
    "checkinUntilDTLocal": datetime,
    "departureDTUTC": datetime,
    "departureDTLocal": datetime,
    "arrivalDTUTC": "2021-10-27T07:40:00",
    "arrivalDTLocal": "2021-10-27T12:40:00",
    "isFlightDelayed": boolean
  }
]
},
"errors": [
  {
    "ibeCode": integer,
    "description": string,
    "messages": [
      {
        "text": string
      }
    ]
  }
],
"success": boolean
}

```

Response parameters

Parameter		Description
data		The data returned by the request
	flightStatuses	Array of objects with information about flights
	fnNumber	Flight number
	fnSuffix	Flight suffix
	fnCarrier	Airline operating the flight
	dayOfOrigin	Departure date of the scheduled flight
	departureAirportActual	IATA code of the airport of departure - actual
	arrivalAirportActual	IATA-code of the airport of arrival - actual
	legState	Flight status

	legType	Тип рейса
	aircraftRegistration	Flight type
	aircraftType	IATA aircraft type code
	departureAirportSched	IATA code of the scheduled departure airport
	arrivalAirportSched	IATA code of the scheduled arrival airport
	departureDTSchedUTC	Scheduled departure time UTC
	arrivalDTSchedUTC	Scheduled arrival time UTC
	departureDTSchedLocal	Scheduled departure time Local
	arrivalDTSchedLocal	Scheduled arrival time Local
	estimatedArrivalDate	Estimated time of arrival
	offBlockDT	Departure time from gate
	airborneDT	Take off time
	landingDT	Boarding time
	onBlockDT	Time of arrival at the parking lot
	updatedDT	Date of last update of flight information
	checkinSinceDTUTC	Check-in start for the flight UTC
	checkinUntilDTUTC	Check-in end for the flight UTC
	checkinSinceDTLocal	Check-in start for the flight UTC
	checkinUntilDTLocal	Check-in end for the flight UTC
	departureDTUTC	Departure time UTC
	departureDTLocal	Departure time Local
	arrivalDTUTC	Arrival time UTC
	arrivalDTLocal	Arrival time Local
	isFlightDelayed	Flight delay sign
	errors	Information about errors that occurred during the operation of the service.
	ibeCode	Error code, see Attachments 1. IBE error codes
	description	Error description, see Attachments 1. IBE error codes

	messages	Array of objects containing errors from the partner's response. The array may be empty, it is recommended to analyze the error by the <code>ibeCode</code> field.
	text	Error text
	success	Sign of the success of the request. If the errors array is empty, then the field value will be true.

Response body example

```
{
  "data": {
    "flightStatuses": [
      {
        "fnNumber": 273,
        "fnSuffix": " ",
        "fnCarrier": "U6",
        "dayOfOrigin": "2021-10-27T00:00:00",
        "departureAirportActual": "DME",
        "arrivalAirportActual": "SVX",
        "legState": "SKD",
        "legType": "J",
        "aircraftRegistration": "XXXXX",
        "aircraftType": "7M8",
        "departureAirportSched": "DME",
        "arrivalAirportSched": "SVX",
        "departureDTSchedUTC": "2021-10-27T05:25:00",
        "arrivalDTSchedUTC": "2021-10-27T07:40:00",
        "departureDTSchedLocal": "2021-10-27T08:25:00",
        "arrivalDTSchedLocal": "2021-10-27T12:40:00",
        "estimatedArrivalDate": "2021-10-27T07:40:00",
        "offBlockDT": "0001-01-01T00:00:00",
        "airborneDT": "0001-01-01T00:00:00",
        "landingDT": "0001-01-01T00:00:00",
        "onBlockDT": "0001-01-01T00:00:00",
        "updatedDT": "2021-03-23T06:57:00",
        "checkinSinceDTUTC": "2021-10-25T05:25:00",
        "checkinUntilDTUTC": "2021-10-27T04:45:00",
        "checkinSinceDTLocal": "2021-10-25T08:25:00",
        "checkinUntilDTLocal": "2021-10-27T07:45:00",
        "departureDTUTC": "2021-10-27T05:25:00",
        "departureDTLocal": "2021-10-27T08:25:00",
        "arrivalDTUTC": "2021-10-27T07:40:00",
        "arrivalDTLocal": "2021-10-27T12:40:00",
        "isFlightDelayed": false
      }
    ]
  },
  "errors": [],
}
```

```
"success": true
}
```

Method POST flights/particularFlightsStatuses

Description: allows to get information about the flight

Request format /api/apiVersion/flights/particularFlightsStatuses

Request body

```
{
  "requests": [
    {
      "departureAirport": string,
      "arrivalAirport": string,
      "date": datetime,
      "flightNumber": integer,
      "dateTimeRange": string
    }
  ]
}
```

Request body parameters

Parameter		Description
requests		
	departureAirport	Departure airport
	arrivalAirport	Arrival airport
	date	Departure date
	flightNumber	Flight number
	dateTimeRange	If send D, the search will be carried out one day from the start time

Request body example

```
{
```

```
"requests": [  
  {  
    "departureAirport": "DME",  
    "arrivalAirport": "SVX",  
    "date": "2021-08-03",  
    "flightNumber": 261  
  }  
]  
}
```

The response of this method matches the response of GET flights/allFlightStatusesForDay method

Booking management service

Booking management service v2.1

API service	reservation
Short description	Service allows to manage booking: - add information to booking (flight, passenger, tariff data); - retrieve booking information; - delete booking;
Supported HTTP(S)-methods	GET, POST, DELETE
Conterminous systems methods used	Amadeus: Create/update booking - PNR_AddMultiElements, Fare_PricePNRWithBookingClass, Ticket_CreateTSTFromPricing Retrieve booking information - PNR_Retrieve, Ticket_RetrieveListOfTSM Delete booking - PNR_Retrieve, PNR_Cancel, PNR_AddMultiElements

POST reservation method

Description: booking creation, adding flights, passengers and tariffs data.

There are 2 possible ways of booking creation:

1. Preliminary booking, when tariff with selected price bound to user's session (Amadeus session switches to Stateful state, it cannot be retrieved by other system's users).
2. Conventional booking, when tariff with selected price is not bound to user due to amount of Stateful sessions exceeds certain preconfigured value. In this case a race condition may appear when several users attempts to buy last ticket of specified tariff: ticket will be displayed to all users, but actual booking will be performed to one who will add passenger data first. Other users may receive error that selected tariff is no longer available.

At present on IBE website following booking algorithm is being used:

1. Execute create preliminary booking query (pass flight and amount of passenger data only).

2. If response is successful, then preliminary booking was created and on next step passenger's personal information and tariff data needs to be added to booking.
3. If an 1003 error code returned in response, then preliminary booking was not created and on next step full flight and passenger information needs to be added to booking.

/api/apiVersion/reservation **request format**

Body

```
{
  "flights": {
    "outboundFlights": [reservationFlight],
    "returnFlights": [reservationFlight],
    "adults": integer,
    "children": integer,
    "infants": integer,
    "pensioners": integer,
    "youths": integer,
    "origin": string,
    "destination": string,
    "passengersCount": integer
  },
  "passengers": {
    "passengers": [passengerData],
    "userAgreement": boolean
  },
  "fares": {
    "outboundFareFamily": {
      "code": string,
      "category": string,
      "isSubsidized": boolean
    },
    "returnFareFamily": {
      "code": string,
      "category": string,
      "isSubsidized": boolean
    },
    "fares": [{
      "fareBasis": string,
      "isBreakpoint": boolean
    }
  ],
  "sessionMarker": string
},
"expectedPrice": ingeger,
"searchPrice": ingeger,
"airShopSessionId": string
}
```

reservationFlight object

```
{
```

```

"origin": string,
"destination": string,
"departureDate": datetime,
"flightNumber": string,
"classOfService": string
}

```

passengerData object

```

{
  "id": integer,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": integer,
  "birthDate": datetime,
  "sex": integer,
  "nationality": string,
  "documentNumber": string,
  "documentExpiryDate": datetime,
  "loyaltyCard": string,
  "email": string,
  "phone": string,
  "documentType": integer,
  "disability": {
    "isDeaf": boolean,
    "isBlind": boolean,
    "hasGuideDog": boolean,
    "assistanceType": integer,
    "wheelchairType": integer,
    "commentary": string
  }
}

```

Request parameters

Parameter	Description
flights	Array of objects with information about <u>transfers</u>
outboundFlights	Array of reservationFlight objects with information about outbound flights.
returnFlights	Array of reservationFlight objects with information about return flights.
adults, children, infants, pensioners, youths, subsidizedChildren	Amount of passengers of each type

	origin	Origin airport
	destination	Destination airport
	passengersCount	Total amount of passengers
passengers		Object with information about passengers
	passengers	Array of passengerData objects with information about passengers
	userAgreement	Sign of user's consent for information mailing. Consent is sent to CRM for all added passengers.
fares		Object with information about tariffs
outboundFareFamily		Object with information about outbound Fare family
	code	Fare family code. Data gets retrieved from api/flights: data.fareFamilies.code response
	category	Fare family category Data gets retrieved from api/flights: data.fareFamilies.category response
	isSubsidized	sign of subsidised tariff
returnFareFamily		Object with information about return Fare family
	code	Fare family code. Data gets retrieved from api/flights: data.fareFamilies.code response
	category	Fare family category Data gets retrieved from api/flights: data.fareFamilies.category response
	isSubsidized	sign of subsidised tariff
fares		Array of objects with information about outbound and return tariffs

	fareBasis	Base tariff title Data gets retrieved from api/flights: data.recommendations.outboundFareDetails.fareBasis and data.recommendations.returnFareDetails.fareBasis response
	isBreakpoint	Flag identifying whether tariff ends on present segment. Data gets retrieved from api/flights: data.recommendations.outboundFareDetails.isBreakpoint and data.recommendations.returnFareDetails.isBreakpoint response
	sessionMarker	Aviasales session marker
	expectedPrice	Expected booking price, gets retrieved from api/flights response
	searchPrice	Search request price, retrieved from api/flights or api/flights/search response
	airShopSessionId	SIG search session ID, gets retrieved from api/flights response

reservationFlight

Parameter	Description
origin	Origin airport
destination	Destination airport
departureDate	Departure date and time Data gets retrieved from api/flights response: data.outboundFlights.flights.departureDateTime or data.returnFlights.flights.departureDateTime Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T01:00:00

flightNumber	<p>Flight number</p> <p>Data gets retrieved from api/flights response:</p> <p>data.outboundFlights.flights.flightNumber</p> <p>or</p> <p>data.returnFlights.flights.flightNumber</p>
classOfService	<p>Information about selected <u>tariff</u></p> <p>Data gets retrieved from api/flights response:</p> <p>data.recommendations.outboundFareDetails.bookingCode</p> <p>or</p> <p>data.recommendations.returnFareDetails.bookingCode</p>

passengerData Request Object

Parameter	Description
id	Passenger's index number (starting from 1)
firstName	Passenger's first name
middleName	Passenger's middle name (might be empty)
surname	Passenger's surname
passengerType	<p>Passenger's type</p> <p>Possible values: see <u>Attachment 2, passenger types</u></p>
birthDate	Date of birth
sex	<p>Passenger's sex</p> <p>Possible values: see <u>Attachment 2, passenger sex</u></p>
nationality	<p>Passenger's nationality</p> <p>Possible values: see <u>Attachment 2, Countries</u></p>
documentNumber	Document number

documentType	Document type Possible values: see Attachment 2, document types
documentExpiryDate	Document expiration date Mandatory only for international passport. For other type of documents automatically set to +10 years from present date.
loyaltyCard	"Wings" Loyalty card number
email	Email
phone	Phone
disability	Information about passenger's disability
isDeaf	sign of deafness
isBlind	sign of blindness
hasGuideDog	requires guide-doc escort. Should be sent only in junction with isBlind flag
assistanceType	Passenger's assistance type Possible values: see Attachment 2, assistant types
wheelchairType	Wheelchair type Possible values: see Attachment 2, wheelchair types
commentary	Commentary (optional)

Response format

```
{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "returnFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string,
      },
      "returnFareFamily": {
        "code": string,
      },
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    }
  }
}
```

```

},
"number": string,
"reservationDate": datetime,
"passengers": [{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": string,
  "birthDate": datetime,
  "sex": integer,
  "nationality": string,
  "documentNumber": string,
  "documentExpiryDate": datetime,
  "email": string,
  "phone": string,
  "documentType": integer,
  "isAdult": boolean,
  "isChanged": boolean
}
],
"services": [],
"seats": [],
"insurance": {},
"partnerProducts": [],
  "ticketFares": [{
    "baseAmount": integer,
    "passengerReferences": [{
      "referenceNumber": string,
      "referenceQualifier": string
    }],
    "passengerType": integer,
    "referenceNumber": string
    "referenceQualifier": string
    "totalAmount": 10964
  }],
  "ticketFaresWithoutZapOff": [{
    "baseAmount": integer,
    "passengerReferences": [{
      "referenceNumber": string,
      "referenceQualifier": string
    }],
    "passengerType": integer,
    "referenceNumber": string
    "referenceQualifier": string
    "totalAmount": 10964
  }],
  "hasSpecialConditions": boolean
},
"errors": [{
  "ibeCode": string,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }
}

```

```

    }
  ]
}
],
"success": boolean
}

```

fullReservationFlight object

```

{
  "aircraft": string,
  "referenceNumber": string,
  "origin": string,
  "destination": string,
  "departureDate": datetime,
  "arrivalDate": datetime,
  "flightNumber": string,
  "classOfService": string,
  "operatingCarrier": string,
  "marketingCarrier": string
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
number	unique booking number (PNR).
reservationDate	Booking creation date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T15:44:00
passengers	Array of passengerData objects with information about passengers

	services	<p>Array of objects with information about auxiliary services.</p> <p>Gets populated in case of api/services service usage. Will be described below.</p>
	seats	<p>Array of objects with information about selected seats.</p> <p>Gets populated in case of api/services service usage. Will be described below.</p>
	insurance	<p>Object with information about selected insurance.</p> <p>Gets populated in case of api/services service usage. Will be described below.</p>
	partnerProducts	<p>Array of objects with information about partner's services.</p> <p>Gets populated in case of api/services service usage. Will be described below.</p>
	journey	Information about flights for present booking.
	outboundFlights	Array of fullReservationFlight objects with detailed information about outbound flights.
	outboundFareFamily	Services field, contains fareFamily code.
	returnFlights	Array of fullReservationFlight objects with detailed information about return flights.
	origin	Origin airport
	destination	Destination airport
	departureDate	Departure date
	returnDate	Return date
	ticketFares	Array of objects with information about tariffs after discount applied.
	baseAmount	Base tariff amount

		passengerReferences	Array of objects with references to passengers
		referenceNumber	Reference number
		referenceQualifier	Reference qualifier, for example: "PA"
		passengerType	Passenger type. See Attachment 2, passenger types
		referenceNumber	Reference number
		referenceQualifier	Reference qualifier for example: "PA"
		totalAmount	Total tariff cost
		ticketFaresWithoutZapOff	Array of objects with information about tariffs without discount.
		baseAmount	Base tariff amount
		passengerReferences	Array of objects with references to passengers
		referenceNumber	Reference number
		referenceQualifier	Reference qualifier, for example: "PA"
		passengerType	Passenger type. See Attachment 2, passenger types
		referenceNumber	Reference number
		referenceQualifier	Reference qualifier for example: "TST"
		totalAmount	Total tariff cost
		errors	Information about errors appeared during query execution.
		ibeCode	Error code, see EN_Attachment 1. IBE error codes
		description	Code description, see EN_Attachment 1. IBE error codes
		messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
		code	Amadeus error code
		text	Amadeus error text

fullReservationFlight

Parameter	Description
aircraft	Aircraft number
referenceNumber	Reference to current flight segment
origin	Current segment's origin airport
destination	Current segment's destination airport
departureDate	Current segment's departure date and time
arrivalDate	Current segment's arrival date and time
flightNumber	Flight number
classOfService	Information about selected tariff
marketingCarrier	Marketing carrier Example: U6, S7
operatingCarrier	Operatin carrier Example: U6, OK
flightDuration	Flight duration

passengerData Response object

Parameter	Description
id	Passenger's index number (starting from 1)
hasInfant	Flag identifying whether passenger travels with infant
referenceNumber	Reference to present passenger in booking
firstName	Passenger' first name
middleName	Passenger's middle name (might be empty)
surname	Passenger's surname

passengerType	Passengers type Possible values: see Attachment 2, passenger types
isAdult	Flag identifying whether passenger is adult person true: for ADT/SRC/YTH types of passengers false: for CHD/INF types of passengers
birthDate	Date of birth
sex	Passenger's sex Possible values: see Attachment 2, passenger sex
nationality	Passenger's nationality Possible values: see Attachment 2, Countries
documentNumber	Document number
documentType	Document type Possible values: see Attachment 2, document types
documentExpiryDate	Document expiration date Mandatory only for international passport. For other type of documents automatically set to +10 years from present date.
email	Email
phone	Phone

Successful request and response examples

See [Attachment X. Request and response examples](#)

Validation rules for subsidised tariffs booking creation

No	Validation rule	Error code	Booking error description
1	Subsidise tariff is available only for SRC / YTH / CNN type of passengers	1008	Only subsidized types of passengers (SRC/YTH/CNN) could be used

2	Unable to add to booking both subsidised and not subsidised type of passengers	1008	Subsidized passengers found. All passengers in request must have subsidies.
3	Subsidised tariff is only available for Russian citizens.	1008	Only passenger with Russian citizenship can use subsidized tariffs
4	Age limits: from 12 to 23 years for YTH type, 55+ for women and 60+ for men of SRC type.	1008	Passenger age is incorrect

Booking examples

1. One way trip booking, no transfers.

The screenshot displays a flight booking interface for a one-way trip from St. Petersburg (LED) to Moscow (DME) on flight U6-80. The flight duration is 1 hour and 25 minutes. The departure time is 18:05 and the arrival time is 19:30. The starting price is 2,209 RUB. Five fare options are shown:

Category	Price (RUB)	Additional Info
ПРОМО	2 209 P	2 217 P (crossed out)
ЭКОНОМ	2 694 P	2 717 P (crossed out), БИЗНЕС ВОЗМОЖНОСТИ
ПРЕМИУМ-ЭКОНОМ	15 391 P	15 807 P (crossed out)
БИЗНЕС-ЛАЙТ	18 815 P	19 337 P (crossed out), 2 места
БИЗНЕС	24 053 P	24 737 P (crossed out), 4 места

Take particular recommendation from data -> recommendations in api/flights response

<p>Recommendation:</p> <pre> { "outboundFlightGroupId": 0, "outboundFareFamilyId": 1, "outboundPrice": 2717.0, "outboundTaxes": 1922.0, "outboundSeatsLeft": 9, "outboundFareDetails": [{ "bookingCode": "W", "fareBasis": "WECOW", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 1922.00, "totalAmount": 2717.00 }] }], "discount": { </pre>	<p>Retrieve flight number from outboundFlightGroupId which needs to be added and retrieve relative information from data -> outboundFlights in api/flights response</p> <pre> { "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "LED", "departureTerminal": "1", "arrivalAirport": "DME", "departureDateTime": "2018-12-08T18:05:00", "arrivalDateTime": "2018-12-08T19:30:00", "aircraft": "Airbus A319", "flightDuration": "1:25", "flightNumber": "80", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-1547914441" }], "flightGroupDuration": "1:25", "hash": "-1643139679" } </pre>
---	---

<pre> "id": "1-1PSJTS5", "amount": 3, "discountType": 0, "usageType": 0 }, "totalPrice": 2694.0 } </pre>	<p>Inserting highlighted values for flights array we'll get:</p> <pre> "flights": { "outboundFlights": [{ "origin": "LED", "destination": "DME", "departureDate": "2018-12-08T18:05:00.000Z", "flightNumber": "80", "classOfService": "W" }], "returnFlights": null, "passengersCount": 1 } </pre> <p>Value of classOfService matches bookingCode from recommendation</p> <p>passengersCount depends on amount of passengers added to booking</p> <p>By outboundFareFamilyId field we define tariff number which needs to be added and take relative information from data -> fareFamilies in api/flights response</p> <pre> { "fareFamilyId": 1, "code": "ECOFLEXRU", </pre>
--	--

	<pre> "shortCode": "EC", "category": "ECOFLEX", "options": [...], "order": 2, "commercialFamily": "U6ECONOMY" } Inserting highlighted values for fares array we'll get: "fares": { "outboundFareFamily": { "category": "ECOFLEX", "code": "ECOFLEXRU" }, "returnFareFamily": null, "fares": [{ "fareBasis": "WECOW", "isBreakpoint": true }] } </pre>
--	---

2. Twi way trip booking creation, one transfer in return flight

Outbound direction

Екатеринбург SVX 2 ч. 35 мин. Москва DME
 18:40 — 19:15
 U6-266

от 3 636 ₺

ПРОМО 3 636 ₺ <small>ВШИ ВОЗМОЖНОСТИ</small>	ЭКОНОМ 3 736 ₺	ПРЕМИУМ-ЭКОНОМ 26 321 ₺	БИЗНЕС 25 881 ₺ <small>8 МЕСТ</small>
--	---------------------------------	--	---

Return direction

Москва DME 8 ч. 20 мин. Екатеринбург SVX
 10:10 — 20:30
 U6-79 1 пересадка, U6-624

от 9 803 ₺

ПРОМО 9 803 ₺ <small>7 МЕСТ</small>	ЭКОНОМ 12 603 ₺ <small>ВШИ ВОЗМОЖНОСТИ</small>	ПРЕМИУМ-ЭКОНОМ 38 203 ₺	БИЗНЕС 51 448 ₺ <small>⊘ недоступно</small>
---	--	--	---

<p>Recommendation:</p> <pre>{ "outboundFlightGroupId": 0, "outboundFareFamilyId": 0, "returnFlightGroupId": 6, "returnFareFamilyId": 1, "outboundPrice": 3636.0, "outboundTaxes": 2321.0, "returnPrice": 12603.0, "returnTaxes": 4318.0, "outboundSeatsLeft": 9, "returnSeatsLeft": 7, "outboundFareDetails": [{ "bookingCode": "O", "fareBasis": "OPROW", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 2321.00, "totalAmount": 3636.00</pre>	<p>Retrieve flight numbers from outboundFlightGroupId and returnFlightGroupId and retrieve related information from data -> outboundFlights and data -> returnFlights in api/flights respectively.</p> <p>Outbound flight:</p> <pre>{ "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "SVX", "arrivalAirport": "DME", "departureDateTime": "2018-12-01T18:40:00", "arrivalDateTime": "2018-12-01T19:15:00", "aircraft": "Airbus A320", "flightDuration": "2:35", "flightNumber": "266", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "1861162940"] }, "flightGroupDuration": "2:35", "hash": "-1144973042"</pre>
--	---

<pre> }] }], "returnFareDetails": [{ "bookingCode": "N", "fareBasis": "NECRT", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 1925.00, "totalAmount": 4260.00 }] }, { "bookingCode": "P", "fareBasis": "PECRT", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, </pre>	<pre> } Return flight: { "flightGroupId": 6, "flights": [{ "flightId": 0, "departureAirport": "DME", "arrivalAirport": "LED", "arrivalTerminal": "1", "departureDateTime": "2019-01-10T10:10:00", "arrivalDateTime": "2019-01-10T11:55:00", "aircraft": "Airbus A319", "flightDuration": "1:45", "stopDuration": "3:45", "flightNumber": "79", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-814761245" }, { "flightId": 1, "departureAirport": "LED", </pre>
---	--

<pre> "taxAmount": 2393.00, "totalAmount": 8343.00 }] }], "totalPrice": 16239.0 } </pre>	<pre> "departureTerminal": "1", "arrivalAirport": "SVX", "departureDateTime": "2019-01-10T15:40:00", "arrivalDateTime": "2019-01-10T20:30:00", "aircraft": "Airbus A319", "flightDuration": "2:50", "flightNumber": "624", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-375095535" }], "flightGroupDuration": "8:20", "hash": "327592200" } </pre> <p>Inserting highlighted values for flights array we'll get</p> <pre> { "flights": { "outboundFlights": [{ "origin": "SVX", "destination": "DME", </pre>
--	--

	<pre>"departureDate": "2018-12-01T18:40:00.000Z", "flightNumber": "266", "classOfService": "O" }], "returnFlights": [{ "origin": "DME", "destination": "LED", "departureDate": "2019-01-10T10:10:00.000Z", "flightNumber": "79", "classOfService": "N" }, { "origin": "LED", "destination": "SVX", "departureDate": "2019-01-10T15:40:00.000Z", "flightNumber": "624", "classOfService": "P" }], "passengersCount": 1 } }</pre>
--	---

	<p>Value of classOfService matches bookingCode from recommendation</p> <p>passengersCount depends on amount of passengers added to booking</p> <p>By outboundFareFamilyId and returnFareFamilyId fields we define tariff numbers which needs to be added and take relative information from data -> fareFamilies in api/flights response</p> <p>Inserting highlighted values for fares array we'll get:</p> <pre>"fares": { "outboundFareFamily": { "category": "ECOLIGHT", "code": "ECOLIGHTRU" }, "returnFareFamily": { "category": "ECOFLEX", "code": "ECOFLEXRU" }, "fares": [{ "fareBasis": "OPROW", "isBreakpoint": true }, { "fareBasis": "NECRT",</pre>
--	---

	<pre> "isBreakpoint": true }, { "fareBasis": "PECRT", "isBreakpoint": true }] } </pre> <p>fareBasis values we take from recommendation</p>
--	---

3. One way trip booking creation with one stop

The screenshot displays a flight booking interface for a one-stop trip from Vladivostok (VVO) to Ekaterinburg (SVX). The flight duration is 9 hours and 25 minutes, with a departure time of 09:50 and an arrival time of 14:15. The flight number is U6-9170, and there is one stop (1 посадка). The price starts from 16 880 P. Four fare options are presented:

Категория	Цена (P)	Дополнительно
ПРОМО	16 880 P	
ЭКОНОМ	17 880 P	
ПРЕМИУМ-ЭКОНОМ	53 445 P	Вашим возможностями
БИЗНЕС	56 945 P	2 места

<p>Recommendation:</p> <pre> { "outboundFlightGroupId": 0, "outboundFareFamilyId": 2, "outboundPrice": 53445.0, "outboundTaxes": 3445.0, "outboundSeatsLeft": 5, "outboundFareDetails": [{ "bookingCode": "Q", "fareBasis": "QFLOW", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 3445.00, "totalAmount": 53445.00 }] }], "totalPrice": 53445.0 } </pre>	<p>Retrieve flight number from outboundFlightGroupId which needs to be added and retrieve relative information from data -> outboundFlights in api/flights response</p> <pre> { "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "VVO", "arrivalAirport": "SVX", "departureDateTime": "2019-02-05T09:50:00", "arrivalDateTime": "2019-02-05T14:15:00", "aircraft": "Airbus A320", "techStops": [{ "arrivalDateTime": "2019-02-05T13:05:00", "departureDateTime": "2019-02-05T13:45:00", "airportCode": "OVB", "stopDuration": "0:40", "flightDurationUntilNextStop": "2:30" }] }, "flightDuration": "9:25", "flightDurationUntilNextStop": "6:15", "flightNumber": "9170", </pre>
---	---

	<pre>"marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "1138974543" }], "flightGroupDuration": "9:25", "hash": "-1673368260" } Inserting highlighted values for flights array we'll get: { "flights": { "outboundFlights": [{ "origin": "VVO", "destination": "SVX", "departureDate": "2019-02-05T09:50:00.000Z", "flightNumber": "9170", "classOfService": "Q" }], "returnFlights": null, "passengersCount": 1 }</pre>
--	--

	<pre>} Value of classOfService matches bookingCode from recommendation passengersCount depends on amount of passengers added to booking By outboundFareFamilyId field we define tariff number which needs to be added and take relative information from data -> fareFamilies in api/flights response { "fareFamilyId": 2, "code": "ECOPREMRU", "shortCode": "FL", "category": "ECOPREM", "options": [...], "order": 3, "commercialFamily": "U6ECONOMY" } Inserting highlighted values for fares array we'll get: "fares": { "outboundFareFamily": { "category": "ECOPREM", "code": "ECOPREMRU" }, "returnFareFamily": null,</pre>
--	--

	<pre> "fares": [{ "fareBasis": "QFLOW", "isBreakpoint": true }] </pre>
--	--

GET reservation method

Description: allows to retrieve information about already created booking

Request format: /api/apiVersion/reservation

Request examples:

/api/apiVersion/reservation?pnrNumber=AAEW5D&lastName=IVANOV

/api/apiVersion/reservation!?ticketNumber=2622471403682&lastName=PETROV

Request parameters

Parameter	Description
pnrNumber	PNR number for which data needs to be retrieved
ticketNumber	Issued ticket number Ticket number should contain 13 digits for example: 2622438214047 or 2622438214046.
lastName	Last name of one of passengers added to booking Parameter is passed in latin. IBE will return error in case last name is not exist in PNR.

Note: when querying information about booking by ticket number, ticketNumber and lastName fields should refer to one and the same passenger.

Response format

```
{
  "data": {
    "tickets": [{
      "number": string,
      "passengerReference": string,
      "isInfantTicket": boolean
    }
  ],
  "outboundFareFamily": fareFamilyObject,
  "returnFareFamily": fareFamilyObject,
  "tsmData": [{
    "passengerReference": string,
    "price": number,
    "referenceType": string,
    "referenceNumber": string
  }
  ],
  "appliedDiscounts": [{
    "flightReference": string,
    "amount": integer,
    "id": string,
    "type": string,
    "passengerReference": string,
    "serviceCode": string,
    "promoNumber": string
  }
  ],
  "baseFarePrices": [{
    "currency": string,
    "passengerReference": string,
    "isInfantFare": boolean,
    "totalPrice": number,
    "basePrice": number,
    "priceHidden": boolean
  }
  ],
  "foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string,
    "commercialFamily": string,
    "ru": string,
    "en": string,
    "zh": string
  }
  ]},
  "spentBonuses": {
    "amount": integer,
    "passengerReference": string
  },
  "spentBonusesByPassengers": [{
    "amount": integer,
    "passengerReference": string
  }
  ]
}
```

```

    }
  ],
  "originOfficeId": string,
  "originOfficeType": string,
  "journey": {
    "outboundFlights": [fullReservationFlight],
    "outboundFareFamily": {
      "code": string
    },
    "returnFareFamily": {
      "code": string
    },
    "outboundDuration": string,
    "returnFlights": [fullReservationFlight],
    "returnDuration": string,
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "returnDate": datetime
  },
  "number": string,
  "reservationDate": datetime,
  "passengers": [passengerObject],
  "services": [{
    "code": string,
    "status": string,
    "text": string,
    "petType": string,
    "category": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string,
    "price": number
  }
],
  "seats": [{
    "number": integer,
    "title": string,
    "code": string,
    "status": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string
  }
],
  "insurance": {
    "number": string,
    "code": string,
    "status": string,
    "referenceNumber": string
  },
  "partnerProducts": [{
    "partnerId": integer,
    "partnerType": integer,
    "code": string,
    "amount": number,
    "currency": string,

```

```

        "passengerReference": string,
        "isInfantProduct": boolean
    }
],
"hasSpecialConditions": boolean
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ],
}
]
"success": boolean
}
}

```

fareFamily object

```

{
    "fareFamilyId": integer,
    "code": string,
    "options": [{
        "code": string,
        "category": string,
        "order": integer,
        "showInPopup": boolean,
        "showInSummary": boolean,
        "showInUpsell": boolean,
        "showInFareIncluded": boolean,
        "showForAdult": boolean,
        "showForInfant": boolean,
        "showInBookingCart": boolean
    }
    ],
    "order": integer,
    "commercialFamily": string,
    "canRefund": boolean,
    "hasUpsale": boolean
}

```

fullReservationFlight object

```

{
    "aircraft": string,
    "referenceNumber": string,
    "origin": string,
    "destination": string,
}

```

```

"departureDate": datetime,
"arrivalDate": datetime,
"flightNumber": string,
"classOfService": string,
"marketingCarrier": string,
"operatingCarrier": string,
"flightDuration": string,
"stopDuration": string
}

```

passenger object

```

{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": integer,
  "isAdult": boolean,
  "birthDate": datetime,
  "sex": integer,
  "nationality": string,
  "documentNumber": string,
  "documentExpiryDate": datetime,
  "email": string,
  "phone": string,
  "documentType": integer
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
tickets	Information about issued tickets
number	Ticket number, example: 262 2439494143
passengerReference	Reference to present ticket's passenger
isInfantTicket	Flag identifying whether present ticket is for infant.

	outboundFareFamily	Array of fareFamily objects with information about selected tariffs for outbound flight.
	returnFareFamily	Array of fareFamily objects with information about selected tariffs for return flight.
	tsmData	Array of objects with information about auxiliary services added to PNR.
	passengerReference	Reference to passenger to whom present service is provided
	price	Service price
	referenceType	PNR record reference type, in general has value «SSR».
	referenceNumber	SSR number. Each service is represented by unique ssr number in PNR.
	appliedDiscounts	Applied discounts in present PNR.
	flightReference	Reference to flight where discount was applied
	amount	Discount amount in percents
	id	CRM campaign ID
	type	Discount type. Possible values: DISCOUNT – conventional discount, PROMO – promocode discount.
	passengerReference	Reference to passenger for whom discount was applied.
	serviceCode	Code of the added service for which the discount was applied.
	promoNumber	The number of promo code from CRM.
	baseFarePrices	Base tariff costs information array.
	currency	currency in which cost is represented Example: RUB
	passengerReference	Reference to passenger for whom discount was applied.
	isInfantFare	Flag identifying whether passenger is infant.
	totalPrice	Total price
	basePrice	Base price (without tariffs and fees)

	priceHidden	Flag identifying whether tariff price should be hidden from client. Will be set to "true" for IT-tariffs.
	foodOnBoards	Array of objects with information about meals available on the flight
	departureAirport	Airport of departure
	arrivalAirport	Airport of arrival
	flightNumber	Flight number
	commercialFamily	Fare family type (different type of meals for economy and business can be available for one flight)
	ru	Option's translation to russian
	en	Option's translation to english
	zh	Options translation to chinese
	spentBonuses	Object with information about overall amount of spent bonuses
	amount	Amount of bonuses
	passengerReference	Reference to first passenger
	spentBonusesByPassengers	Array of objects with information about bonuses spent by particular passenger
	amount	Amount of bonuses
	passengerReference	Reference to passenger
	originOfficeId	Office id used for booking creation Example, SVXU608AZ
	originOfficeType	Office type used for booking creation Possible values: <u>Attachment 2, Amadeus office types</u>
	journey	Information about current booking's fights
	outboundFlights	Array of fullReservationFlight objects with detailed information about outbound flights
	outboundFareFamily	Service field with information about selected tariff for outbound flight. Example: ECOFLEXRU

	returnFareFamily	Service field with information about selected tariff for return flight
	outboundDuration	Total outbound flight duration
	returnFlights	Array of fullReservationFlight objects with detailed information about outbound flights
	returnDuration	Total return flight duration
	origin	Origin airport
	destination	Destination airport
	departureDate	Departure date
	returnDate	Return date
	number	PNR number
	reservationDate	Booking creation date and time
	passengers	Array of passengerData objects with information about passengers
	services	Array of objects with information about auxiliary services added to booking
	code	Auxiliary service code Possible values: see Attachment 2, Auxiliary services codes
	status	Service status Example: HK – service is successfully added
	text	Text in SSR format to send to Amadeus. Example: DOG 1PC 8KG 45X35X25
	petType	Type of the pet. Possible values: DOG, CAT, PARROT, CANARY
	category	Auxiliary service category Possible values: see Attachment 2, Auxiliary services codes
	passengerReference	Reference to passenger to whom present service is provided
	flightReference	Reference to segment on which service is provided

	referenceNumber	SSR number. Each service is represented by unique ssr number in PNR.
	price	Service price
	seats	Array of objects with information about seats added to booking
	number	seat row
	title	seat place (letter)
	code	"select seat" service code. Value: RQST
	status	Service status Example: HK – service is successfully added
	passengerReference	Reference to passenger to whom present service is provided
	flightReference	Reference to segment on which service is provided
	referenceNumber	SSR number. Each service is represented by unique ssr number in PNR.
	insurance	Object with information about insurance added to booking.
	number	Insurance document number
	code	Insurance code Possible values: see Attachment 2, auxiliary services codes
	status	Service status Example: HK – service is successfully added
	referenceNumber	SSR number. Each service is represented by unique ssr number in PNR.
	partnerProducts	Array of objects with information about partner's services added to booking
	partnerId	Partner's unique ID in IBE Possible values: see EN Attachment 4. Partner's codes
	partnerType	Partner's type in IBE Possible values: see EN Attachment 4. Partner's types

	code	Code of partner's service Each partner has its unique codes, see Possible values: see EN Attachment 4. Partner's services codes
	amount	Partner's service price
	currency	Price currency
	passengerReference	Reference to passenger to whom present service is provided
	isInfantProduct	Flag identifying that service is provided to an infant
	hasSpecialConditions	Sign of that one of the following services was added to booking: PETC, CBBG, WEAP, UMNr, MAAS, MEDA, EXST, STCR, WCHR, WCHC, WCHS, WCBD, WCBW, WCMP, WCOB, DEAF, BLND, DEPA, DEPU
errors		Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode
	code	Amadeus error code
	text	Amadeus error text

fareFamily

Parameter	Description
fareFamilyId	Fare family index; used in recommendations.
code	Fare family code, example: ECOLIGHT/ECOFLEX/ECOPREM etc. Possible values: see Attachment 2, Main fare family categories
order	Fare family display index
commercialFamily	fare family type for ex: U6ECONOMY or U6BUSINESS
canRefund	Sign of refundable tariff

hasUpsale	<p>Sign indentifying whether present tariff can be upgraded to next one in price category.</p> <p>Used for management of Upsale block display in frontend.</p>
options	<p>Array of fareFamiliesOptions objects with information about tariff options.</p> <p>Possible values: see Attachment 2, tariff options</p>
code	<p>Tariff option code</p> <p>Example:</p> <p>upTo5bonuses</p> <p>refundExchangeProhibited</p> <p>mealsOnBoard</p> <p>etc.</p> <p>used in IBE website</p>
category	<p>Код Tariff option category</p> <p>Example:</p> <p>bonus, ffServices etc.</p> <p>used in IBE website</p>
order	Option display index
showInPopup	flag determining whether option should be displayed in "your options" pop-up
showInSummary	flag determining whether option should be displayed on Summary page.
showInUpsell	flag determining whether option should be displayed in tariff upgrade information block
showInFareIncluded	flag determining whether option should be displayed when filter is selected
showForAdult	flag determining whether option should be displayed for adult passenger.
showForInfant	flag determining whether option should be displayed for infant.

showInBookingCart	flag determining whether option should be displayed in cart
-------------------	---

fullReservationFlight

Parameter	Description
aircraft	Aircraft number
referenceNumber	Reference to current flight segment
origin	Current segment origin airport
destination	Current segment destination airport
departureDate	Current segment departure date and time
arrivalDate	Current segment arrival date and time
flightNumber	Flight number
classOfService	Information about selected tariff
marketingCarrier	Marketing carrier Example: U6, S7
operatingCarrier	Operating carrier Example: U6, OK
flightDuration	Current segment flight duration

passengerData

Parameter	Description
id	Passenger's index number (starting from 1)
hasInfant	Flag identifying whether passenger travels with infant
referenceNumber	Reference to present passenger in booking
firstName	Passenger's first name
middleName	Passenger's middle name (might be empty)
surname	Passenger's surname

passengerType	Passengers type Possible values: see Attachment 2, passenger types
isAdult	Flag identifying whether passenger is adult person true: for ADT/SRC/YTH types of passengers false: for CHD/INF types of passengers
birthDate	Date of birth
sex	Passenger's sex Possible values: see Attachment 2, passenger sex
nationality	Passenger's nationality Possible values: see Attachment 2, Countries
documentNumber	Document number
documentType	Document type Possible values: see Attachment 2, document types
documentExpiryDate	Document expiration date Mandatory only for international passport. For other type of documents automatically set to +10 years from present date.
loyaltyCard	"Wings" loyalty card number
email	Email
phone	Phone

Successful request and response examples

See Attachment X. Request and response examples

Failed request and response examples

Invalid PNR used in query

```
{
```

```
"errors": [{
  "ibeCode": 1101,
  "description": "Amadeus connectivity error.",
  "messages": []
}]
"success": false
}
```

Last name used in query does not exist in PNR

```
{
  "errors": [{
    "ibeCode": 1111,
    "description": "Amadeus PNR_Retrieve function returned error.",
    "messages": []
  }]
  "success": false
}
```

Last name used does not match one presented in retrieved ticket

```
{
  "errors": [{
    "ibeCode": 1111,
    "description": "Amadeus PNR_Retrieve function returned error.",
    "messages": []
  }]
  "success": false
}
```

Invalid ticket number used in query

```
{
  "errors": [{
    "ibeCode": 1008,
    "description": "Data validation error.",
    "messages": [{
      "text": "'Ticket Number' is not in the correct format."
    }]
  }]
  "success": false
}
```

DELETE reservation method

Description: method allows to delete created booking

Request format /api/apiVersion/reservation

Request parameters

Absent. Booking for cancellation is identified by session ID passed in X-Session header.

Response format

```
{
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ],
}
]
"success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Failed request and response examples

Delete attempt when booking is not loaded in context:

```
{
  "errors": [
    {
      "ibeCode": 1001,
      "description": "Operation can not be performed due to invalid
reservation state.",
      "messages": []
    }
  ],
  "success": false
}
```

Booking management service v2.2

Object fullReservationFlightСервис API	reservation
Short description	The service allows to manage booking: - add information to the booking (data on flights, passengers and fares); - receive information about the reservation; - delete a booking.
Supported HTTP(S)-methods	GET, POST, DELETE
Used methods of related systems	Amadeus: Create/update booking - PNR_AddMultiElements, Fare_PricePNRWithBookingClass, Ticket_CreateTSTFromPricing Getting information on booking - PNR_Retrieve, Ticket_RetrieveListOfTSM Deleting a booking - PNR_Retrieve, PNR_Cancel, PNR_AddMultiElements

Method POST reservation

Description: creating a booking, adding information about flights, passengers and fares.

There are 2 options for creating a booking:

1. Advance booking, when the tariff with the selected price is assigned to the user's session (the Amadeus session goes into the Stateful state, other users of the system cannot receive it).
2. Regular booking, when the tariff with the selected price is not assigned to the user, since the number of available Stateful sessions has exceeded a certain preset value. In this case, a conflict is possible when customers buy the last ticket at a certain rate: the ticket will be displayed for all users, but direct booking will be made only for the customer who sends the passenger data earlier than others. Other customers may receive an error stating that their chosen plan is no longer available.

The booking engine website currently uses the following booking creation algorithm:

1. Send a request to create an advance booking (only flight data and number of passengers).
2. If the answer is successful, it means that a preliminary booking has been created, and in the next step, you will need to add data about passengers and fares to the booking.
3. If you receive an error with the code 1003 in response, it means that the preliminary booking has not been created and at the next step of booking, you will need to add full information about the flight and passengers.

Request format /api/apiVersion/reservation

Request body

```
{
  "flights": {
    "outboundFlights": [reservationFlight],
    "returnFlights": [reservationFlight],
    "adults": integer,
    "children": integer,
    "infants": integer,
    "pensioners": integer,
    "youths": integer,
    "subsidizedChildren": integer,
    "origin": string,
    "destination": string,
    "passengersCount": integer
  },
  "passengers": {
    "passengers": [passengerData],
    "userAgreement": boolean
  },
  "fares": {
    "accountCode": "string",
    "tourCode": "string",
    "outboundFareFamily": {
      "code": string,
      "category": string,

```

```

        "isSubsidized": boolean
    },
    "returnFareFamily": {
        "code": string,
        "category": string,
        "isSubsidized": boolean
    },
    "fares": [{
        "passengerType": "Adult",
        "flightIndex": integer,
        "flightDirection": integer,
        "fareBasis": string,
        "isBreakpoint": boolean
    }
    ],
    "sessionMarker": string,
    "currency": string
},
"expectedPrice": integer,
"searchPrice": integer,
"airShopSessionId": string
}

```

reservationFlight object

```

{
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "flightNumber": string,
    "classOfService": string
}

```

passengerData object

```

{
    "id": integer,
    "firstName": string,
    "middleName": string,
    "surname": string,
    "passengerType": integer,
    "birthDate": datetime,
    "sex": integer,
    "nationality": string,
    "documentNumber": string,
    "documentExpiryDate": datetime,
    "loyaltyCard": string,
    "loyaltyCardValidTo": datetime,
    "serviceClassUpOrdered": boolean,
    "email": string,
}

```

```

"phone": string,
"documentType": integer,
"disability": {
  "isDeaf": boolean,
  "isBlind": boolean,
  "hasGuideDog": boolean,
  "assistanceType": integer,
  "wheelchairType": integer,
  "commentary": string
},
"endorsements": [{
  "endorsement": boolean,
  "isDisability": boolean,
  "endorsementDocNumber": string,
  "endorsementDocDate": datetime,
  "loyaltyPartnerCard": string
}
]
}

```

Request parameters

Parameter	Description
flights	Object with information about flights
outboundFlights	Array of reservationFlight objects with information about flights in the "outbound" direction
returnFlights	Array of reservationFlight objects with information about flights in the "return" direction
adults, children, infants, pensioners, youths, subsidizedChildren	Count of passengers of each type
origin	Departure airport
destination	Destination airport
passengersCount	Total count of passengers
passengers	Object with information about passengers
passengers	An array of passengerData objects with passengers information
userAgreement	Sign of sending consent to the newsletter Consent is sent to CRM for all added passengers
fares	Object with information about tariffs

outboundFareFamily		Object with information on fares for "there" direction
	code	Fare family code Data is taken from the response api/flights: data.fareFamilies.code
	category	Fare family category Data is taken from the response api/flights: data.fareFamilies.category
	isSubsidized	Sign of a subsidized tariff
returnFareFamily		Object with information about fare family tariffs for return flight
	code	Fare family code Data is taken from the response api/flights: data.fareFamilies.code
	category	Fare family category Data is taken from the response api/flights: data.fareFamilies.category
	isSubsidized	Sign of a subsidized tariff
fares		An array with information about tariffs for directions "outbound" and "return"
	accountCode	Account Code
	tourCode	Tour Code
	fareBasis	Base tariff title Data is taken from the response api/flights: data.recommendations.outboundFareDetails.fareBasis and data.recommendations.returnFareDetails.fareBasis
	isBreakpoint	Flag identifying whether tariff ends on present segment. Data is taken from the response api/flights: data.recommendations.outboundFareDetails.isBreakpoint and data.recommendations.returnFareDetails.isBreakpoint

	passengerType	Passenger type Possible values: see Attachment 2. Passenger types
	flightDirection	Flight direction 0 - outbound, 1 - return
	flightIndex	Flight index for a specific flight
	sessionMarker	Aviasales session marker
	currency	Reservation currency
	expectedPrice	The expected price of the booking, taken from the api/flights
	searchPrice	The price that was received in the search request, taken from the api/flights or api/flights/search
	airShopSessionId	SIG search session ID, taken from the api/flights response

reservationFlight

Parameter	Description
origin	Departure airport
destination	Destination airport
departureDate	Date and time of departure The data is taken from the api/flights response: data.outboundFlights.flights.departureDateTime or data.returnFlights.flights.departureDateTime Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T01:00:00

flightNumber	<p>Flight number.</p> <p>The data is taken from the api/flights response:</p> <p>data.outboundFlights.flights.flightNumber</p> <p>or</p> <p>data.returnFlights.flights.flightNumber</p>
classOfService	<p>Information about the selected tariff</p> <p>The data is taken from the api/flights response:</p> <p>data.recommendations.outboundFareDetails.bookingCode</p> <p>or</p> <p>data.recommendations.returnFareDetails.bookingCode</p>

passengerData Request object

Parameter	Description	Format
id	Passenger's serial number (numbered from 1)	
firstName	Passenger's name	[A-Z], 2-20 chars
middleName	Passenger's patronymic (may be absent)	[A-Z], 2-20 chars
surname	Passenger's surname	[A-Z], 2-20 chars
passengerType	<p>Passenger type</p> <p>Possible values: see Attachment 2. Passenger types</p>	
birthDate	Date of birth	

sex	Passenger gender Possible values: see dictionary "gender of passengers"	
nationality	Nationality (country) Possible values: dictionary of countries	
documentNumber	Document number	
documentType	Document type Possible values: dictionary of document types	
documentExpiryDate	Date until which the document is valid Mandatory only for a foreign passport. For other documents, +10 years from the current date is automatically set	
loyaltyCard	Loyalty card number "Wings"	
loyaltyCardValidTo	Expiration date of the "Wings" loyalty card (transmitted only if serviceClassUpOrdered = true)	
serviceClassUpOrdered	Flag indicating that the owner of the "Wings" gold card wants to upgrade the service class	
email	Email	
phone	Phone	
disability	Information about people with disabilities	
isDeaf	Hearing impairment	
isBlind	Visual impairment	
hasGuideDog	A guide dog must be accompanied, must be sent only together with the isBlind sign	

	assistanceType	Passenger assistance type Possible values: dictionary of assistance types	
	wheelchairType	Wheelchair type Possible values: dictionary of wheelchair type	
	commentary	Comment (optional)	
endorsements		Array with information about documents to buy tickets on the subsidized fares or to apply discounts on partner loyalty cards.	
	endorsement	The sign that an endorsement is adding to booking.	
	isDisability	Disability endorsements.	
	endorsementDocNumber	Subsidy document number. E.g disability certificate, ID number or other document confirming the status of a large family.	
	endorsementDocDate	Subsidy document issue date.	
	loyaltyPartnerCard	Loyalty partner card number.	

Response format

```
{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "returnFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string,
      },
      "returnFareFamily": {
        "code": string,
      },
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    },
    "number": string,
    "reservationDate": datetime,
  }
}
```



```

"passengers": [{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": string,
  "birthDate": datetime,
  "sex": integer,
  "nationality": string,
  "documentNumber": string,
  "documentExpiryDate": datetime,
  "email": string,
  "phone": string,
  "documentType": integer,
  "isAdult": boolean,
  "isChanged": boolean,
    "checkedIn": {
      flightReference: boolean
    },
    "checkInHasStarted": {
      flightReference: boolean
    },
    "hasPassengersDataErrors": boolean,
  "passengerDataErrors": [{
    "key": string,
    "value": string
  }]
}]
],
"services": [],
"seats": [],
"insurance": {},
"luggageSeats": [{
  "referenceNumber": string,
  "passengerReference": string
}]
],
  "partnerProducts": [],
"ticketFares": [{
  "baseAmount": integer,
  "passengerReferences": [{
    "referenceNumber": string,
    "referenceQualifier": string
  }],
  "passengerType": integer,
  "referenceNumber": string,
  "referenceQualifier": string,
  "totalAmount": 10964,
  "currency": string
}],
"ticketFaresWithoutZapOff": [{
  "baseAmount": integer,
  "passengerReferences": [{
    "referenceNumber": string,
    "referenceQualifier": string
  }],

```

```

    }],
    "passengerType": integer,
    "referenceNumber": string,
    "referenceQualifier": string,
    "totalAmount": 10964,
        "currency": string
    ]],
    "hasSpecialConditions": boolean,
    "isRefundAvailable": boolean,
    "annulUnavailabilityReason": integer,
    "isAnnulAvailable": boolean
},
"errors": [{
    "ibeCode": string,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ]
}
],
"success": boolean
}

```

fullReservationFlight object

```

{
    "aircraft": string,
    "aircraftIataCode": string,
    "referenceNumber": string,
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "arrivalDate": datetime,
    "arrivalDateUtc": datetime,
    "departureDateUtc": datetime,
    "flightNumber": string,
    "classOfService": string,
    "marketingCarrier": string,
    "operatingCarrier": string,
    "flightDuration": string,
    "stopDuration": string,
    "techStops": [{
        "arrivalDateTime": datetime,
        "departureDateTime": datetime,
        "airportCode": string,
        "stopDuration": string,
        "flightDurationUntilNextStop": string
    }
    ],
    "flightDurationUntilNextStop": string,
    "arrivalTerminal": string,
    "departureTerminal": string,

```

```

    "hash": string,
    "isCancelledByAirline": boolean
}

```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true.
data	The data returned by the request.
number	Unique booking number (PNR).
reservationDate	Date and time of booking creation Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T15:44:00
passengers	An array of the passengerData objects with passenger information
services	An array of objects with information about additional services. Filled in when using the api/services service and will be described below
seats	An array of objects with information about the selected locations. Filled in when using the api/services service and will be described below
insurance	Object with information about the selected insurance. Filled in when using the api/services service and will be described below
luggageSeats	An array of objects with information about luggage with separate air ticket and ssr CBBG.
referenceNumber	The reference to CBBG luggage.
passengerReference	The reference to passenger who added CBBG luggage. Associated with field data → passengers[n] → referenceNumber.
partnerProducts	An array of objects with information about partner services. Filled in when using the api/services service and will be described below

Parameter		Description
	journey	Flight information for current booking
	outboundFlights	An array of the fullReservationFlight objects with details of outbound flights
	outboundFareFamily	Service field, contains the fareFamily code
	returnFlights	An array of fullReservationFlight objects with detailed information about the "return" flights
	origin	Initial airport of departure
	destination	Initial airport of destination
	departureDate	Departure date
	returnDate	Return date
	ticketFares	An array of objects with information about tariffs after applying the discount
	baseAmount	Base rate
	passengerReferences	An array of objects with links to passengers
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "PA"
	passengerType	Passenger type Possible values: see Attachment 2. Passenger types
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "TST"
	totalAmount	The total amount of the tariff
	currency	Reservation currency
	ticketFaresWithoutZapOff	An array of objects with information about tariffs without applying a discount
	baseAmount	Base rate
	passengerReferences	An array of objects with links to passengers
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "PA"

Parameter		Description
	passengerType	Passenger type Possible values: see Attachment 2. Passenger types
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "TST"
	totalAmount	The total amount of the tariff
	currency	Reservation currency
	hasSpecialConditions	Sign of the additional conditions
	isRefundAvailable	Sign of the availability of the return
	annulUnavailabilityReason	Reason for unavailability of booking cancellation
	isAnnulAvailable	Indication of availability of booking cancellation
errors		Information about errors that occurred during the operation of the service
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from Amadeus' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
	code	Amadeus error code
	text	Amadeus error text

fullReservationFlight

Parameter	Description
aircraftIataCode	IATA aircraft code
referenceNumber	Reference to the current flight segment
origin	Departure airport for the current segment
destination	Destination airport for the current segment
departureDate	Departure date and time for the current segment
arrivalDate	Arrival date and time for the current segment

Parameter	Description
arrivalDateUtc	Date and time of arrival for the current segment in UTC
departureDateUtc	Departure date and time for the current segment in UTC
flightNumber	Flight number
classOfService	Information about the selected tariff
marketingCarrier	Marketing airline For example: U6, S7
operatingCarrier	The airline actually performing the flight For example: U6, OK
techStops	An array with information about technical landings
arrivalDateTime	Date and time of arrival at the pick-up point
departureDateTime	Date and time of departure from the pick-up point
airportCode	Airport code of the stop
stopDuration	Duration of the stop
flightDurationUntilNextStop	Duration of the flight to the next stop
flightDurationUntilNextStop	Duration of the flight to the next stop
arrivalTerminal	Terminal number at the point of arrival
departureTerminal	Terminal number at the point of departure
hash	Unique hash of the flight
isCancelledByAirline	Sign of cancellation of flights by the airline

passengerData Response object

Parameter	Description
id	Passenger's serial number (numbered from 1)
hasInfant	An sign that this passenger has a baby
referenceNumber	Reference to this passenger in the booking
firstName	Passenger's name

	Parameter	Description
	middleName	Passenger's patronymic (may be absent)
	surname	Passenger's surname
	passengerType	Passenger type Possible values: see Attachment 2. Passenger types
	isAdult	Whether the passenger is an adult true for ADT/SRC/YTH type passengers false for CHD/INF type passengers
	birthDate	Date of birth
	sex	Passenger gender Possible values: see dictionary "gender of passengers"
	nationality	Nationality (country) Possible values: dictionary of countries
	documentNumber	Document number
	documentType	Document type Possible values: dictionary of document types
	documentExpiryDate	Date until which the document is valid Mandatory only for a foreign passport. For other documents, +10 years from the current date is automatically set
	email	Email
	phone	Phone
	checkedIn	Passenger check-in on flights
	checkInHasStarted	Starting check-in on flights
	hasPassengersDataErrors	The sign of errors in passengers data.
	passengerDataErrors	Array with detailed information of errors in passengers data.
	key	The key.
	value	The value.

Examples of successful requests and responses

See the Attachment X. Examples of responses from services.

Validation rules for creating a booking for subsidized fares

№	Validation rule	Error code	Description of booking error
1	Subsidized fare is only available for bookings for SRC / YTH / CNN passenger types	1008	Only subsidized types of passengers (SRC/YTH/CNN) could be used
2	You cannot add both subsidized and regular passenger types to your booking.	1008	Subsidized passengers found. All passengers in request must have subsidies.
3	Subsidized rate can be booked only for Russian citizens	1008	Only passenger with Russian citizenship can use subsidized tariffs
4	Age restrictions: from 12 to 23 years old for YTH type, 55 years old for women and 60 years old for men for SRC type.	1008	Passenger age is incorrect

Examples of booking

1. *Creation of one-way booking, direct.*



Take a specific recommendation from the api / flights response: data -> recommendations:

<p>Recommendation:</p> <pre>{ "outboundFlightGroupId": 0, "outboundFareFamilyId": 1, "outboundPrice": 2717.0, "outboundTaxes": 1922.0, "outboundSeatsLeft": 9, "outboundFareDetails": [{ "bookingCode": "W", "fareBasis": "WECOW", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 1922.00, "totalAmount": 2717.00 }] }</pre>	<p>Using the outboundFlightGroupId field, determine the flight number that needs to be added and take information on it from the api/flights response, data -> outboundFlights:</p> <pre>{ "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "LED", "departureTerminal": "1", "arrivalAirport": "DME", "departureDateTime": "2018-12-08T18:05:00", "arrivalDateTime": "2018-12-08T19:30:00", "aircraft": "Airbus A319", "flightDuration": "1:25", "flightNumber": "80", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-1547914441" }], "flightGroupDuration": "1:25", "hash": "-1643139679" }</pre>
--	---

<pre> "id": "1-1PSJTS5", "amount": 3, "discountType": 0, "usageType": 0 }, "totalPrice": 2694.0 } </pre>	<p>Substitute the highlighted values, for the flights array and get:</p> <pre> "flights": { "outboundFlights": [{ "origin": "LED", "destination": "DME", "departureDate": "2018-12-08T18:05:00.000Z", "flightNumber": "80", "classOfService": "W" }], "returnFlights": null, "passengersCount": 1 } </pre> <p>The classOfService value matches the bookingCode from the recommendation.</p> <p>The passengersCount value depends on the number of passengers added to the booking.</p> <p>Using the outboundFareFamilyId field, determine the number of the fare that will need to be added and take information on it from the api/flights response, data -> fareFamilies:</p> <pre> { "fareFamilyId": 1, </pre>
--	--

```
"code": "ECOFLEXRU",  
"shortCode": "EC",  
"category": "ECOFLEX",  
"options": [ ... ],  
"order": 2,  
"commercialFamily": "U6ECONOMY"  
}
```

Substitute the highlighted values, for the fares array and get:

```
"fares": {  
  "outboundFareFamily": {  
    "category": "ECOFLEX",  
    "code": "ECOFLEXRU"  
  },  
  "returnFareFamily": null,  
  "fares": [{  
    "fareBasis": "WECOW",  
    "isBreakpoint": true  
  }  
]  
}
```

2. *Creation of a booking in both directions, one transfer on the “return” segment.*

Direction outbound



Direction return



<p>Recommendation:</p> <pre> { "outboundFlightGroupId": 0, "outboundFareFamilyId": 0, "returnFlightGroupId": 6, "returnFareFamilyId": 1, "outboundPrice": 3636.0, "outboundTaxes": 2321.0, "returnPrice": 12603.0, "returnTaxes": 4318.0, "outboundSeatsLeft": 9, "returnSeatsLeft": 7, "outboundFareDetails": [{ "bookingCode": "O", "fareBasis": "OPROW", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 2321.00, "totalAmount": 3636.00 }] } </pre>	<p>Using the outboundFlightGroupId and returnFlightGroupId fields, we determine the flight numbers and take information on them from the api/flights response, data -> outboundFlights and data -> returnFlights, respectively.</p> <p>Flight outbound:</p> <pre> { "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "SVX", "arrivalAirport": "DME", "departureDateTime": "2018-12-01T18:40:00", "arrivalDateTime": "2018-12-01T19:15:00", "aircraft": "Airbus A320", "flightDuration": "2:35", "flightNumber": "266", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "1861162940" }], </pre>
--	---

<pre>] }], "returnFareDetails": [{ "bookingCode": "N", "fareBasis": "NECRT", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 1925.00, "totalAmount": 4260.00 }] }], { "bookingCode": "P", "fareBasis": "PECRT", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 2393.00, </pre>	<pre> "flightGroupDuration": "2:35", "hash": "-1144973042" } Flight return: { "flightGroupId": 6, "flights": [{ "flightId": 0, "departureAirport": "DME", "arrivalAirport": "LED", "arrivalTerminal": "1", "departureDateTime": "2019-01-10T10:10:00", "arrivalDateTime": "2019-01-10T11:55:00", "aircraft": "Airbus A319", "flightDuration": "1:45", "stopDuration": "3:45", "flightNumber": "79", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-814761245" </pre>
--	---

<pre> "totalAmount": 8343.00 }] }], "totalPrice": 16239.0 } </pre>	<pre> }, { "flightId": 1, "departureAirport": "LED", "departureTerminal": "1", "arrivalAirport": "SVX", "departureDateTime": "2019-01-10T15:40:00", "arrivalDateTime": "2019-01-10T20:30:00", "aircraft": "Airbus A319", "flightDuration": "2:50", "flightNumber": "624", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-375095535" }], "flightGroupDuration": "8:20", "hash": "327592200" } </pre> <p>Substitute the highlighted values, for the flights array and get:</p> <pre> { "flights": { </pre>
--	---

	<pre>"outboundFlights": [{ "origin": "SVX", "destination": "DME", "departureDate": "2018-12- 01T18:40:00.000Z", "flightNumber": "266", "classOfService": "O" }], "returnFlights": [{ "origin": "DME", "destination": "LED", "departureDate": "2019-01- 10T10:10:00.000Z", "flightNumber": "79", "classOfService": "N" }, { "origin": "LED", "destination": "SVX", "departureDate": "2019-01- 10T15:40:00.000Z", "flightNumber": "624", "classOfService": "P" } }</pre>
--	---


```
],  
  
  "passengersCount": 1  
  
}  
  
}
```

The classOfService value matches the bookingCode from the recommendation.

The passengersCount value depends on the number of passengers added to the booking.

Using the outboundFareFamilyId and returnFareFamilyId fields, we determine the fare numbers that need to be added and take information on them from the api/flights response, data -> fareFamilies:

Substitute the highlighted values, for the flights fares and get:

```
"fares": {  
  
  "outboundFareFamily": {  
  
    "category": "ECOLIGHT",  
  
    "code": "ECOLIGHTRU"  
  
  },  
  
  "returnFareFamily": {  
  
    "category": "ECOFLEX",  
  
    "code": "ECOFLEXRU"  
  
  },  
  
  "fares": [{  
  
    "fareBasis": "OPROW",
```

	<pre> "isBreakpoint": true }, { "fareBasis": "NECRT", "isBreakpoint": true }, { "fareBasis": "PECRT", "isBreakpoint": true }] } </pre> <p>The fareBasis values are taken from the recommendation.</p>
--	--

Example of creating a booking for subsidized passengers

3. Creation of a booking for a set of passengers: 1 SRC + 1 YTH + 1 CNN (types of passengers are described in the Attachment 2).

Direction outbound



Direction return



Find a recommendation for the selected flight (in the api/flights response)



See that the array of fares for outbound and outbound directions contains information about the type of passenger for which the fare should be used, and about the flight segment number.

Note: it is the type of passenger that is used, not a link to it, since at this stage passengers have not yet been added to the booking.

Tariff information should be sent as follows:

```
"fares": {
  "outboundFareFamily": {
    "category": "SUBSIDIZED",
    "code": "SUBSIDIZED",
    "isSubsidized": true
  },
  "returnFareFamily": {
    "category": "ECOLIGHT",
    "code": "ECOLIGHTRU",
    "isSubsidized": true
  },
  "fares": [
    {
      "fareBasis": "RBDRTCD",
      "isBreakpoint": true,
      "passengerType": 1,
      "flightDirection": 0,
      "flightIndex": 0
    },
    {
      "fareBasis": "RBDRTZZ",
      "isBreakpoint": true,
      "passengerType": 2,
      "flightDirection": 0,
      "flightIndex": 0
    },
    {
      "fareBasis": "RBDRTCH",
      "isBreakpoint": true,
      "passengerType": 5,
      "flightDirection": 0,
      "flightIndex": 0
    },
    {
      "fareBasis": "EPROW",
      "isBreakpoint": true,
```

```

    "passengerType": 1,
    "flightDirection": 1,
    "flightIndex": 0
  },
  {
    "fareBasis": "EPRRT",
    "isBreakpoint": true,
    "passengerType": 1,
    "flightDirection": 1,
    "flightIndex": 1
  },
  {
    "fareBasis": "EPROW",
    "isBreakpoint": true,
    "passengerType": 2,
    "flightDirection": 1,
    "flightIndex": 0
  },
  {
    "fareBasis": "EPRRT",
    "isBreakpoint": true,
    "passengerType": 2,
    "flightDirection": 1,
    "flightIndex": 1
  },
  {
    "fareBasis": "EPROW",
    "isBreakpoint": true,
    "passengerType": 5,
    "flightDirection": 1,
    "flightIndex": 0
  },
  {
    "fareBasis": "EPRRT",
    "isBreakpoint": true,
    "passengerType": 5,
    "flightDirection": 1,
    "flightIndex": 1
  }
],
"sessionMarker": null
}

```

4. Creation of one way booking with one landing.



<p>Recommendation:</p> <pre> { "outboundFlightGroupId": 0, "outboundFareFamilyId": 2, "outboundPrice": 53445.0, "outboundTaxes": 3445.0, "outboundSeatsLeft": 5, "outboundFareDetails": [{ "bookingCode": "Q", "fareBasis": "QFLOW", "isBreakpoint": true, "fareAmounts": [{ "paxType": 0, "paxCount": 1, "taxAmount": 3445.00, "totalAmount": 53445.00 }] }, "totalPrice": 53445.0 } </pre>	<p>Using the outboundFlightGroupId field, determine the flight number that will need to be added and take information on it from the api/flights response, data -> outboundFlights:</p> <pre> { "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "VVO", "arrivalAirport": "SVX", "departureDateTime": "2019-02-05T09:50:00", "arrivalDateTime": "2019-02-05T14:15:00", "aircraft": "Airbus A320", "techStops": [{ "arrivalDateTime": "2019-02-05T13:05:00", "departureDateTime": "2019-02-05T13:45:00", "airportCode": "OVB", "stopDuration": "0:40", "flightDurationUntilNextStop": "2:30" }] }, "flightDuration": "9:25", "flightDurationUntilNextStop": "6:15", </pre>
---	---

```
"flightNumber": "9170",
```

```
"marketingCarrier": "U6",
```

```
"operatingCarrier": "U6",
```

```
"hash": "1138974543"
```

```
}
```

```
],
```

```
"flightGroupDuration": "9:25",
```

```
"hash": "-1673368260"
```

```
}
```

Substitute the highlighted values, for the flights array and get:

```
{
```

```
"flights": {
```

```
"outboundFlights": [{
```

```
"origin": "VVO",
```

```
"destination": "SVX",
```

```
"departureDate": "2019-02-05T09:50:00.000Z",
```

```
"flightNumber": "9170",
```

```
"classOfService": "Q"
```

```
}
```

```
],
```

```
"returnFlights": null,
```

	<pre>"passengersCount": 1 } }</pre> <p>The classOfService value matches the bookingCode from the recommendation.</p> <p>The passengersCount value depends on the number of passengers added to the booking.</p> <p>Using the outboundFareFamilyId field, we determine the fare number that needs to be added and take information on it from the api/flights response, data -> fareFamilies:</p> <pre>{ "fareFamilyId": 2, "code": "ECOPREMRU", "shortCode": "FL", "category": "ECOPREM", "options": [...], "order": 3, "commercialFamily": "U6ECONOMY" }</pre> <p>Substitute the highlighted values, for the fares array and get:</p> <pre>"fares": { "outboundFareFamily": { "category": "ECOPREM", "code": "ECOPREMRU"</pre>
--	---

	<pre> }, "returnFareFamily": null, "fares": [{ "fareBasis": "QFLOW", "isBreakpoint": true }] } </pre>
--	---

Method GET reservation

Description: allows to get information about a reservation that has already been created.

Request format /api/apiVersion/reservation

Examples of requests

/api/apiVersion/reservation?pnrNumber=AAEW5D&lastName=IVANOV

/api/apiVersion/reservation1?ticketNumber=2622471403682&lastName=PETROV

Request parameters

Parameter	Description
pnrNumber	PNR number for which information is needed
ticketNumber	Issued ticket number. The ticket number must be 13 digits, for example: 2622438214047 or 2622438214046.

Parameter	Description
lastName	Last name of one of the passengers added to the booking. The parameter is passed in Latin. IBE will return an error if the surname is not in the PNR.

Note: when requesting information about an order by ticket number, the ticketNumber and lastName fields must refer to the same passenger.

Response format

```
{
  "data": {
    "tickets": [{
      "number": string,
      "passengerReference": string,
      "isInfantTicket": boolean,
      "issueDate": datetime
    }
  ],
  "outboundFareFamily": fareFamilyObject,
  "returnFareFamily": fareFamilyObject,
  "tsmData": [{
    "passengerReference": string,
    "price": number,
    "referenceType": string,
    "referenceNumber": string
  }
  ],
  "appliedDiscounts": [{
    "flightReference": string,
    "amount": integer,
    "discountAmount": decimal,
    "id": string,
    "type": string,
    "passengerReference": string,
    "serviceCode": string,
    "promoNumber": string,
    "serviceReference": string,
    "usageType": integer,
    "promotionType": integer
  }
  ],
  "baseFarePrices": [{
    "currency": string,
    "passengerReference": string,
    "isInfantFare": boolean,
    "totalPrice": number,
```

```

    "basePrice": number,
    "priceHidden": boolean,
    "taxDetailsInformation": [{
      "taxCategory": string,
      "taxDetails": [{
        "rate": string,
        "currencyCode": string,
        "type": string
      }
    ]
  }
],
"foodOnBoards": [{
  "departureAirport": string,
  "arrivalAirport": string,
  "flightNumber": string
  "commercialFamily": string,
  "ru": string,
  "en": string,
  "zh": string
}],
"reservationRetrieveStatus": integer,
"ticketCouponStatuses": [{
  "ticketNumber": string,
  "couponStatuses": {
    flightReference: boolean
  }
}
],
"engineType": byte,
"isGroupReservation": bool,
"spentBonuses": {
  "amount": integer,
  "passengerReference": string
},
"spentBonusesByPassengers": [{
  "amount": integer,
  "passengerReference": string
}
],
"appliedCertificates": [{
  "amount": integer,
  "number": string
}
],
"originOfficeId": string,
"originOfficeType": string,
"journey": {
  "outboundFlights": [fullReservationFlight],
  "outboundFareFamily": {
    "code": string
  },
  "returnFareFamily": {
    "code": string
  },
},

```

```

"outboundDuration": string,
"returnFlights": [fullReservationFlight],
"returnDuration": string,
"origin": string,
"destination": string,
"departureDate": datetime,
"returnDate": datetime,
    "outboundBaggageAllowances": [{
        "quantity": integer,
        "weight": integer,
        "passengerType": integer
    }
],
    "returnBaggageAllowances": [{
        "quantity": integer,
        "weight": integer,
        "passengerType": integer
    }
]
},
"number": string,
"reservationDate": datetime,
"passengers": [passengerObject],
"services": [{
    "code": string,
    "status": string,
    "text": string,
    "petType": string,
    "category": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string,
    "packReferenceNumber": string,
    "price": number,
    "issueDate": datetime,
    "issuedByThirdPartyOffice": boolean
}
],
"seats": [{
    "number": integer,
    "title": string,
    "code": string,
    "status": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string,
    "packReferenceNumber": string,
    "issueDate": datetime,
    "issuedByThirdPartyOffice": boolean
}
],
"insurance": {
    "number": string,
    "code": string,
    "status": string,
    "referenceNumber": string,
    "packReferenceNumber": string,

```

```

    "issueDate": datetime,
    "issuedByThirdPartyOffice": boolean
  },
  "partnerProducts": [{
    "partnerId": integer,
    "partnerType": integer,
    "code": string,
    "amount": number,
    "currency": string,
    "passengerReference": string,
    "isInfantProduct": boolean
  }
],
"hasSpecialConditions": boolean
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }
]
}
]
"success": boolean
}
}

```

fareFamily object

```

{
  "fareFamilyId": integer,
  "code": string,
  "options": [{
    "code": string,
    "category": string,
    "order": integer,
    "showInPopup": boolean,
    "showInSummary": boolean,
    "showInUpsell": boolean,
    "showInFareIncluded": boolean,
    "showForAdult": boolean,
    "showForInfant": boolean,
    "showInBookingCart": boolean
  }
],
  "order": integer,
  "commercialFamily": string,
  "canRefund": boolean,
  "hasUpsale": boolean
}

```

fullReservationFlight object

```
{
  "aircraft": string,
  "aircraftIataCode": string,
  "referenceNumber": string,
  "origin": string,
  "destination": string,
  "departureDate": datetime,
  "arrivalDate": datetime,
  "flightNumber": string,
  "classOfService": string,
  "marketingCarrier": string,
  "operatingCarrier": string,
  "flightDuration": string,
  "stopDuration": string
}
```

passenger object

```
{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": integer,
  "isAdult": boolean,
  "birthDate": datetime,
  "sex": integer,
  "nationality": string,
  "documentNumber": string,
  "documentExpiryDate": datetime,
  "email": string,
  "phone": string,
  "documentType": integer
}
```

Response options

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true

Parameter		Description
data		The data returned by the request
	tickets	Information about issued tickets
	number	Ticket number. Example: 262 2439494143
	passengerReference	A reference to the passenger to whom this ticket belongs.
	isInfantTicket	An sign that this passenger has a baby
	issueDate	Date and time of issuing the tickets.
	outboundFareFamily	An array of farFamily objects with information about the selected fares for the "outbound" flight
	returnFareFamily	An array of farFamily objects with information about the selected fares for the "return" flight
	tsmData	An array of objects with information about additional services added to PNR
	passengerReference	Reference to the passenger to whom this service relates
	price	Service cost
	referenceType	The type of reference to the entry in the PNR, usually "SSR"
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR
	appliedDiscounts	Discounts applied in this PNR.
	flightReference	Reference to the segment for which the discount was applied
	amount	Discount rate as a percentage
	discountAmount	The absolute value of the discount in rubles
	id	ID of the campaign in CRM for which the discount was applied

Parameter		Description
	type	Discount type. Possible values: DISCOUNT - regular discount, PROMO - discount on the flight promo code. PROMOCODE - discount for additional services with a promotional code.
	passengerReference	Reference to the passenger for whom the discount was applied.
	serviceCode	The code of the added service for which the discount was applied.
	promoNumber	The number of the applied promotional code from CRM.
	serviceReference	Service identifier within the current booking.
	usageType	Discount type: on tariff or on order
	promotionType	Type of promo code: for a flight or for a service
	baseFarePrices	An array with information about base prices for tariffs.
	currency	The currency in which the value is presented. Example value: RUB
	passengerReference	Reference to the passenger for whom the cost is determined.
	isInfantFare	The sign if the passenger is an infant
	totalPrice	Total price
	basePrice	Base price (excluding tariffs and fees)
	priceHidden	The sign if the rate amount should be hidden for the client. Will be "true" for IT-tariffs.

Parameter			Description
		taxDetailsInformation	Services information about tariffs and fees. Full parameter's description information can be find in Amadeus documentation.
		taxCategory	Tax category
		taxDetails	Array of objects with information about tariffs and fees details.
		rate	Tariff/fee sum
		currencyCode	Currency code
		type	Tariff/fee type
		foodOnBoards	An array with information about meals available on flights
		departureAirport	Departure airport
		arrivalAirport	Arrival airport
		flightNumber	Flight number
		commercialFamily	Fare family type (different meals may be available on the same flight for economy and business).
		ru	Translation of the option into Russian
		en	Translation of the option into Englishкий язык
		zh	Translation of the option into Chinese

Parameter		Description
	reservationRetrieveStatus	Reservation receipt status. Can take values: 0 - Ok (success) 1 - TicketRequired (group booking, the passenger did not indicate the ticket number) 2 - InvalidTicketNumber (group booking, the passenger indicated an incorrect ticket number)
	ticketCouponStatuses	Array of information about statuses of flight coupons.
	ticketNumber	Ticket number.
	couponStatuses	Statuses of flight coupons.
	engineType	The system in which the reservation was created. Can take values: 0 - Unknown, 1 - Amadeus, 2 - Tais
	isGroupReservation	Group booking flag
	spentBonuses	Object with information about the total count of bonuses spent
	amount	Count of bonuses
	passengerReference	Reference to the first passenger
	spentBonusesByPassengers	An array with information about the number of bonuses spent by a specific passenger
	amount	Count of bonuses
	passengerReference	Reference to the passenger

Parameter		Description
	appliedCertificates	An array with information about applied certificates
	amount	Certificate denomination
	number	Certificate number
	originOfficeId	The office under which the reservation was created. For example, SVXU608AZ
	originOfficeType	The type of office under which the reservation was created. Possible values: dictionary of offices
	journey	Flight information for this booking
	outboundFlights	An array of fullReservationFlight objects with details of "outbound" flights
	outboundFareFamily	Service field with information about the selected fare for the "outbound" flight. For example, ECOFLEXRU
	returnFareFamily	Service field with information about the selected fare for the "return" flight
	outboundDuration	Total duration of the "outbound" flight
	returnFlights	An array of fullReservationFlight objects with detailed information about the "return" flights
	returnDuration	Total duration of the "return" flight
	origin	Initial airport of departure
	destination	Initial airport of arrival
	departureDate	Departure date
	returnDate	Return date
	outboundBaggageAllowances	An array of objects containing detailed information about baggage allowances for the "outbound" route

Parameter		Description
	returnBaggageAllowances	An array of objects containing detailed information about baggage allowances for the “return” route
	number	Number of PNR
	reservationDate	Date and time of booking creation
	passengers	An array of passengerData objects with passenger information
	services	Array of objects with information about additional services added to the booking
	code	Additional service code Possible values: dictionary of additional codes services
	status	Service adding status. Example: HK - service added successfully
	text	Text in ssr format for sending pet information to Amadeus. Example: DOG 1PC 8KG 45X35X25
	petType	Pet type Possible values: DOG, CAT, PARROT, CANARY
	category	Additional service category Possible values: dictionary of additional service codes
	passengerReference	Reference to the passenger for which the additional service has been added
	packReferenceNumber	Reference to the service package in which the service was added

Parameter		Description
	flightReference	Reference to the segment on which the service operates
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR.
	price	Additional service cost
	issueDate	Date and time of issuing the service.
	issuedByThirdPartyOffice	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued.
	seats	An array of objects with information about the places added to the booking
	number	The row on which the seat is selected
	title	Place
	code	Seat selection service code. Value: RQST
	status	Service adding status. Example: HK - service added successfully
	passengerReference	Reference to the passenger for whom the additional service has been added
	flightReference	Reference to the segment on which the service operates
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR.
	packReferenceNumber	Reference to the service package in which the location was added
	issueDate	Date and time of issuing the service.

Parameter		Description
	issuedByThirdPartyOffice	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued.
	insurance	Object with information about the insurance added to the booking
	number	Insurance document number
	code	Insurance code Possible values: dictionary of additional services (category insurance)
	status	Service adding status. Example: HK - service added successfully
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR.
	packReferenceNumber	Reference to the service package in which the service was added
	issueDate	Date and time of issuing the service.
	issuedByThirdPartyOffice	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued.
	partnerProducts	Array of objects with information about the partner services added to the booking
	partnerId	Unique ID of the partner in the IBE engine. The list of partners is available in Attachment 5
	partnerType	Unique code of partner's type The list of partner's types is available in Attachment 5

Parameter		Description
	code	Partner service code. Each partner has unique service codes. See the codes for partner services in Attachment 5.
	amount	Partner's service cost.
	currency	Tariffication currency
	passengerReference	Reference to the passenger for whom the service has been added
	isInfantProduct	The sign of service for infant
	hasSpecialConditions	The sign if one of the following services has been added to the booking : PETC, CBBG, WEAP, UMR, MAAS, MEDA, EXST, STCR, WCHR, WCHC, WCHS, WCBD, WCBW, WCMP, WCOB, DEAF, BLND, DEPA, DEPU
errors		Information about errors that occurred during the operation of the service
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from Amadeus' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
	code	Amadeus error code
	text	Amadeus error text

fareFamily

Parameter	Description
fareFamilyId	Serial number of the tariff family, used in recommendations.
code	Family code, for example: ECOLIGHT/ECOFLEX/ECOPREM and etc. Possible values: see the dictionary of the main tariff categories
order	Fare family display order
commercialFamily	Fare family type, for example: U6ECONOMY or U6BUSINESS
canRefund	The sign if the tariff is refundable
hasUpsale	The sign shows whether the given tariff can be "increased" to the next one in the price category.
options	Array of objects with information about tariff options. Used for the website of the booking engine.
code	Tariff option code For example: upTo5bonuses refundExchangeProhibited mealsOnBoard and etc. Used for the website of the booking engine.
category	Category code for option For example: bonus, ffServices and etc. Used for the website of the booking engine.
order	Order for option output
showInPopup	Whether to show an option in the "your options" pop-up window.

Parameter	Description
showInSummary	Whether to show the option on the Summary page.
showInUpsell	Whether to show the option in the information about the tariff increase.
showInFareIncluded	Whether to show an option when selecting a filter.
showForAdult	Whether to show an option for an adult passenger.
showForInfant	Whether to show an option for a baby.
showInBookingCart	Whether to show an option in the shopping cart.

fullReservationFlight

Parameter	Description
aircraft	Board number
aircraftIataCode	IATA aircraft code
referenceNumber	Reference to the current flight segment
origin	Departure airport for the current segment
destination	Arrival airport for the current segment
departureDate	Departure date and time for the current segment
arrivalDate	Arrival date and time for the current segment
flightNumber	Flight number
classOfService	Information about the selected tariff
marketingCarrier	Marketing airline For example: U6, S7
operatingCarrier	Airline operating the flight For example: U6, OK
flightDuration	Flight duration for the current segment

outboundBaggageAllowances and returnBaggageAllowances

Parameter	Description
quantity	count of baggage
weight	weight of baggage
passengerType	passenger type

passengerData

Parameter	Description
id	Passenger's serial number (numbered from 1)
hasInfant	An indication that this passenger has a baby
referenceNumber	Reference to this passenger in the booking
firstName	Passenger's name
middleName	Passenger's patronymic (may be absent)
surname	Passenger's surname
passengerType	Passenger type. Possible values: dictionary of passenger types
isAdult	The sign if the passenger is an adult true for passengers type ADT/SRC/YTH false for passengers type CHD/INF
birthDate	Date of birth
sex	Passenger gender Possible values: dictionary gender of passengers
nationality	Nationality (country) Possible values: dictionary of countries
documentNumber	Document number
documentType	Document type Possible values: dictionary of document types

Parameter	Description
documentExpiryDate	Date until which the document is valid Mandatory only for a foreign passport. For other documents, +10 years from the current date is automatically set
loyaltyCard	Loyalty card number "Wings"
email	Email
phone	Phone

Examples of successful requests and responses

Attachment X. Examples of responses from services

Examples of unsuccessful requests and responses

The request used a non-existent PNR number

```
{
  "errors": [{
    "ibeCode": 1101,
    "description": "Amadeus connectivity error.",
    "messages": []
  }
],
  "success": false
}
```

The request used a surname that is not in the PNR

```
{
  "errors": [{
    "ibeCode": 1111,
    "description": "Amadeus PNR_Retrieve function returned error.",
    "messages": []
  }
],
  "success": false
}
```

The request used a surname that does not match the surname in the requested ticket

```
{
  "errors": [{
    "ibeCode": 1111,
    "description": "Amadeus PNR_Retrieve function returned error.",
    "messages": []
  }
],
  "success": false
}
```

The request contains a ticket number that does not meet the requirements

```
{
  "errors": [{
    "ibeCode": 1008,
    "description": "Data validation error.",
    "messages": [{
      "text": "'Ticket Number' is not in the correct format."
    }
  ]
},
  "success": false
}
```

Method DELETE reservation

Description: the method allows you to delete the created reservation.

Request format /api/apiVersion/reservation

Request parameters

None. Cancellation reservations are identified by the session ID passed in the X-Session header.

Response format

```
{
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
},
  "success": false
}
```

```

    }
  ]
  "success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code, see Attachment 1. IBE error codes
description	Error description, see Attachment 1. IBE error codes
messages	An array of objects containing errors from Amadeus' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
code	Amadeus error code
text	Amadeus error text

Examples of unsuccessful requests and responses

An attempt was made to delete if the booking information was not loaded into the context.

```

{
  "errors": [
    {
      "ibeCode": 1001,
      "description": "Operation can not be performed due to invalid
reservation state.",
      "messages": []
    }
  ],
  "success": false
}

```

Booking management service v2.3

Changes in v 2.3:

In GET reservation service's response insurance object was replaced with insurances array.

Response format

```
{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string
      },
      "outboundDuration": string,
      "returnFlights": [fullReservationFlight],
      "returnDuration": string,
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    },
    "fareBasisCodesFlightRefs" : {[fareBasisCodesFlightRefs object]},
    "number": string,
    "reservationDate": datetime,
    "bookingTimeLimit": datetime,
    "passengers": [{
      "id": integer,
      "hasInfant": boolean,
      "referenceNumber": string,
      "firstName": string,
      "middleName": string,
      "surname": string,
      "passengerType": integer,
      "isAdult": boolean,
      "birthDate": datetime,
      "sex": integer,
      "nationality": string,
      "documentNumber": string,
      "documentExpiryDate": datetime,
      "email": string,
      "phone": string,
      "documentType": integer
    }
  ],
  "services": array,
  "seats": array,
  "insurances": [{
    "number": string,
    "passengerType": integer,
    "code": string,
    "status": string,
    "passengerReference": string,
    "price": double
  }
  ],
  "isAgentReservation": boolean,
  "isPayForbiddenForAgency": boolean,
  "partnerProducts": array
},
"errors": [{
```

```

        "ibeCode": integer,
        "description": string,
        "messages": [{
            "code": string,
            "text": string
        }
    ],
    "success": boolean
}
]
}

```

fullReservationFlight object

```

{
    "aircraft": string,
    "referenceNumber": string,
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "arrivalDate": datetime,
    "flightNumber": string,
    "classOfService": string,
    "operatingCarrier": string,
    "flightDuration": string,
    "statuses": [string]
}

```

fareBasisCodesFlightRefs object

```

{
    "fareBasis": string,
    "flightReferenceNumber": string,
    "passengerReferenceNumber": string
}

```

Response parameters (short format)

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
isGroupReservation	IT/BT reservation sign
isCompanyGroupReservation	Group booking sign
number	PNR number
reservationDate	Booking creation date and time

bookingTimeLimit	Date and time until which the unpaid booking will exist (rules are set in the settings in IBE)
passengers	Array of passengerData objects with information about passengers
services	Array of objects with information about auxiliary services.
seats	Array of objects with information about selected seats.
insurances	Array of objects with information about selected insurances.
partnerProducts	Array of objects with information about partner's services.
journey	Information about flights for present booking.
isAgentReservation	Agent booking sign to disable the booking management module
isPayForbiddenForAgency	Sign that payment is not available for an unpaid booking
fareBasisCodesFlightRefs	Array of objects with fare basis info
errors	Information about errors appeared during query execution.

insurances structure description

Parameter	Description
number	Insurance document number
passengerType	Passenger type
code	Service code. Example: "URFL"
status	Service status, Example: "HK"
passengerReference	Reference to passenger to whom present service is provided
price	Service price

Obtain partner's services information service: api/reservation/partners

Obtain partner's services information service

API service	reservation/partners
Short description	Allows to retrieve information about partner's services in booking and perform resend of paid services. Used in general for obtaining of references to tickets/partner's policies
Supported HTTP(S)-methods	GET, POST
Conterminous systems methods used	Amadeus: PNR_Retrieve, PNR_AddMultiElements Aeroexpress: getOrderTickets

GET api/reservation/partners method

Description: allows to receive information about partner's service in booking

Request format /api/apiVersion/reservation/partners/{ {bookingId} }/details

where bookingId - unique order ID of partner's service in present booking

Query example

After adding of parnter's service, for instance Aeroexpress to booking, following structure gets returned in api/services response:

```
"partnerProducts": [  
  {  
    "partnerId": 4,  
    "partnerType": 2,  
    "bookingStatus": 1,  
    "code": "STANDARTOW",  
    "amount": 500.00,  
    "currency": "RUB",  
    "passengerReference": "2",  
    "isInfantProduct": false,  
    "bookingId": "65664d9b-cc47-46d0-8937-162f582512e8"  
  }  
]
```

To receive information about this booking following request should be sent:

/api/v2.2/reservation/partners/65664d9b-cc47-46d0-8937-162f582512e8/details

Response format

```
{
  "data": {
    "bookingId": string,
    "description": string,
    "document": string,
    "documentUrl": string,
    "passengerReferences": [
      string
    ],
    "flightReferences": [
      string
    ]
  },
  "errors": [{
    "ibeCode": string,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
},
  "success": true
}
```

Response parameter

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
bookingId	unique ID of partnter's service in IBE. Matches value from request.

	description	Description of partner's service For instance "СТАНДАРТ" for STANDARTOW Aeroexpress tariff. Or "ТРӘБЕЛ" for Alfastrah partner's insurance.
	document	order ID in partner's system
	documentUrl	reference to partner's ticket/policy
	passengerReferences	Array of references to passengers to whom present service is provided Optional field.
	flightReferences	Array of references to flights to which present service is added. Optional field.
errors		Information about errors appeared during query execution.
	ibeCode	IBE error code
	description	IBE error description
	messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
	code	Internal error code
	text	Internal error description

Example of response for Aeroexpress partner

```
{
  "data": {
    "bookingId": "8d7d39d9-2d87-49ff-970c-de94f4b0cb43",
    "description": "СТАНДАРТ",
    "document": "990801",
    "documentUrl":
"https://api.aeroexpress.ru:48000/RenderTicket?oid=551913&guid=017669A2606DEA57D5EEDEE2F1DC86C9&token=74277705"
  },
  "errors": [],
  "success": true
}
```

```
}
```

Example of response for Alfastrah

```
{
  "data": {
    "bookingId": "b1bfb8b3-ebfe-4434-b97b-1e90298a360a",
    "description": "TPЭBEJ",
    "document": "27514142",
    "documentUrl": "http://uat-tes.alfastrah.ru/travel-ext-
services/reports?parameters=7E964978DD9AC0ED5D3165945FBED849D2C8D60734B455FA1
93B482A91E92FEC4E302791F557381FF437709023D1F30E"
  },
  "errors": [],
  "success": true
}
```

Example of response with error

Invalid bookingId sent

```
{
  "errors": [
    {
      "ibeCode": 1953,
      "description": "Unknown partner product booking ID.",
      "messages": []
    }
  ],
  "success": false
}
```

POST api/reservation/partners/send method

Description: allows to resent issued tickets/policies for partner's service.

Request format /api/apiVersion/reservation/partners/{ {bookingId} }/send

where bookingId - unique order ID of partner's service in present booking

Request parameters

no

Query example

/api/v2.2/reservation/partners/65664d9b-cc47-46d0-8937-162f582512e8/send

Response format

```
{
  "errors": [{
    "ibeCode": string,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
},
],
"success": true
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	IBE error code
description	IBE error description
messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
code	Internal error code
text	Internal error description

Example of response

```
{
  "errors": [],
  "success": true
}
```

Example of response with error

Invalid bookingId sent

```
{
  "errors": [
    {
      "ibeCode": 1953,
      "description": "Unknown partner product booking ID.",
      "messages": []
    }
  ],
  "success": false
}
```

Example of response with error

Attempt to resend not issued ticket.

```
{
  "errors": [
    {
      "ibeCode": 1952,
      "description": "Partner product wasn't issued.",
      "messages": []
    }
  ],
  "success": false
}
```

PNR Tariff confirmation service

API service	pricing
Short description	Service allows to confirm availability of tariffs used in booking creation.
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Amadeus: PNR_AddMultiElements, Fare_PricePNRWithBookingClass, Ticket_CreateTSTFromPricing

Description: allows to confirm availability of tariffs used in booking creation.

Service is indirectly called during adding of all necessary data to PNR. Service should be directly called if information about prices gets changed after booking creation.

For instance if promocode was added to a flight.

Request format /api/apiVersion/pricing

Request body

```
{
  "outboundFareFamily": {
    "category": string,
    "code": string,
    "isSubsidized": boolean
  },
  "returnFareFamily": {
    "category": string,
    "code": string,
    "isSubsidized": boolean
  },
  "fares": [{
    "fareBasis": string,
    "isBreakpoint": boolean
  }
],
  "sessionMarker": string
}
```

Request parameters

Parameter	Description
outboundFareFamily	Object with information about outbound Fare family
category	Fare family category, for ex: ECOLIGHT/ECOFLEX/ECOPREM Possible values see: Attachment 2. Main fare family categories
code	Fare family code, for ex: ECOLIGHTRU/ECOFLEXRU/ECOPREMRU etc.
isSubsidized	sign of subsidised tariff
returnFareFamily	Object with information about return Fare family
category	Fare family category, for ex: ECOLIGHT/ECOFLEX/ECOPREM Possible values see: Attachment 2. Main fare family categories
code	Fare family code, for ex: ECOLIGHTRU/ECOFLEXRU/ECOPREMRU etc.
isSubsidized	sign of subsidised tariff
fares	Array of objects with information about selected tariffs
fareBasis	Base tariff title Example: WPROW or WECOW
isBreakpoint	Flag identifying whether tariff ends on present segment.
sessionMarker	Amadeus session marker

Response format

```
{
  "data": {
    "ticketFares": [{
      "referenceNumber": string,
      "referenceQualifier": string,
      "baseAmount": number,
      "totalAmount": number,
      "passengerReferences": [{
```

```

        "referenceNumber": string,
        "referenceQualifier": string
    }
  ]],
  "fareZapOff": {
    "amount": integer,
    "discountSsr": string,
    "discountCardSsr": string
  }
},
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }]
}],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
ticketFares	Information about confirmed prices
referenceNumber	TST number
referenceQualifier	reference type Possible value: «TST»
baseAmount	Base ticket price (without tariffs and fees) for present segment and passenger
totalAmount	full ticket price for present segment and passenger
passengerReferences	Array of objects with information about passengers for which current ticket price is actual
referenceNumber	Reference to passenger

		referenceQualifier	Reference type PA – for adults, PI – for infants
	fareZapOff		Object with discount information presented in booking
		amount	Discount amount in percent
		discountSsr	SSR with information about discount Example: "DISCOUNT/DS03/1-1PSJTS5/" where DS03 – discount percent special format 1-1PSJTS5 – CRM Discount campaign ID
		discountCardSsr	SSR with information about "Wings" loyalty card which was used for discount application Example: "DISCOUNTCARD/1000332007"
	errors		Information about errors appeared during query execution.
		ibeCode	Error code, see EN_Attachment 1. IBE error codes
		description	Code description, see. EN_Attachment 1. IBE error codes
	messages		Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode
		code	Amadeus error code
		text	Amadeus error text

Failed request and response examples

Attempt to confirm PNR with tariff which is no longer available:

```
{
  "errors": [{
    "ibeCode": 1117,
    "description": "Amadeus Fare_PricePNRWithBookingClass function
return error. No fare for booking code",
    "messages": [{
      "code": "0",
      "text": "NO FARE FOR BOOKING CODE-TRY OTHER PRICING
OPTIONS"
```

```
    }
  ]
},
],
"success": false
}
```

Service for obtaining a seatmap without PNR

API service	SeatmapStandalone
Short description	Allows to obtain Seat Map without PNR
Supported HTTP(S)-methods	POST
Conterminous systems methos used	Amadeus: Air_RetrieveSeatMap

POST SeatmapStandalone method

Description: Gets information about the seat map on the plane without pre booking.

Request format: api/apiVersion/SeatmapStandalone

Request body:

```
{
  "passengers": [{
    "id": integer,
    "firstName": string,
    "middleName": string,
    "surname": string,
    "passengerType": string,
    "birthDate": datetime,
    "sex": integer,
    "loyaltyCard": string,
    "disability": {
      "isDeaf": boolean,
      "isBlind": boolean,
      "hasGuideDog": boolean,
      "assistanceType": integer,
      "wheelchairType": integer,
      "commentary": string
    },
    "hasInfant": boolean,
    "nationality": string
  }
],
```

```

"origin": string,

"destination": string,

"departureDate": datetime,

"flightNumber": string,

"classOfService": string,

"commercialFamily": string,

"currency": string
}

```

Request paramaters

Parameter	Description
passengers	Array of objects with information about passengers
id	Passenger's index number (starting from 1)
firstName	Passenger's first name
middleName	Passenger's middle name (might be empty)
surname	Passenger's surname
passengerType	Passenger's type Possible values: see Attachment 2, passenger types
birthDate	Date of birth This parameter determines the availability of seats in the emergency row.
sex	Passenger's sex Possible values: see Attachment 2, passenger sex
loyaltyCard	"Wings" Loyalty card number (might be empty)
disability	Information about passenger's disability
isDeaf	sign of deafness
isBlind	sign of blindness

	hasGuideDog	requires guide-dog escort. Should be sent only in junction with isBlind flag
	assistanceType	Passenger's assistance type Possible values: see Attachment 2, assistant types
	wheelchairType	Wheelchair type Possible values: see Attachment 2, wheelchair types
	commentary	Commentary (optional)
	hasInfant	Sign that the passenger has an Infant
	nationality	Passenger nationality
	origin	Origin airport
	destination	Destination airport
	departureDate	Date and time of departure
	flightNumber	Flight number
	classOfService	Information about selected tariff. It can be found in “GET api/flights” response: data.outboundFlights.flights.flightNumber or data.returnFlights.flights.flightNumber
	commercialFamily	Fare family type
	currency	Search currency

Response format

```
{
  "data": {
    "compartmentDetails": [{
      "number": integer,
      "seats": [{
        "seat": string,
        "number": integer,
        "description": string,
        "characteristics": [string],
        "occupation": string,
        "passengerRelatedPrice": [{
          "name": string,
          "type": string,
          "uniqueCustomerIdentifier": string,
          "seatPrice": number,
          "isSeatUnavailable": boolean,
          "hasInfant": boolean
        }
      ],
      "classDesignator": string
    }
  ]
}
```

```
    }
  ],
  "isEmpty": boolean,
  "passengersTotalData": [{
    "passengerReference": string,
    "passengerPriceLevels": [{
      "price": decimal,
      "priceLevel": integer
    }
  ]
}
]
},
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string,
    "additionalInfo": {
      "ssrType": string,
      "info": string
    }
  ]
}
```



```

    }
  }
]
}
],
"success": boolean
}

```

Response parameter

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
isEmpty	Flag identifying whether data is returned from amadeus This field can be used to determine possibility of selecting seats on present flight segment
compartmentDetails	Array of objects with information about seats grouped by rows
number	Row number
seats	Array of objects which contains detailed information about seats
seat	Seat. Possible values: A-F
number	Seat. Possible values: A-F
description	Seat description 9 – central seat A – aisle seat W – window seat

		characteristics	Seat characteristics Possible values: see Attachment 2. Most widely spread Seatmap seat characteristics
		occupation	Sign of seat's occupation Possible values: O – occupied, F- free seats
		passengerRelatedPrice	Array of objects with information about seat prices for. Only available for seats with CH characteristics. .
		name	Passenger name
		type	Passenger type Possible values: see EN Attachment 2. Passenger types
		uniqueCustomerIdentifier	Unique passenger ID
		seatPrice	Seat price
		isSeatUnavailable	Seat availability flag
		hasInfant	Flag identifying that passenger travels with infant
		classDesignator	Information about seat class. Possible values: ECONOMY, BUSINESS
		passengersTotalData	Array of objects with information about the cost of seats for passengers
		passengerReference	Passenger reference
		passengerPriceLevels	Array of objects of seat prices available to the passenger
		price	Seat price
		priceLevel	Price level for seats
errors			Information about errors appeared during query execution.
		ibeCode	IBE error code, see. EN Attachment 1. IBE error codes
		description	IBE error description, see EN Attachment 1. IBE error codes
		messages	Array of objects containing errors from Amadeus' response. Array might be empty. It is recommended to analyze ibeCode field.
		code	Amadeus error code
		text	Amadeus error text

	additionalInfo	Additional error information
	ssrType	SSR type
	info	information

Auxiliary services management service

Auxiliary services management service v2.1

API Service	catalogue / services
Short description	Services allows to retrieve information about auxiliary services and add them to PNR.
Supported HTTP(S)-methods	GET, POST
Conterminous systems methods used	Amadeus: PNR context retrieve - PNR_Retrieve Obtain list of available services - Service_IntegratedCatalogue, Insurance_Pricing Adding of services - PNR_AddMultiElements Service issuing - Ticket_RetrieveListOfTSM, Ticket_CreateTSMFromPricing Partner's services : Aeroexpress

GET catalogue method

Description: Obtaining of information about auxiliary services available for adding to booking.

Request format /api/apiVersion/catalogue

Request parameters

Parameter	Description
currency	Currency that is used for pricing services

Response format

```
{
  "data": {
    "services": [{
      "code": string,
      "flightReferences": [{
        "referenceNumber": string,
        "passengerPrices": [{
          "referenceNumber": string,
```

```

        "amount": number
      }
    ]
  }
]
},
"seatServices": [{
  "code": string,
  "flightReferences": [{
    "referenceNumber": string,
    "passengerPrices": [{
      "referenceNumber": string,
      "amount": number
    }
  ]
}]
},
],
"petsServiceConfiguration": [{
  "petsList": [string],
  "flightReferences": [string]
}
],
"insuranceProducts": [{
  "code": string,
  "amount": number
}
],
"partnerProducts": [{
  "partnerId": integer,
  "partnerType": integer,
  "code": string,
  "amount": number,
  "currency": string,
  "bookingStatus": integer
}
],
"portions": [{
  "type": string,
  "number": string,
  "flightReferenceNumbers": [string]
}
],
"serviceCategories": [{
  "code": string,
  "category": string,
  "order": integer,
  "fullDirection": bool,
  "onlyForRussianCitizens": bool
}
],
"serviceOptions": [{
  "code": string,
  "preselect": boolean,
  "category": string,

```

```

        "priority": integer,
        "flightReferences": [string]
    }
],
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
]
}
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
services	Array of objects with information about airline's available services
code	Service code Possible values: see Attachment 2. Auxiliary services types
flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
referenceNumber	Reference to flight segment
passengerPrices	Array of objects with information about passenger and service price
referenceNumber	Reference to passenger
amount	Price of service
serviceOptions	Array of objects with information about which services should be automatically pre-selected in IBE frontend.

	category	Service category
	code	Service code Possible values: see Attachment 2. Auxiliary services types
	flightReferences	Array of objects with information about segments on which services are available
	preselect	If true, service should be automatically pre-selected in UI. <i>Note: Initial logic was applied to Web-client usage</i>
	priority	Priority of automatic service selection, 0 - highest. For instance, if service with priority 0 is unavailable, then service with priority 1 will be added etc.
	seatServices	Array of objects with information about flight's seats available for selection IMPORTANT: api/seatmap service returns more detailed information about available seats and should be used for "seat selection" service booking.
	code	Service code, SEAT
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	petsServiceConfiguration	Array of objects with information about "pet transportation" service availability on different segments. Needed for pet transportation validation on "business" family fares.
	petsList	Array of strings, containing list of pet types available for transportation
	flightReferences	Array of strings, containing list of segments on which service is available
	insuranceProducts	Array of objects with information about insurances available for selection

	code	Insurance code Possible values: see Attachment 2. Auxiliary services types
	amount	Insurance cost
	serviceCategories	Array of objects with information about services and order of its display in website
	code	Service code Attachment 2. Auxiliary services types
	category	Service category
	order	Display index in auxiliary services page. 1 - page top
	fullDirection	Service for directions
	onlyForRussianCitizens	Service only for russian citizens
	partnerProducts	Array of objects with information about partners services
	partnerId	Partners unique ID in IBE. Alfastrakh – 1, Sealine insurance – 2, Aeroexpress – 4
	partnerType	Partner type Insurance – 1, Aeroexpress – 2
	code	Partner's service code. Each partner has its own unique set of codes. See EN Attachment 4. Partner's services.
	amount	Partner's service cost.
	currency	currency

	bookingStatus	Partner's service booking status Possible values: 0 - unknown, 1 - service is not issued, 2 - service is issued, 3 - error issuing of service, 4 - refund performed
	portions	Service field
	type	Service field
	number	Service field
	flightReferenceNumbers	Service field
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode
	code	Amadeus error code
	text	Amadeus error text

Successful request and response examples

EN_Attachment X. Request and response examples

POST services method

Description: allows to add services to booking.

Request format /api/apiVersion/services

Request body

```
{
  "insurance": string,
  "seats": [{
    "number": integer,
    "title": string,
```

```

    "passengerReference": string,
    "flightReference": string
  }
],
"partnerProducts": [{
  "partnerId": integer,
  "code": string
}
],
"services": [{
  "code": string,
  "passengerReference": string,
  "flightReference": string
}
],
"currency": string
}

```

Request parameters

Parameter	Description
insurance	Code of insurance which needs to be added Possible values: Attachment 2. Auxiliary services types
services	Array of objects with information about adding services
code	Auxiliary service code Possible values: Attachment 2. Auxiliary services types (categories excluding insurance, seat)
passengerReference	Reference to passenger to whom service is being added
flightReference	Reference to flight segment to which service is being added
seats	Array of objects with information about adding seats
number	Seat row number
title	Seat title
passengerReference	Reference to passenger to whom service is being added
flightReference	Reference to flight segment to which service is being added
partnerProducts	Array of objects with information about adding partners services
partnerId	Partner's unique Id
code	Partner's unique code

currency	Currency used for pricing services
----------	------------------------------------

Request examples with airline's services

Примеры запросов с услугами Авиакомпаний

Add avia-insurance

```
{
  "seats": [],
  "insurance": "URFL",
  "partnerProducts": [],
  "services": []
}
```

Add one additional luggage bag and travel insurance

```
{
  "seats": [],
  "insurance": "URTR",
  "partnerProducts": [],
  "services": [
    {
      "code": "ABAG",
      "passengerReference": "2",
      "flightReference": "1"
    }
  ]
}
```

Add seat

```
{
  "seats": [
    {
      "number": 7,
      "title": "A",
      "passengerReference": "2",
      "flightReference": "1"
    }
  ],
  "insurance": null,
  "partnerProducts": [],
  "services": []
}
```

Request samples with partner's services

Add "Standart" Aeroexpress tariff

```

{
  "seats": [],
  "insurance": null,
  "partnerProducts": [
    {
      "partnerId": 4,
      "code": "STANDARTOW"
    }
  ],
  "services": []
}

```

Logic of adding/deleting of services in booking

IBE always refreshes booking auxiliary services data, basing on request parameters.

Example:

Add avia-insurance and luggage bag

```

{
  "seats": [],
  "insurance": "URFL",
  "partnerProducts": [],
  "services": [{
    "code": "ABAG",
    "passengerReference": "2",
    "flightReference": "1"
  }
]
}

```

To delete insurance from booking following request should be sent

```

{
  "seats": [],
  "insurance": null,
  "partnerProducts": [],
  "services": [{
    "code": "ABAG",
    "passengerReference": "2",
    "flightReference": "1"
  }
]
}

```

Present logic is implemented for all service types including partner's ones.

Response format

Adding service part in response format matches one in response of GET api/reservation method.

Description of services, seats, insurance, partnerProducts arrays can be found in that section.

Failed add partner services request examples:

Invalid partner code sent

There is no partner with 111 code

```
{
  "seats": [],
  "insurance": null,
  "partnerProducts": [
    {
      "partnerId": 111,
      "code": "STANDARTOW"
    }
  ],
  "services": []
}
```

Response

```
{
  "errors": [
    {
      "ibeCode": 1950,
      "description": "Unknown Partner.",
      "messages": []
    }
  ],
  "success": false
}
```

Auxiliary services management service v2.2

Changes in version 2.2:

1. In API v2.2 there were introduced ability to perform additional purchase (upsale) of services after booking payment. All services existing in booking should be passed in api/services method, therefore referenceNumber field, obtained from GET api/reservation response should be passed for **all paid services**.

2. There were introduced ability to purchase partner's insurance in IBE. Concurrent adding of Amadeus insurances in insurance block and partner's insurances in partnerProducts is prohibited.

Example of auxiliary services upsale

1. Create and perform payment of booking without auxiliary services
2. Add luggage bag in booking, referenceNumber is not specified because this service is not exists in booking yet.

REQUEST api/services

```
{
  "insurance": null,
  "partnerProducts": [],
  "seats": [],
  "services": [
    {
      "passengerReference": "2",
      "flightReference": "1",
      "code": "ABAG"
    }
  ]
}
```

3. Perform payment of added service
4. Attempt to upsale HBAG without specifying ABAG service, payed earlier:

REQUEST api/services

```
{
  "insurance": null,
  "partnerProducts": [],
  "seats": [],
  "services": [
    {
      "passengerReference": "2",
      "flightReference": "1",
      "code": "HBAG"
    }
  ]
}
```

Receive error because it is not possible to delete payed service

RESPONSE api/services

```
{
  "errors": [
    {
      "ibeCode": 1012,
      "description": "Unable to cancel additional service with issued
ticket.",
      "messages": []
    }
  ],
  "success": false
}
```

5. Add HBAG service, pass payed ABAG service (referenceNumber is specified)

REQUEST api/services

```
{
  "insurance": null,
  "partnerProducts": [],
  "seats": [],
  "services": [
    {
      "passengerReference": "2",
      "flightReference": "1",
      "code": "ABAG",
      "referenceNumber": "17"
    },
    {
      "passengerReference": "2",
      "flightReference": "1",
      "code": "HBAG"
    }
  ]
}
```

6. In GET api/reservation response two services gets returned: just added and previously payed

RESPONSE PART api/reservation

```
"services": [
  {
    "code": "ABAG",
    "status": "HK",
    "passengerReference": "2",
    "flightReference": "1",
    "referenceNumber": "17",
    "ticketNumber": "262 4550915479",
    "price": 2000
  },
  {
    "code": "HBAG",
    "status": "HK",
    "passengerReference": "2",

```

```
        "flightReference": "1",
        "referenceNumber": "29"
    }
]
```

7. POST api/payment will return link for payment of HBAG service only.

Example of purchasing of partner's insurance in IBE

Limitations:

1. Sealine insurance only available for mobile app API-key and mobile Amadeus office correspondingly
2. Alfastrah insurance only available for web API-key and Amadeus web-office correspondingly
3. Concurrent adding of Amadeus insurances in insurance block and partner's insurances in partnerProducts is prohibited.

Partner's and service's codes are available in [Attachment 4. Partner's services.](#)

Adding of Alfastrah insurance

1. Create booking, request GET api/catalogue with web-office API-key, check that partner's insurances is present in response. Add one of them to booking via POST services method.

REQUEST api/services

```
{
  "insurance": null,
  "seats": [],
  "partnerProducts": [{
    "partnerId": 1,
    "code": "TRAVEL"
  }
],
  "services": []
}
```

2. Block with partner's insurance exists in POST service response.

RESPONSE api/services

```
"partnerProducts": [
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 847.00,
  }
]
```



```

    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "3146ad12-bb61-445f-922f-10f36d0fc188"
  }
]

```

3.Example of booking 1 adult + 1 child + 1 infant:

RESPONSE api/services

```

"partnerProducts": [
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 1694.00,
    "currency": "RUB",
    "passengerReference": "3",
    "isInfantProduct": true,
    "bookingId": "27713956-2687-49b8-a96f-25e61d252824"
  },
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 847.00,
    "currency": "RUB",
    "passengerReference": "3",
    "isInfantProduct": false,
    "bookingId": "1ad570e6-48da-4615-baf5-53769925a253"
  },
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 847.00,
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "9068c420-514e-4ea7-8790-46afb31dd7b0"
  }
]

```

Adding of Sealine insurance

1. Create booking, request GET api/catalogue with mobile office API-key, check that partner's insurances is present in response. Add one of them to booking via POST services method.

REQUEST api/services

```
{
```

```

"seats": [],
"insurance": "",
"partnerProducts": [
  {
    "code": "SLAVIA",
    "partnerId": 2
  }
],
"services": []
}

```

2. Block with partner's insurance exists in POST service response.

RESPONSE api/services

```

"partnerProducts": [
  {
    "partnerId": 2,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "SLAVIA",
    "amount": 260.00,
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "0e91e664-e4fa-4177-b2bc-e675ca197fa8"
  }
],

```

Auxiliary services management service v2.3

Changes in version 2.3: insurance block is no longer returned in service's response. Only partner's insurances from partnerProducts block are now available.

Method's structure is described in api/reservation version 2.3.

Request and responses examples

Error adding of seat (seat is unavailable)	

<pre>{ "seats": [{ "number": 44, "title": "A", "passengerReference": "2", "flightReference": "1" }], "insurance": null, "partnerProducts": [], "services": [] }</pre>	<pre>"errors": { "seats": [{ "number": 44, "title": "A", "internalError": 2002, "internalErrorMessage": "Seat is unavailable." }] }</pre>
<p>Error adding of seat (seat is engaged)</p>	
<pre>{ "seats": [{ "number": 5, "title": "A", "passengerReference": "2", "flightReference": "1" }], "insurance": null, "partnerProducts": [], "services": [] }</pre>	<pre>"errors": { "seats": [{ "number": 5, "title": "A", "internalError": 2002, "internalErrorMessage": "Seat is unavailable." }] }</pre>
<p>Error adding of seat (attempt to book seat for adult with infant not in the middle row)</p>	
<pre>{ "seats": [{ "number": 6, "title": "C", "passengerReference": "2", "flightReference": "1" }], "insurance": null, "partnerProducts": [], "services": [] }</pre>	<pre>"errors": { "seats": [{ "number": 6, "title": "C", "passengerReference": "2", "flightReference": "1", "internalError": 2001, "internalErrorMessage": "Seat is unavailable for this passenger type." }] }</pre>
<p>Error adding of airline's service (invalid service code)</p>	

<pre>{ "seats": [], "insurance": null, "partnerProducts": [], "services": [{ "code": "ABBB", "passengerReference": "2", "flightReference": "1" }] }</pre>	<pre>"errors": { "services": [{ "code": "ABBB", "passengerReference": "2", "flightReference": "1", "internalError": 4002, "internalErrorMessage": "Invalid service code." }] }</pre>
<p>Error adding of airline's service (invalid reference to passenger)</p>	
<pre>{ "seats": [], "insurance": null, "partnerProducts": [], "services": [{ "code": "ABAG", "passengerReference": "5", "flightReference": "1" }] }</pre>	<pre>"errors": { "services": [{ "code": "ABAG", "passengerReference": "5", "flightReference": "1", "internalError": 1001, "internalErrorMessage": "Invalid passenger or segment reference." }] }</pre>
<p>Error adding of partner's service (invalid code)</p>	
<pre>{ "seats": [], "insurance": null, "partnerProducts": [{ "partnerId": 4, "code": "STAND" }], "services": [] }</pre>	<pre>"errors": { "partnerProducts": [{ "internalError": 1000, "internalErrorMessage": "Unknown error.", "partnerId": 4, "partnerType": 2, "code": "STAND", "amount": 0, "currency": "RUB" }] }</pre>

Описание: allows to get a favorite services for passengers

Формат запроса /api/apiVersion/services/favorite

Request parameters

Parameter	Description
docNumbers	Document numbers of all passengers
emdCodes	Checked codes for favorite services
months	Search period in months

Request body example

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
{
  "DocNumbers": [
    string
  ],
  "EmdCodes": [
    string
  ],
  "Months": integer
}
```

Response body

```
{
  "data": [
    {
      "docNum": string,
      "emdCode": string,
      "value": string
    }
  ],
  "errors": [],
  "success": boolean
}
```

Response parameters

Parameter	Description
data	The data returned by the request
docNum	Passenger's document number
emdCode	EmdCode of favorite service
value	Value for favorite service
errors	Information about errors that occurred during the operation of the service
ibeCode	Error code, see Attachments 1. IBE error codes
description	Error description, see Attachments 1. IBE error codes
messages	An array of objects containing errors. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Error code
text	Error text
success	Sign of success of the request If the errors array is empty, then the field value will be true.

Response body example

```
{
  "data": [
    {
      "docNum": "3344112121",
      "emdCode": "RQST",
      "value": "6A"
    }
  ],
  "errors": [],
  "success": true
}
```

Auxiliary partner's services management service

API service	partners/types, partners/[id]
Short description	Services allows to retrieve information about partner's services and add them to booking
Supported HTTP(S)-methods	GET, POST
Conterminous systems methods used	Partner's API

GET partners/types method

Description: Service allows to retrieve list of partner's services which are available for booking for specific application type.

Validated via API-key passing in method.

Request Format /api/apiVersion/partners/types

Request header X-session is not needed for call, no data from booking context is used.

Request parameters no parameters

Response format

```
{
  "data": [{
    "partnerId": integer,
    "name": string
  }
],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
}
],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
partnerId	Unique partner's code See Attachment 4. Partner's services .
name	Partner's name
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Response example (Website application type)

```
{
  "data": [
    {
      "partnerId": 4,
      "name": "Aeroexpress"
    },
    {
      "partnerId": 1,
      "name": "AlfaInsurance"
    }
  ],
  "errors": [],
  "success": true
}
```

Response example (mobile application type)

```
{
  "data": [
    {
```



```

        "partnerId": 4,
        "name": "Aeroexpress"
    },
    {
        "partnerId": 2,
        "name": "SealineInsurance"
    }
],
"errors": [],
"success": true
}

```

GET partners/id method

Description: service allows to retrieve list of services available for specific partner.

Request format /api/apiVersion/partners/id

Request example /api/apiVersion/partners/1 - for retrieving of Alfastrah services.

Request headers X-Session **should** be passed because information about available partner's services depends on passenger's list, flight directions etc.

All information will be taken from booking context.

Request parameters

id - partner's code

Response format

```

{
  "data": [{
    "partnerId": integer,
    "partnerType": integer,
    "bookingStatus": integer,
    "code": string,
    "amount": nubmer,
    "preselectedOption": {
      "preselect": boolean,
      "passengers": [{
        "referenceNumber": string,
        "passengerType": string
      }
    ]
  },
  "compatibleProductCodes": [
    string
  ],
  "internalError": integer,
  "internalErrorMessage": string
}

```

```

],
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string,
    "additionalInfo": {
      "ssrType": string,
      "info": string
    }
  }
]
}
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
partnerId	Unique partner's code See Attachment 4. Partner's services.
partnerType	Unique partner's type See Attachment 4. Partner's services.
bookingStatus	Booking service's status Field is left for backward compatibility and will always be set to 0.
code	Code of partner's service
amount	Cost of partner's service
currency	Currency
preselectedOption	An array of settings for pre-selecting an partners service (in the basket at the third step)

	preselect	Sign of preselection of a partner service
	passengers	An array of passengers for whom a partner service is preselected
	referenceNumber	Passenger reference
	passengerType	Passenger type, see Attachment 2. Dictionary of Passenger Types
	compatibleProductCodes	An array of partner service codes that can be added to a booking at the current rate
	internalError	Internal engine error code
	internalErrorMessage	Description of an internal engine error
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Amadeus error code
	text	Amadeus error text
	additionalInfo	Additional error information
	ssrType	Ssr type
	info	Information

Response example, Aeroexpress

Request /api/apiVersion/partners/4

```
{
  "data": [{
    "partnerId": 4,
    "partnerType": 2,
    "bookingStatus": 0,
    "code": "STANDARTOW",
    "amount": 500.0,
    "currency": "RUB"
  }, {
    "partnerId": 4,
    "partnerType": 2,
    "bookingStatus": 0,
    "code": "ADVSTANDART",
    "amount": 450.0,
    "currency": "RUB"
  }
}
```

```
],
"errors": [],
"success": true
}
```

Response example, Alfastrah

Request /api/apiVersion/partners/1

API-key - mobile application

```
{
  "errors": [{
    "ibeCode": 1957,
    "description": "A partner type is not available for current App Type.",
    "messages": []
  }
],
"success": false
}
```

Response example, Telemedicine

Request /api/apiVersion/partners/7

```
{
  "data": [{
    "partnerId": 7,
    "partnerType": 3,
    "bookingStatus": 0,
    "code": "SBERDOC",
    "amount": 300.00,
    "currency": "RUB",
    "preselectedOption": {
      "preselect": true,
      "passengers": [{
        "referenceNumber": "2",
        "passengerType": 0
      }
    ]
  }
], {
  "partnerId": 7,
  "partnerType": 3,
  "bookingStatus": 0,
  "code": "SBERVET",
  "amount": 300.00,
  "currency": "RUB"
}
],
"errors": [],
"success": true
}
```

POST partners method

Description: service allows to add partner's services to booking

Request format /api/apiVersion/partners

Request header X-Session **should** be passed in call because services are added to specific booking

Request format

```
{
  "products": [{
    "partnerId": integer,
    "code": string,
    "amount": decimal,
    "passengerReference": string,
    "isInfantProduct": boolean
  }
]
```

Request parameters

Parameter	Description
partnerId	Partner's unique code See Attachment 4. Partner's services.
code	Code of partner's service
amount	The cost of the added service
passengerReference	Reference to the passenger to whom the service is being added (mandatory parameter for telemedicine)
isInfantProduct	Sign that the services was bought for an infant (mandatory parameter for alfamedicine)

Response format matches GET api/reservation method response.

Response example

```
{
  "data": {
    "journey": {
      "outboundFlights": [{
```

```

    "aircraft": "32N",
    "referenceNumber": "1",
    "origin": "LED",
    "destination": "DME",
    "departureDate": "2020-01-22T14:35:00",
    "arrivalDate": "2020-01-22T16:25:00",
    "flightNumber": "60",
    "classOfService": "E",
    "operatingCarrier": "U6",
    "marketingCarrier": "U6"
  }
],
"outboundFareFamily": {
  "code": "ECOFLEXRU"
},
"origin": "LED",
"destination": "DME",
"departureDate": "2020-01-22T14:35:00"
},
"number": "VY9NCD",
"passengers": [{
  "id": 1,
  "hasInfant": false,
  "referenceNumber": "2",
  "firstName": "TEST",
  "middleName": "TESTOVICH",
  "surname": "TESTOV",
  "passengerType": 0,
  "birthDate": "1988-08-11T00:00:00",
  "sex": 0,
  "nationality": "RUS",
  "documentNumber": "14001232244",
  "documentExpiryDate": "2029-12-25T00:00:00",
  "email": "TEST@YANDEX.RU",
  "phone": "79510002233",
  "documentType": 0,
  "isAdult": true,
  "isChanged": false
}
],
"services": [],
"seats": [],
"partnerProducts": [{
  "partnerId": 1,
  "partnerType": 1,
  "bookingStatus": 1,
  "code": "TRAVEL",
  "amount": 847.00,
  "currency": "RUB",
  "passengerReference": "2",
  "isInfantProduct": false,
  "bookingId": "94cf3c93-b6aa-49c5-8de1-f3fa2749b5e2"
}, {
  "partnerId": 4,
  "partnerType": 2,
  "bookingStatus": 1,
  "code": "STANDARTOW",

```

```

    "amount": 500.00,
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "abf3bf5c-acc3-442d-98fa-df317ab439e2",
    "orderId": string
  }
],
"hasSpecialConditions": false
},
"errors": [],
"success": true
}

```

Examples of invalid requests and responses

Request	Response
<p>Invalid partner's code for TRAVEL service</p> <pre> { "products": [{ "partnerId": 4, "code": "TRAVEL" }, { "partnerId": 4, "code": "STANDARTOW" }] } </pre>	<pre> "partnerProducts": [{ "partnerId": 4, "partnerType": 2, "bookingStatus": 1, "code": "STANDARTOW", "amount": 500.00, "currency": "RUB", "passengerReference": "2", "isInfantProduct": false, "bookingId": "f4b21bb1-7a31-46c2-b6b4-5198632e5f7f" }], "hasSpecialConditions": false, "errors": { "partnerProducts": [{ "internalError": 5000, "internalErrorMessage": "Invalid product code", "partnerId": 4, "partnerType": 2, "bookingStatus": 0, "code": "TRAVEL", "amount": 0.0, "currency": "RUB" }] } </pre>

Invalid service code for Sealine partner

```
{
  "products": [
    {
      "partnerId":
2,
      "code":
"AVIAPL"
    },
    {
      "partnerId":
4,
      "code":
"STANDARTOW"
    }
  ]
}
```

```
"partnerProducts": [{
  "partnerId": 4,
  "partnerType": 2,
  "bookingStatus": 1,
  "code": "STANDARTOW",
  "amount": 500.00,
  "currency": "RUB",
  "passengerReference": "2",
  "isInfantProduct": false,
  "bookingId": "f4b21bb1-7a31-46c2-b6b4-
5198632e5f7f"
}
],
"hasSpecialConditions": false,
"errors": {
  "partnerProducts": [{
    "internalError": 5000,
    "internalErrorMessage": "Invalid product
code",
    "partnerId": 2,
    "partnerType": 1,
    "bookingStatus": 0,
    "code": "AVIAPL",
    "amount": 0.0,
    "currency": "RUB"
  }
]
}
```


Retrieve aircraft seatmap service

API service	seatmap
Short description	Allows to receive information about seatmap scheme in aircraft
Supported HTTP(S)-methods	GET
Conterminous systems methods used	Amadeus: Air_RetrieveSeatMap

GET seatmap method

Description: receives information about seatmap scheme in aircraft.

Request format: api/apiVersion/seatmap

Request sample:

api/apiVersion/seatmap?referenceNumber=1&origin=LED&destination=DME&departureDate=20190923&flightNumber=80&classOfService=E&commercialFamily=U6ECONOMY

Request parameters

Information about flight for which seatmap is needed

Parameter	Description
referenceNumber	Reference to flight segment
origin	Origin airport
destination	Destination airport
departureDate	Date and time of departure
flightNumber	Flight number
classOfService	Information about selected tariff
commercialFamily	Fare family type, for instance: U6ECONOMY or U6BUSINESS
currency	Currency in which search results will be returned

Response format

```
{
  "data": {
    "isEmpty": boolean,
    "compartmentDetails": [{
      "number": integer,
      "seats": [{
        "seat": string,
        "number": integer,
        "description": string,
        "characteristics": [string],
        "occupation": string,
        "passengerRelatedPrice": [{
          "name": string,
          "type": string,
          "uniqueCustomerIdentifier": string,
          "seatPrice": number,
          "isSeatUnavailable": boolean,
          "hasInfant": boolean
        }
      ]
    }],
    "classDesignator": string
  }
],
  "passengersTotalData": [{
    "passengerReference": string,
    "passengerPriceLevels": [{
      "price": decimal,
      "priceLevel": integer
    }
  ]
}],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
}],
  "success": boolean
}
```

Response parameter

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	isEmpty	Flag identifying whether data is returned from amadeus This field can be used to determine possibility of selecting seats on present flight segment
compartmentDetails		Array of objects with information about seats grouped by rows
	number	Row number
	seats	Array of objects which contains detailed information about seats
	seat	Seat. Possible values: A-F
	number	Row number for seat
	description	Seat description 9 – central seat A – aisle seat W – window seat
	characteristics	Seat characteristics Possible values: see Attachment 2. Most widely spread Seatmap seat characteristics
	occupation	Sign of seat's occupation Possible values: O – occupied
passengerRelatedPrice		Array of objects with information about seat prices for. Only available for seats with CH characteristics.
	name	Passenger name
	type	Passenger type Possible values: see EN Attachment 2. Passenger types
	uniqueCustomerIdentifier	unique passenger ID

			seatPrice	seat price
			isSeatUnavailable	Seat availability flag
			hasInfant	Flag identifying that passenger travels with infant
			classDesignator	Information about seat class. Possible values: ECONOMY, BUSINESS
			passengersTotalData	An array of objects with information about the cost of seats in the context of passengers
			passengerReference	Passenger reference
			passengerPriceLevels	An array of objects of the cost of seats available to the passenger
			price	Seat cost
			priceLevel	Seat value category
			errors	Information about errors appeared during query execution.
			ibeCode	IBE error code, see. EN Attachment 1. IBE error codes
			description	IBE error description, see EN Attachment 1. IBE error codes
			messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
			code	Amadeus error code
			text	Amadeus error text

Successful request and response examples

See [EN Attachment X. Request and response examples](#)

Failed request and response examples

```
{
  "errors": [{
    "ibeCode": 1113,
    "description": "Amadeus Air_RetrieveSeatMap function returned error.",
    "messages": [{
```

```
        "code": "99",
        "text": null
      }
    ]
  }
],
"success": false
}
```

From the Amadeus documentation: Code 99 is the error "Seat map not available for requested zone, seat may be requested", "Seat map is not available for the requested zone, you can request a seat."

Booking payment service

API service	payment
Short description	Service allows to obtain payment link from BPC system
Supported HTTP(S)-methods	GET, POST
Conterminous systems methods used	BPC: Obtain payment link - registerOrder Retrieve order information - getPaymentResult Retrieve information to display on the payment page - getPaymentJson Request for payment - performPayment CRM: Holding of unique promocode: UALPromotionCodesholdPromoCode

POST payment method

Description: forms link for redirection to booking payment form

Request format /api/apiVersion/payment

Request body

```
{  
  "lang": "string",  
  "redirectLang": "string",  
  "returnUrl": "string",  
  "callbackUrl": "string",  
  "engine": "string"  
}
```

Request parameters

Parameter	Description
-----------	-------------

lang	<p>Interface language for the BPC payment form.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ru - Russian; • en - English; • cn - Chinese.
redirectLang	<p>Redirection language to BMM after payment.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ru - Russian; • en - English; • zh - Chinese; • uz - Uzbek.
returnUrl	<p>Return address after payment. For example: https://service.uralairlines.ru/</p>
callbackUrl	Url for retrieve order status
engine	engine for BMM

Response format

```
{
  "data": {
    "orderId": string,
    "paymentFormURL": string
  },
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }]
},
"success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
orderId	Order ID
paymentFormURL	payment link Format: <a href="https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=<order_ID>&hash=<hash>&systemId=<system_id>&language=<language>">https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=<order_ID>&hash=<hash>&systemId=<system_id>&language=<language>
errors	Information about errors appeared during query execution.
ibeCode	IBE error code.
description	IBE error description
messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
code	Amadeus error code
text	Amadeus error text

Successful requests and response examples

```
{
  "data": {
    "orderId": "a2c0fa9a-9712-498e-b7f4-d10e21e43e68",
    "paymentFormURL":
"https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=a2c0fa
9a-9712-498e-b7f4-
d10e21e43e68&hash=7d0cb651e8b9bf767c8d1291d5c25b0fb1f9c58ac83b61ab1b4bb985350
5eeb0&systemId=mainweb&language=ch"
  },
  "errors": [],
  "success": true
}
```


POST payment/email method

Description: sends a letter to pay for the order to the mail specified when booking.

Request format /api/apiVersion/payment/email

Request body

```
{  
  
  { "lang": "string",  
  
    , "redirectLang": "string"  
  
}
```

Request parameters

Parameter	Description
lang	Interface language for the BPC payment form. Possible values: <ul style="list-style-type: none">• ru - Russian;• en - English;• cn - Chinese.
redirectLang	Redirection language to BMM after payment. Possible values: <ul style="list-style-type: none">• ru - Russian;• en - English;• zh - Chinese;• uz - Uzbek.

Response format

```
{  
  "errors": [{  
    "ibeCode": integer,  
    "description": string,  
  }]
```

```

    "messages": [{
      "code": string,
      "text": string
    }
  ]
},
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	IBE error code.
description	IBE error description
messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
code	Amadeus error code
text	Amadeus error text

Successful requests and response examples

```

{
  "errors": [],
  "success": true
}

```

Updated logic after purchase partner's services implementation

1. If at least one partner's service was added to booking, then POST registerOrder method should be invoked in order to obtain payment link.

Important: all SSR with partner's services should be passed to PBC in following format:

```

{

```

```

"ssrCodes": [{
  "id": string,
  "code": string
}]
"gds": string
}

```

Field	Description
ssrCodes	Array of objects, containing blocks with SSR for auxiliary services
id	Unique order Id
code	Unique auxiliary service code
gds	For partner's services GetOrderInfo is used as source of information about booking. Therefore gds = GOI

Example (Aeroexpress):

```

{
  "ssrCodes": [{
    "id": "B487BE18BB824F748E26C04AEE211B8C",
    "code": "STANDARTOW"
  }]
  "gds": "GOI"
}

```

2. If booking does not contain partner's services, then GET registerOrder method should be invoked. Described above structure should not be passed.

POST payment/googlePay method

Description: payment request with GooglePay token parameter

Request format /api/apiVersion/payment/googlePay

Request body

```

{
  "lang": "string",
  "paymentToken": "string"
}

```

Request parameter

Parameter	Description
-----------	-------------

lang	<p>Redirection language to BMM after payment.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ru - Russian; • en - English; • zh - Chinese.
paymentToken	GooglePay token

Response format

```
{
  "data": {
    "mobileErrorCode": string,
    "redirectUrl": "string",
    "mdOrder": "string",
    "acsParams": {
      "acsUrl": "string",
      "paReq": "string",
      "redirect": "string",
      "cReq": "string"
    }
  },
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
},
  "success": boolean
}
```

Response parameters

Parameter	Description
success	<p>Sign of query execution success.</p> <p>If errors array is empty, field value will be set to true</p>
data	Returned data.

	mobileErrorCode	Error code that returns from BPC. If the request finishes without errors then the value is "OK".
	redirectUrl	Redirection url.
	mdOrder	Unique id in payment system
	acsParams	Parameters block
	acsUrl	Redirection URL to emitent ACS
	paReq	auth for ACS
	redirect	Return URL
	cReq	Only for 3DS 2.0. Information about payment
errors		Information about errors appeared during query execution.
	ibeCode	IBE error code.
	description	IBE error description.
	messages	Array of objects, containing conterminous system's errors. Array might be empty. It is recommended to analyze ibeCode field.
	code	Amadeus error code.
	text	Amadeus error text.
	additionalInfo	Additional information about errors.
	ssrType	Ssr type.
	info	Information.

POST payment/applePay method

Description: payment request with ApplePay token parameter

Request format: /api/apiVersion/payment/applePay

Request body

```
{
  "lang": "string",
  "paymentToken": "string"
}
```

Request parameter

Parameter	Description
lang	Redirection language to BMM after payment. Possible values: <ul style="list-style-type: none"> • ru - Russian; • en - English; • zh - Chinese.
paymentToken	ApplePay token.

Response format

```
{
  "data": {
    "mobileErrorCode": string,
    "mdOrder": "string",
  },
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
},
],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data.
mobileErrorCode	Error code that returns from BPC. If the request finishes without errors then the value is "OK".
mdOrder	Unique id in payment system

errors		Information about errors appeared during query execution.
	ibeCode	IBE error code.
	description	IBE error description.
	messages	Array of objects, containing conterminous system's errors. Array might be empty. It is recommended to analyze ibeCode field.
	code	Amadeus error code.
	text	Amadeus error text.
	additionalInfo	Additional information about errors.
	ssrType	Ssr type.
	info	Information.

GET payment method

Description: allows to check booking payment status

Request format /api/apiVersion/payment

Query sample:

api/apiVersion/payment/?orderId=f3999077-86bf-495e-a48d-1d16ac49af5f

Request parameter

Parameter	Description
orderId	Checking order number

Response format

```
{
  "data": string,
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }]
}
```

```

    ]
  }
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Information about payment status If empty error is returned, then order is not payed
errors	Information about errors appeared during query execution.
ibeCode	IBE error code.
description	IBE error description.
messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
code	Amadeus error code
text	Amadeus error text

Successful requests and response examples

```

{
  "data": "VI411111X1111",
  "errors": [],
  "success": true
}

```

Failed requests and response examples

```

{
  "errors": [{
    "ibeCode": 1300,

```



```

        "description": "Unknown Payment gate error.",
        "messages": [{
            "code": "NO_PAYMENTS_IN_DB_BY_ORDER",
            "text": null
        }
    ]
},
"success": false
}

```

GET payment/info method

Description: retrieving the information that should be displayed on the payment page before making payment.

Request format /api/apiVersion/Payment/Info

Query sample:

/api/apiVersion/Payment/Info?lang=ru

Request parameter

Parameter	Description
lang	<p>Interface language for the BPC.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ru - Russian; • en - English; • cn - Chinese.

Response format

```

{
  "data": {
    "orderId": string,
    "errorCode": string,
    "mobileErrorCode": string,
    "status": string,
    "pnr": string,
    "currency": string,
    "remainMiliseocs": decimal,
    "amount": decimal,
    "language": string
  }
}

```

```

    },
    "errors": [{
        "ibeCode": integer,
        "description": string,
        "messages": [{
            "code": string,
            "text": string,
            "additionalInfo": {
                "ssrType": string,
                "info": string
            }
        }
    ]
}
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data.
orderId	OrderId in BPC.
errorCode	Error code that returns from BPC. If the request finishes without errors then the value is "OK" (it is the same with mobileErrorCode).
mobileErrorCode	Error code that returns from BPC. If the request finishes without errors then the value is "OK".
status	The status that returns from BPC.
pnr	Code of reservation.
currency	Three-letter currency code.
remainMilisecs	The remaining time in milliseconds during which payment will be available.
amount	The amount to be paid in the minimum currency units.
language	Two-letter language code.
errors	Information about errors appeared during query execution.
ibeCode	IBE error code.

	description	IBE error description.
	messages	Array of objects, containing conterminous system's errors. Array might be empty. It is recommended to analyze ibeCode field.
	code	Amadeus error code.
	text	Amadeus error text.
	additionalInfo	Additional information about errors.
	ssrType	Ssr type.
	info	Information.

Successful requests and response examples

```
{
  "data": {
    "orderId": "e65f9ade-dc0f-4f58-b6ed-c83254b34aea",
    "errorCode": "OK",
    "mobileErrorCode": "OK",
    "pnr": "UEX7OU",
    "currency": "RUB",
    "remainMilisecs": 1490243.0,
    "amount": 209900.0,
    "language": "ru"
  },
  "errors": [],
  "success": true
}
```

Failed requests and response examples

```
{
  "errors": [{
    "ibeCode": 1300,
    "description": "Unknown Payment gate error.",
    "messages": [{
      "code": "TRY_TO_PAY_AGAIN"
    }
  ]
},
],
"success": false
}
```


Authorise in CRM: api/user

API service	user
Short description	Allows to authorise as existing "Wings" loyalty program user in CRM
Supported HTTP(S)-methods	POST
Conterminous systems methods used	CRM: AuthorizeMember, GetEntity, GetClientHousehold

Method POST user

Description: authorisation in CRM

Request format /api/apiVersion/user

Request parameters

Parameter	Description
login	User login
password	User password
token	Auth token

Method allows to authorise either with login/pass or with token, returned by getSession CRM method.

Request examples

Authorisation with login and password

```
{"login":"1000761309","password":"Alex"}
```

Authorisation with token

1. Call getSession method with auth parameters. Receive auth token:

```
{  
  "Login": "1000761309",  
  "SessionId": "1626e097-35d3-4972-98fe-ae23dac23766"  
}
```

2. Authorise with login and token

```
{"login":"1000761309","token":"1626e097-35d3-4972-98fe-ae23dac23766"}
```

Response format

```
{
  "data": {
    "userId": string,
    "firstName": string,
    "firstNameLat": string,
    "middleName": string,
    "middleNameLat": string,
    "lastName": string,
    "lastNameLat": string,
    "birthDate": datetime,
    "email": string,
    "cellPhone": string,
    "homePhone": string,
    "address": {
      "city": string,
      "country": string,
      "region": string,
      "street": string,
      "zip": string
    },
    "documents": [{
      "number": string,
      "title": string,
      "lastUpdateDate": string,
      "documentType": integer,
      "countryCode": string,
      "countryName": string,
      "docId": string
    }
  ],
  "logins": [{
    "loginName": string,
    "loginType": string
  }
],
  "loyalties": [{
    "memberId": string,
    "level": string,
    "program": string,
    "programArt": string,
    "joinDate": datetime,
    "currentPoints": integer,
    "levelPoints": integer,
    "levelBonus": integer,
  }
]
```

```

"levelStartDate": date,
"lastFlightDate": date,
"cards": [{
  "active": boolean,
  "ffn": string,
  "name": string,
  "cardLevel": string,
  "cardType": string,
  "issueDate": datetime,
  "validFromDate": datetime,
  "validToDate": datetime
}],
"activeCard": {
  "active": boolean,
  "ffn": string,
  "name": string,
  "cardLevel": string,
  "cardType": string,
  "issueDate": datetime,
  "validFromDate": datetime,
  "validToDate": datetime
}
}],
"sex": integer,
"wingsLoyalty": {
  "memberId": string,
  "level": string,
  "program": string,
  "programArt": string,
  "joinDate": datetime,
  "currentPoints": integer,
  "levelPoints": integer,
  "levelBonus": integer,
  "levelStartDate": date,
  "lastFlightDate": string,
  "cards": [{
    "active": boolean,
    "ffn": string,
    "name": string,
    "cardLevel": string,
    "cardType": string,
    "issueDate": datetime,
    "validFromDate": datetime,
    "validToDate": datetime
  }],
  "activeCard": {
    "active": boolean,
    "ffn": string,
    "name": string,
    "cardLevel": string,
    "cardType": string,
    "issueDate": datetime,
    "validFromDate": datetime,
    "validToDate": datetime
  }
}
},

```

```

"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }]
}],
"success": boolean
}

```

Response parameters

Important! All data received from CRM and not validated on IBE side.

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
userId	User ID in CRM
firstName	Name (Cyrillic)
firstNameLat	Name (Latin)
middleName	Middle name (Cyrillic)
middleNameLat	Middle name (Latin)
lastName	Last name (Cyrillic)
lastNameLat	Last name (Latin)
birthDate	Date of birth Format: yyyy-MM-ddTHH:mm:ss
email	Email
cellPhone	Cellular phone number
homePhone	Home phone number
address	Object with information about user's address
city	City
country	Country

	region	Region
	street	Street
	zip	Zip
	documents	Array of objects with information about user's documents
	number	Document number
	title	Document title
	lastUpdateDate	Document's last update time
	documentType	Document type Possible values: Attachment 2. Document types
	countryCode	Country code where document was issued Possible values: EN Attachment 2. Countries
	countryName	Country name where document was issued For instance "Russian Federation" Specified on CRM side, may not match with values from countries dictionary.
	docId	Document's unique ID in CRM
	logins	Array of objects with information about available user's logins
	loginName	Login name
	loginType	Login type Possible values: email, phone, other
	loyalties	Array of objects with information about loyalty programs in which user is participating
	memberId	Member ID
	level	Program level
	program	Program
	programArt	Program vendor code
	joinDate	Join date
	currentPoints	Current Points

	levelPoints	Service field
	levelBonus	Service field
	levelStartDate	Level start date Format: dd.MM.yyyy
	lastFlightDate	Date of last flight
	cards	Array of Card objects with information about user's cards
	activeCard	Card object with information about user's active card
	sex	User's sex Possible values: Attachment 2. Passenger sex
	wingsLoyalty	Object with information about "Wings" loyalty program
	memberId	Member ID
	level	Level
	program	Program
	programArt	Program vendor code
	joinDate	Join date
	currentPoints	Current points
	levelPoints	Service field
	levelBonus	Service field
	levelStartDate	Level start date
	lastFlightDate	Date of last flight
	cards	Array of Card objects with information about user's cards
	activeCard	Card object with information about user's active card
	errors	Sign of query execution success. If errors array is empty, field value will be set to true
	ibeCode	IBE error code, see Attachment 1. IBE error codes
	description	IBE error description, see see Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.

	code	Amadeus error code
	text	Amadeus error text

Card

Parameter	Description
active	Sign of card's activity
ffn	Card number
name	Loyalty program's user name
cardLevel	Card level
cardType	Card type
issueDate	Card issue datetime Format: yyyy-MM-ddTHH:mm:ss
validFromDate	Valid from date Format: yyyy-MM-ddTHH:mm:ss
validToDate	Valid to date Format: yyyy-MM-ddTHH:mm:ss

Examples of successful and failed requests.

EN Attachment X. Request and response examples

Retrieve information about linked users service (CRM)

API service	households
Short description	Allows to retrieve information about linked users (family in the sky)
Supported HTTP(S)-methods	GET
Conterminous systems methods used	CRM: GetClientHousehold

IMPORTANT: functionality is deprecated on CRM side and is no longer used.

GET households method

Description: allows to retrieve list of linked users for authorised user.

Request format /api/apiVersion/households

Request parameters

Additional parameters are not required. Information is taken from from booking context by session ID.

Response format

```
{
  "data": [{
    "userId": string,
    "firstName": string,
    "firstNameLat": string,
    "middleName": string,
    "middleNameLat": string,
    "lastName": string,
    "lastNameLat": string,
    "birthDate": datetime
  }
],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
}
],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Array of objects with information about linked users
userId	user ID in CRM
firstName	first name (Cyrillic)
firstNameLat	first name (Latin)
middleName	middle name (Cyrillic)
middleNameLat	middle name (Latin)
lastName	last name (Cyrillic)
lastNameLat	last name (Latin)
birthDate	date of birth Format: yyyy-MM-ddTHH:mm:ss
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Successful request and response examples

see [Attachment X. Request and response examples](#)

Failed request and response examples

401 Unauthorised error will be returned by method if X-Session header is absent or unable to find specific session or user in present booking context is not authorised in CRM.

GET households/id method

Description: allows to retrieve information about specific linked user

Request format /api/apiVersion/households/id

Request example

api/apiVersion/households/1-1PIR4T7

ID is taken from data.userId field in api/households response.

Response format

Matches response format of api/user method.

Promo code management service (CRM)

API service	promotions
Short description	Allows to add and delete promotional code for present booking
Supported HTTP(S)-methods	POST, DELETE
Conterminous systems methods used	CRM: UALPromotionCodescheckPromoCode

POST promotions method

Description: allows to add promotional code to booking. One promotional code for flight and one for a service can be applied at the same time.

Request format /api/apiVersion/promotions

Request body

```
{
  "number": string,
  "origin": string,
  "destination": string,
  "departureDate": date,
  "returnDate": date,
  "passengersCount": integer
}
```

Request parameters

Parameter	Description
number	Code number
origin	Origin airport
destination	Destination airport
departureDate	Departure date Format: yyyy-MM-dd
returnDate	Return date Format: yyyy-MM-dd

passengersCount	Passengers count
-----------------	------------------

Response format

```
{
  "data": {
    "number": string,
    "amount": number,
    "promotionStatus": integer,
    "promotionType": integer,
    "description": string,
    "startFlightDate": datetime,
    "endFlightDate": datetime,
    "startDate": datetime,
    "endDate": datetime,
    "isUnique": boolean,
    "campaignId": string,
    "directions": [{
      "origin": string,
      "destination": string
    }
  ],
  "services": [{
    "code": string
  }
  ],
  "families": [{
    "code": string
  }
  ],
  "excludedFlights": [{
    "flightNumber": string,
    "departureAirport": string,
    "arrivalAirport": string
  }
  ],
  "excludedPeriods": [{
    "startDate": datetime,
    "endDate": datetime
  }
  ],
  "loyaltyCardIsRequired": boolean,
  "hasConflict": boolean
},
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }
  ]
}
}
```



```

    ],
    "success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
number	Promotional code number
amount	Discount amount
promotionStatus	Promotional code status Possible values: see Attachment 2. Promocode statuses
promotionType	Code status 0 – applied to flight 1 – applied to certain service
description	Promotion campaign description
startFlightDate	Start flight date for promotional code
endFlightDate	End flight date for promotional code
startDate	Promotional campaign start date
endDate	Promotional campaign end date
isUnique	sign of promotional code's uniqueness
campaignId	Promotional campaign id
directions	Directions of exceptions/inclusions
origin	Origin airport
destination	Destination airport
services	Array of objects with information about services to which promotional code can be applied

	code	service code Possible values: see Attachment 2. Auxiliary services types (except insurance)
	families	Array of objects with information about tariffs to which promotional code can be applied
	code	tariff code
	excludedFlights	General exceptions - flights where discounts can not be applied
	flightNumber	Flight number
	departureAirport	Departure airport
	arrivalAirport	Arrival airport
	excludedPeriods	Promotional code exclusion period
	startDate	Start date of the promotional code exclusion period
	endDate	Expiration date of the promotional code exclusion period
	loyaltyCardIsRequired	Flag denoting a promo code for the loyalty card holder
	hasConflict	sign of success adding promotional code to booking False if there appeared error during promotional code adding.
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects containing errors from CRM response
	code	CRM error code
	text	CRM error text

Successful requests and responses examples.

See [Attachment X. Request and response examples](#).

Failed requests and responses examples

```
{
  "errors": [{
    "ibeCode": 1205,
```

```

    "description": "CRM CheckPromoCode function returned error.",
    "messages": [{
      "code": "undefined",
      "text": "Promotion code not found or canceled"
    }]
  }
],
"success": false
}

```

DELETE promotions method

Description: allows to delete previously added promotional code.

Request format /api/apiVersion/promotions

Request examples

/api/promotions?number=oGVy099w

Request parameters

Parameter	Description
number	Promotional code number which needs to be deleted

Response format

```

{
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }],
}
]
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects containing errors from CRM response
code	CRM error code
text	CRM error text

Example of failed response and requests

If deleted promocode is not exist method will return 404 Not found error.

Error codes and error texts of promotional codes validation service.

CRM Error	Text in engine
code: "1", text: "Promocode not found"	RUS: Промокод не найден ENG: Promocode not found
code: "2", text: "Early purchase is not available for these flight dates" или code: "2", text: "Promotional code is not suitable for this return date" или code: "2", text: "Promotional code is not suitable for this flight date"	RUS: Данный промокод не может быть применен к перелетам с указанной датой ENG: This promotional code is not suitable for this flight date
code: "3", text: "It's not possible to apply a promotional code to a given destination"	RUS: Данный промокод не может быть применен для указанного направления ENG: It is not possible to apply this promotional code to a given destination

code: "4", text: "Promotional code is suitable for return flight"	RUS: Промокод можно применить только купив билеты туда-обратно ENG: This promotional code is suitable for return flight
code: "5", text: "This promotional code can not be applied to this number of passengers"	RUS: Данный промокод не может быть применен к данному количеству пассажиров ENG: This promo code is not suitable for this number of passengers
code: "6", text: "City of registration is not correct" или code: "6", text: "Loyalty card number for search return flight not passed" или code: "6", text: "The holder of this loyalty card can not apply the promotional code" или code: "6", text: "Date of registration is not correct"	RUS: Ваша карта лояльности не подходит для указанного промокода ENG: Your loyalty card can not be applied to this promotional code
code: "7", text: "This promotional code is available for purchase N ticket(s)"	RUS: Билеты по данному промокоду закончились ENG: Tickets for this promo are over
code: "8", text: "The entered personal promotional code is not available for this member"	RUS: Введенный персональный промокод недоступен для Вашего аккаунта ENG: The entered personal promotional code is not available for your account
code: "9", text: "This promocode is only for cardholders. Enter card number"	RUS: Данный промокод доступен только для держателей карт. Пожалуйста, введите номер карты ENG: This promocode is only for cardholders. Please, enter card number

Discount retrieving service

API service	discounts
Short description	Allows to retrieve information about discounts for certain flight directions
Supported HTTP(S)-methods	GET
Conterminous systems methods used	CRM: CalculateDiscount, CalculateDiscountList2

POST discounts method

Description: requests information from CRM about discounts for certain flights directions.

Request format /api/apiVersion/discounts

Request body

```
{
  "origin": string,
  "destination": string,
  "departureDate": datetime,
  "returnDate": datetime,
  "adults": integer,
  "children": integer,
  "pensioners": integer,
  "youths": integer,
  "subsidizedChildren": integer,
  "infants": integer,
  "fareFamilies": [
    string
  ],
  "flights": [
    {
      "flightId": integer,
      "flightNumber": string,
      "departureAirport": string,
      "arrivalAirport": string,
      "bookingClass": string,
      "referenceNumber": string
    }
  ]
}
```

Request parameters

Parameter	Description
origin	Original airport, IATA-code
destination	Destination airport, IATA-code
departureDate	Departure date
returnDate	Return date
adults	number of adult passengers
children	number of child passngers
pensioners	number of senior citizens
youths	number of passengers of "Youth" type
subsidizedChildren	Number of children eligible for subsidies
infants	number of infants
fareFamilies	An array with information about tariff families
flights	An array of objects with information about flights
flightId	Flight ID
flightNumber	Flight number
departureAirport	Departure airport
arrivalAirport	Arrival airport
bookingClass	Booking class. In general, this is the first character of fareBasis
referenceNumber	Flight segment reference number

Response format

```
{
  "data": {
    "id": string,
    "name": string,
    "amount": integer,
    "discountType": integer,
    "usageType": integer,
    "startDate": datetime,
    "endDate": datetime,
    "fareFamilies": [
```

```

        string
    ]
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ]
}
],
"success": true
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
id	Unique discount campaign ID in CRM
name	CRM discount campaign name
amount	Discount amount
discountType	Discount type 0 – percent 1 – absolute (RUB)
usageType	Discount application type 0 – tariff discount 1 – order discount
startDate	Start date of the discount campaign
endDate	Expiration date of the discount campaign
fareFamilies	An array with information about tariff families
errors	Information about errors appeared during query execution.

	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects containing errors from CRM response
	code	CRM error code
	text	CRM error text

Itinerary receipt receive and send to email service

API Service	receipt
Short description	Allows to receive Itinerary receipt for present booking and send it to email.
Supported HTTP(S)-methods	GET, POST
Conterminous systems methods used	CRM: Receipt/PDF, Receipt/Send

GET receipt method

Description: method requests Itinerary receipt from CRM for present booking in context. Booking information should be retrieved via GET api/reservation method before calling GET api/receipt.

Request format /api/apiVersion/receipt

Request parameters

No parameters

Response format

PDF-file with itinerary receipt encoded in base64.

Failed request and response examples

If booking information is not found within context

```
{
  "errors": [
    {
      "ibeCode": 1001,
      "description": "Operation can not be performed due to invalid
reservation state.",
      "messages": []
    }
  ],
  "success": false
}
```

Timeout of itinerary receipt retrieve.

```
{
```

```

"errors": [{
  "ibeCode": 1402,
  "description": "CRM receipt timeout exceeded.",
  "messages": []
}]
,
"success": false
}

```

POST receipt method

Description: method allows to send itinerary receipt for present booking in context to email. Booking information should be retrieved via GET api/reservation method before calling POST api/receipt.

Request body

```

{
  "email": string
}

```

Request parameters

Parameter	Description
email	email address to which itinerary receipt needs to be sent

Response format

```

{
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }],
  "success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Failed request and response examples

Error sending of itinerary receipt

```
{
  "errors": [{
    "ibeCode": 1401,
    "description": "CRM receipt send email service returned error.",
    "messages": [{
      "text": "Receipt service responded with status Forbidden"
    }]
  }]
},
"success": false
}
```

Voluntary tickets and services refund service

API Service	refund
Short description	Allows to calculate refund sum and perform tickets and services refund
Supported HTTP(S)-methods	POST
Conterminous systems methods used	<p>Amadeus:</p> <p>Receive information about booking - PNR_Retrieve, Ticket_RetrieveListOfTSM, Ticket_ProcessEDoc, PNR_AddMultiElements</p> <p>Initiate refund - DocRefund_InitRefund, DocRefund_IgnoreRefund, DocRefund_ProcessRefund</p> <p>Cancel of booking - PNR_Cancel</p> <p>CRM:</p> <p>Create refund request - ServiceReqInsert_Input</p> <p>GOI:</p> <p>Receive VAT - GET payment/pnr</p>

POST refund/calculate method

Description: allows to receive breakdown of refund sums for each passenger.

Request format: /api/apiVersion/refund/calculate

Request parameters

Parameter	Mandadatory	Description
pnrNumber	yes	PNR number, unique 6-digit booking identifier
passengers	yes	Array of passengers for which refund sums needs to be calculated
lastName	yes	Passenger's last name

passengerReferenceNumber	yes	Reference to passenger Linked with data → passengers → reference Number
luggageSeats	no	Array with information about luggage with separate air ticket and ssr CBBG.
referenceNumber	no	Reference to CBBG luggage. Linked with GET Reservation data → luggageSeats → referenceNumber
passengerType	yes	Passenger type. Possible values: see Attachment 2. Passenger types

Request format

Refund for adult passenger

```
{
  "pnrNumber": "OWDCZO",
  "passengers": [
    {
      "lastName": "TESTOV",
      "passengerReferenceNumber": 2,
      "luggageSeats": [],
      "passengerType": 0
    }
  ]
}
```

Refund for adult and infant

```
{
  "pnrNumber": "OWCXEV",
  "passengers": [
    {
      "lastName": "TESTOV",
      "passengerType": 0,
      "luggageSeats": [],
      "passengerReferenceNumber": "3"
    },
    {
      "lastName": "TESTOV",
      "passengerType": 3,

```

```

        "luggageSeats": [],
        "passengerReferenceNumber": "2"
    }
]
}

```

Response format

```

{
  "data": {
    "passengerRefundDetails": [{
      "hasInfant": boolean,
      "passengerType": integer,
      "referenceNumber": string,
      "surname": string,
      "ticketRefundDetails": {
        "isInfantTicket": boolean,
        "commissionAmount": integer,
        "refundBonuses": integer,
        "refundCurrency": integer,
        "refundTotal": integer,
        "taxDetailsInformation": [{
          "taxCategory": string,
          "taxDetails": [{
            "rate": string,
            "currencyCode": string,
            "type": string
          }
        ]
      }
    ]
  },
  "ticketNumber": string
},
"luggageSeatsRefundDetails": [{
  "hasInfant": boolean,
  "passengerType": integer,
  "referenceNumber": string,
  "surname": "CBBG",
  "ticketRefundDetails": {
    "isInfantTicket": boolean,
    "commissionAmount": integer,
    "refundBonuses": integer,
    "refundCurrency": integer,
    "refundTotal": integer,
    "taxDetailsInformation": [{
      "taxCategory": string,
      "taxDetails": [{
        "rate": string,
        "currencyCode": string,
        "type": string
      }
    ]
  }
}
]
},
],

```

```

        "ticketNumber": string
    },
    "luggageSeatsRefundDetails": [],
    "serviceRefundDetails": [],
    "partnerProducts": []
}
],
"serviceRefundDetails": [{
    "code": string,
    "refundBonuses": integer,
    "refundCurrency": integer,
    "refundTotal": integer,
    "taxDetailsInformation": [{
        "taxCategory": string,
        "taxDetails": [{
            "rate": string,
            "currencyCode": string,
            "type": string
        }
    ]
}
]
},
    "ticketNumber": string
}
],
"partnerProducts": [{
    "document": string,
    "refundAmount": integer,
    "isRefundAmountUnknown": boolean,
    "partnerId": integer,
    "partnerType": integer,
    "bookingStatus": integer,
    "code": string,
    "amount": integer,
    "currency": string,
    "passengerReference": string,
    "isInfantProduct": boolean,
    "bookingId": string
}
]
},
    "refundWithTicketOnlyServices": [
        "string"
    ]
},
"errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
]
}
],
"success": boolean

```


}

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	passengerRefundDetails	Array of objects with refund details information for each passenger.
	hasInfant	Flag identifying whether passenger travels with infant
	passengerType	Passenger type Possible values: Attachment 2. Passenger types
	referenceNumber	Reference to passenger
	surname	Passenger's surname
	ticketRefundDetails	Array of objects with refund details information
	isInfantTicket	Flag identifying whether ticket is infant's document
	commissionAmount	Commission amount
	refundBonuses	Number of refunded "Wings" bonuses
	refundCurrency	Number of refunded roubles
	refundTotal	Total refund sum
	ticketNumber	ATC number Ticket number should contain 13 digits

				taxDetailsInformation	<p>Services information about tariffs and fees.</p> <p>Full parameter's description information can be find in Amadeus documentation.</p> <p>Important: Information from present segment can be ignored.</p> <p>refundBonuses and refundCurrency values should be displayed to a user.</p>
				taxCategory	Tax category
				taxDetails	Array of objects with information about tariffs and fees details.
				rate	Tariff/fee sum
				currencyCode	Currency code
				type	Tariff/fee type
				luggageSeatsRefundDetails	<p>Array with information about CBBG luggage with separate air ticket to refund.</p> <p>All array elements are the same with passengerRefundDetails array elements, the field "surname" has value "CBBG".</p>
				serviceRefundDetails	Array of objects with service's refund information.
				code	<p>Service code</p> <p>Possible values: see Attachment 2. Auxiliary services types</p>
				refundBonuses	Number of refunded "Wings" bonuses
				refundCurrency	Number of refunded roubles
				refundTotal	Total refund sum
				ticketNumber	EMD number
				taxDetailsInformation	Services information about tariffs and fees.
				taxCategory	Tax category
				taxDetails	Array of objects with information about tariffs and fees.
				rate	Tariff/fee sum

				currencyCode	Currency code
				type	Tariff/fee type
		partnerProducts			Array of objects with partner's products refund information
			amount		Partner's service price
			bookingId		Unique partner's service order id
			bookingStatus		Partner's services order status
			code		Partner's service tariff code. See <u>Attachment 4. Partner's codes.</u>
			currency		Currency code
			document		Partner's service document number. For insurances - issued insurance policy number.
			isInfantProduct		Flag identifying whether service is issued for an infant.
			isRefundAmountUnknown		Flag identifying whether partner's service refund sum is unknown.
			partnerId		Partner's id. See <u>Attachment 4. Partner's services.</u>
			partnerType		Partner's type. See <u>Attachment 4. Partner's services.</u>
			passengerReference		Reference to passenger for whom partner's service is being refunded
			refundAmount		Refundable sum (in RUR)
errors					Information about errors appeared during query execution.
			ibeCode		Error code, see EN_Attachment 1. IBE error codes
			description		Code description, see. EN_Attachment 1. IBE error codes
		messages			Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
			code		Amadeus error code
			text		Amadeus error text

POST refund/partial method

Description: allows you to refund tickets for booking in the context of passengers.

Request format: /api/apiVersion/refund/partial

Response parameters:

Parameter	Mandadatory	Description
pnrNumber	yes	PNR number, unique 6-digit booking identifier
passengers	yes	An array of passengers for which refund sums needs to be calculated
isTicketRefund	yes	The sign that a ticket with services or only services is being returned
partnerProducts	no	An array of partner services to be returned (the array parameters are similar to the apiVersion/partners/id response parameters)
lastName	yes	Passenger's last name
passengerReferenceNumber	yes	Passenger reference. Associated with field data -> passengers -> referenceNumber
luggageSeats	no	Array with information about luggage with separate air ticket and ssr CBBG.
passengerType	yes	Passenger type. Possible types of passengers are presented in Attachment 2.
services	no	An array with information about purchased and refundable additional services of the airline
code	no	Returned additional service code
ticketNumber	no	Ticket number of the returned additional service

Response format: the same as the refund/calculate method, the crmRefundRequestId field is added with the number of the created request for a refund to CRM.

Response options (short description):

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
passengerRefundDetails	Array with refund information Full analogue of data in response of api/refund/calculate method
crmRefundRequestId	Refund request number in CRM. Example: 1-46MJI9D.
errors	Information about errors appeared during query execution.

Requests examples

Example of requesting a full refund of a ticket with partner services

```
{
  "pnrNumber": "RZRTVO",
  "lastName": "TESTOV",
  "passengers": [{
    "lastName": "TESTOV",
    "passengerType": 0,
    "passengerReferenceNumber": "2",
    "isTicketRefund": true,
    "partnerProducts": [{
      "partnerId": 1,
      "partnerType": 1,
      "bookingId": "9e8450f2-c07a-4730-9bbb-07c0931aeb53",
      "code": "AVIADYNAMIC"
    }
  ],
  "services": [{
    "ticketNumber": "2624552270079",
    "code": "PETC"
  }
]
}
```

Example of request for refund of partner services without a ticket

```
{
  "pnrNumber": "RZROML",
  "lastName": "TESTOV",
```

```

"passengers": [{
  "lastName": "TESTOV",
  "passengerType": 0,
  "passengerReferenceNumber": "2",
  "isTicketRefund": false,
  "partnerProducts": [{
    "partnerId": 1,
    "partnerType": 1,
    "bookingId": "9940f823-4bb8-4c7c-a485-7341551c3285",
    "code": "AVIADYNAMIC"
  }
]
}
]
}

```

An example of a request for a refund of additional airline services without a ticket

```

{
  "pnrNumber": "RZRTVO",
  "lastName": "TESTOV",
  "passengers": [{
    "lastName": "TESTOV",
    "passengerType": 0,
    "passengerReferenceNumber": "2",
    "isTicketRefund": true,
    "partnerProducts": [{
      "partnerId": 1,
      "partnerType": 1,
      "bookingId": "9e8450f2-c07a-4730-9bbb-07c0931aeb53",
      "code": "AVIADYNAMIC"
    }
  ],
  "services": [{
    "ticketNumber": "2624552270079",
    "code": "PETC"
  }
]
}

```

Description of voluntary refund process

Voluntary refund is initiated by airline's client own will.

Short description of voluntary refund business process.

1. Client opens BMM (booking management module) using PNR number and passenger's last name.
2. Selects option "Return tickets", pop-up window gets opened with information about passengers existing in booking.

3. Selects a Voluntary refund.
4. Selects passengers for whom refund is needed to be performed and clicks on "Calculate amount" (for refund).
5. api/refund/calculate method is called, refund details data is displayed to client (refund bonuses, sum, fines etc). Actual refund is not performed at this step.
6. Client selects "Confirm refund" option, api/refund/partial method is called.
7. The IBE service performs a refund of services and/or air tickets to the API of partners and Amadeus
8. The IBE service validates the possibility of registering an auto return order.
9. The IBE service registers an order for return (using a separate method for registering a return order in the PM. Registration is always on the original PNR for which the payment was made).
10. The PM checks the possibility of auto-refund for the current order: it requests a GET api/refund, validates the contents of the order and the amounts to be refunded.
11. The PM sends a refund request to the bank.
12. The PM sends the status of the request for a refund: Option A - the order is accepted by the bank for return; Option B - the order is not accepted for return, or an error has been received from the PM
13. Then option A:
 1. The IBE service informs the user about a successfully completed refund request with information that the funds will be credited to the card within some time, depending on the bank
 2. The PM fiscalizes the return receipt. The check is sent to the client.
 3. When you re-enter the MUB, information is displayed that the order has been returned.
14. Then option B:
 1. The IBE service generates a request for a refund in CRM.
 2. If the return fails, one of the possible error messages is displayed.
 3. The IBE service informs the user about a successfully created refund request with information about the request number.
 4. Within 30 days, the request in CRM is processed manually by the operator, and the amount to be returned is requested from the engine.
 5. After financial refund operation is performed, confirmation is sent to IBE (information about refund completion is being recorded to DB).
 6. Refund considered to be complete.

General technical information.

1. For each refund (initiated by api/refund method call) a record with unique refund session ID is created in DB.
2. 3 entities can be linked with refund session: a) tickets refund information entity, b) airline's services refund information entity c) partner's services refund information entity
3. Refund session can have several states. During refund process IBE transfers session from one state to another performing following operations:

1. Initiated - refund session is initiated;
 2. VatCalculated - VAT information is calculated for refund;
 3. PartnerServicesCanceled - partners services cancellation performed if it is necessary;
 4. PnrSplitted - PNR is split into several parts (in case of partial refund);
 5. CrmRequestCreated - Refund request is created in CRM;
 6. PnrCanceled - PNR and tickets are cancelled (irreversible operation). In case of partial refund child PNR and its tickets gets cancelled.
 7. ManualAnalysis - an error appeared in voluntary refund process. See point 5.
4. Entities with information about tickets, airline's and partner's services also has several states:
1. NotAvailable - entity is not available for refund sum information retrieve;
 2. Available - refund sum information can be retrieved for a ticket/service;
 3. Processed - refund is confirmed;
 4. ManualProcessingRequired - manual processing is required for this refund;
 5. Failed - refund confirmation end with an error
5. Deferred refund session
- If an error appears during refund procedure and operation hangs in one of intermediate states (before PnrCanceled state) then special job periodically attempts to continue refund process.
- If refund session does not transfer to terminal state, during allotted time, IBE logs an error and transfers session to ManualAnalysis state.
- Error gets handled in Azure Insights and email notification is being sent for this event. Manual analysis of this situation is performed afterwards.

For retrieving of refund session information and refund confirmation GET refund/pnr and PUT refund/pnr methods of GOI-proxy is being used.

These methods are described in GOI documentation.

Voluntary refund of non-refundable tariff tickets: promo-light, promo, bagless.

This type of refund is a variety of conventional voluntary refund and applies for tickets with non-refundable tariff: promo-light, promo and bagless.

Refund process is similar to conventional voluntary refund with several exceptions.

1. Client opens BMM (booking management module) using PNR number and passenger's last name.
2. Selects option "Return tickets", pop-up window gets opened with information about passengers existing in booking.
3. Selects passengers for whom refund is needed to be performed and clicks on "Calculate amount" (for refund).

4. api/refund/calculate method is called. **Message is displayed "Your ticket is non-refundable. Only airport taxes and cost of auxiliary services for unused flights are refundable"**
Refund sum for tickets is not displayed. ticketRefundDetails.refundTotal = 0 is returned.
 If there are no auxiliary services in booking then "Refund total:" block is not displayed as well.
 If auxiliary services exists in booking then only auxiliary services refund sum is displayed in "Refund total:" block (without breakdown).
 refund details data is displayed to client (refund bonuses, sum, fines etc). Actual refund is not performed at this step.
5. Client selects "Confirm refund" option, api/refund method is called.
6. In case of successful refund request creation, information with voluntary refund request ID in CRM is displayed to client.
7. In case of failure one of possible errors is displayed to client.
8. During 30 days refund request is processed manually by CRM operator. During this procedure refund sum gets queried from IBE.

Failed request and response examples

Description	Request format	Response format
<p>There is no information about refund request in CRM in api/refund response. crmRefundRequestId field is empty.</p> <p>It is considered that refund hangs in one of intermediate states. Additional actions for such refund is not required. Refund either will be completed automatically or manual analysis will be performed after several attempts to complete process in deferred mode.</p>	-	-

<p>Booking for 2 adults.</p> <ol style="list-style-type: none"> 1. Refund is initiated for 1st adult but is not completed yet. 2. Attempt to refund for 2nd passenger is initiated. 	<p>Request for point 2.</p> <pre>{ "pnrNumber": "OY2P8Q", "passengers": [{ "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 0 }] }</pre>	<pre>{ "errors": [{ "ibeCode": 1033, "description": "There are incomplete refund processes.", "messages": [] }], "success": false }</pre>
<p>Lets consider that in OY2P8Q booking adult passenger (of type 0) is added and correct reference to passenger - 2.</p> <p>An error will appear in case of:</p> <ol style="list-style-type: none"> 1. Incorrect reference to passenger is passed. 2. Incorrect type of passenger is passed. 	<p>Request for point 1.</p> <pre>{ "pnrNumber": "OY2P8Q", "passengers": [{ "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 0 }] }</pre> <p>Request for point 2.</p> <pre>{ "pnrNumber": "OY2P8Q", "passengers": [{ "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 1 }] }</pre>	<pre>{ "errors": [{ "ibeCode": 1041, "description": "Some passengers are not exist in reservation.", "messages": [] }], "success": false }</pre>

<p>Booking for 1 adult and 1 child</p> <p>It is not possible to perform refund only for adult passenger. Error "The remaining passengers are of the wrong type." will appear.</p>	<pre>{ "pnrNumber": "OWWDII", "passengers": [{ "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 0 }] }</pre>	<pre>{ "errors": [{ "ibeCode": 1042, "description": "The remaining passengers are of the wrong type.", "messages": [] }], "success": false }</pre>
---	---	--

Involuntary refund of tickets service. Simplified scheme.

Api service	InvoluntaryRefund
Short description	Allows to create involuntary refund request
Supported HTTP(S)-methods	POST
Conterminous systems methods used	<p>Amadeus: PNR_Retrieve, PNR_AddMultiElements</p> <p>For moneyless involuntary refund: DocRefund_InitRefund, DocRefund_IgnoreRefund, DocRefund_ProcessRefund, PNR_Cancel</p> <p>CRM: Create refund request - ServiceReqInsert_Input</p>

POST InvoluntaryRefund method

Description: In simplified scheme no any operations is performed with PNR in Amadeus. Method allows to create refund request in CRM.

Request format: /api/apiVersion/InvoluntaryRefund

Request parameters:

Parameter	Mandatory	Description
pnrNumber	yes	PNR number, unique 6-digit booking identifier
lastName	yes	Passenger's last name
passengers	yes	Array of objects with information about passengers.
lastName	yes	Passenger's last name
passengerReferenceNumber	yes	Reference to passenger Relates to field data → passengers → referenceNumber
passengerType	yes	Passenger's type. Possible values: see Attachment 2. Dictionaries

refundReason	yes	1 - due to flight cancellation 2 - other reason
contact	yes	
firstName	yes	Passenger's first name
surname	yes	Passenger's last name
middleName	no	Passenger's middle name
email	yes	Passenger's email
phone	yes	Passenger's phone number
comment	yes	Applies only if refundReason = 2 . Comment field in which passengers specifies reason of involuntary refund.

Request examples:

1. "Due to flight cancellation" case:

```
-----WebKitFormBoundaryjZgbkEAPdqmWbi3R
Content-Disposition: form-data; name="refundDetails"

{
  "pnrNumber":"URSOXM",
  "lastName":"KOLTSOVA",
  "passengers":
  [
    {"lastName":"KOLTSOVA","passengerType":0,"passengerReferenceNumber":"3"},
    {"lastName":"KOLTSOVA","passengerType":4,"passengerReferenceNumber":"3"},
    {"lastName":"KOLTSOVA","passengerType":3,"passengerReferenceNumber":"2"}
  ],
  "refundReason":1,
  "contact":
  {
    "firstName":"Ольга",
    "surname":"Кольцова",
    "middleName":"Михайловна",
    "email":"koltsovaolga5@gmail.com",
    "phone":"89959965000"
  }
}
```

2. **"Other reason"** case. In this case comment field should be filled and document file uploaded, confirming reason of involuntary refund.

```
-----WebKitFormBoundary56NzvwpTRorAE6Pc
Content-Disposition: form-data; name="files"; filename="photo_2019-09-10_12-27-09.jpg"
Content-Type: image/jpeg
```

```
-----WebKitFormBoundary56NzvwpTRorAE6Pc
Content-Disposition: form-data; name="refundDetails"

{
  "pnrNumber": "URSOXM",
  "lastName": "KOLTSOVA",
  "passengers":
  [
    {
      "lastName": "KOLTSOVA",
      "passengerType": 4,
      "passengerReferenceNumber": "3",
    }
  ],
  "refundReason": 2,
  "comment": "Не смогла явиться на рейс по причине болезни",
  "contact":
  {
    "firstName": "Ольга",
    "surname": "Кольцова",
    "middleName": "Михайловна",
    "email": "koltsovaolga5@gmail.com",
    "phone": "88120007676"
  }
}
```

Response parameters (short form):

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
crmRefundRequestId	Refund request number in CRM. Example: 1-46MJI9D.

autoRefundCompleted	Sign of autorefund processing
errors	Information about errors appeared during query execution.

Simplified involuntary refund scheme description

1. Client opens BMM (booking management module) using PNR number and passenger's last name.
2. Selects option "Return tickets", switched to "Involuntary" tab.
3. Selects refund reason. By default "Other reason" is set. In this case client should describe reason of involuntary refund and upload document, confirming refund reason. In case of "Due to flight cancellation" it is not necessary.
4. Selects passengers for whom refund is needed to be performed.
5. Fill personal data fields: first/last name, phone number, email.
6. Client selects "Confirm refund" option, api/InvoluntaryRefund method is called.
7. In case of successful creation of refund request, information with refund request ID in CRM is displayed to client.
8. In case of failure one of possible errors is displayed to client.

No refund sum calculation is performed.

No history of involuntary requests is stored in IBE DB.

There is no control of request execution on operator's side.

There is no limit of amount of refund requests formed for single PNR.

POST involuntaryRefund/moneyless method

Description: let to process involuntary refund without money refund

Request format: /api/apiVersion/InvoluntaryRefund/Moneyless

Request parameters:

Parameter	Mandatory	Description
pnrNumber	yes	Unique booking number (PNR).
passengers	yes	Array with information about passengers whom needs to process the involuntary moneyless refund
isTicketRefund	yes	The sign that a ticket with services or only services is being returned.

Parameter	Mandatory	Description
partnerProducts	no	An array of partner services to be returned (the array parameters are similar to the apiVersion/partners/id response parameters)
lastName	yes	Passenger's last name
passengerReferenceNumber	yes	Passenger reference. Associated with field data -> passengers -> referenceNumber
passengerType	yes	Passenger type. Possible types of passengers are presented in Attachment 2.
services	no	An array with information about purchased and refundable additional services of the airline.
code	no	Returned additional service code.
ticketNumber	no	Ticket number of the returned additional service.

Response format: is the same with POST api/refund/partial method response.

Response parameters:

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
passengerRefundDetails	Array with refund information Full analogue of data in response of api/refund/calculate method
crmRefundRequestId	Refund request number in CRM. Example: 1-46MJI9D.
errors	Information about errors appeared during query execution.

POST involuntaryRefund/Calculate method

Description: gets information about involuntary refund information about passengers.

Request format: /api/apiVersion/InvoluntaryRefund/Calculate

Request parameters:

Parameter	Mandatory	Description
passengers	yes	Array with information about passengers whom needs to process the involuntary moneyless refund
passengerReferenceNumber	yes	Passenger reference. Associated with field data -> passengers -> referenceNumber
passengerType	yes	Passenger type. Possible types of passengers are presented in Attachment 2.

Response format: is the same with POST api/refund/partial method response.

Response parameters:

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
passengerRefundDetails	Array with refund information Full analogue of data in response of api/refund/calculate method
crmRefundRequestId	Refund request number in CRM. Example: 1-46MJI9D.
refundWithTicketOnlyServices	List of services that can be returned only if the ticket is returned
errors	Information about errors appeared during query execution.

Change passenger's data service

API service	passenger
Short description	Allows to change passenger's data in booking
Supported HTTP(S)-methods	PUT
Conterminous systems methods used	Amadeus: PNR_Retrieve, PNR_ChangeElement, PNR_NameChange, PNR_AddMultiElements, PNR_Cancel

Important: at present personal data in partner's services is not changing when changes in booking's data is performed.

Method should not be used if there are partner's services exist in booking. Service is planned to be modernised.

PUT passenger method

Description: allows to change passenger's personal data in booking

Implemented logic:

1. The fields "Last name", "Name" and "Middle name" it isn't allowed to change.
2. The fields "Sex", "Birth date", "Document number", "E-mail", "Phone" it is allowed to change one time.
3. The field "Birth date" it is allowed to change in according to chosen passenger type.

Request format: /api/apiVersion/passenger

Request parameters

Parameter	Mandatory	Description
referenceNumber	yes	Reference to passenger
passengerType	yes	Passenger's type
surname	no	Last name
firstName	no	First name
middleName	no	Middle name
email	no	Email
phone	no	Phone

nationality	no	Nationality
sex	no	Sex
birthDate	no	Date of birth
documentNumber	no	Document number
callbackUrl	no	Url for retrieve payment status
engine	no	engine for BMM
lang	no	Interface language for the BPC payment form. Possible values: <ul style="list-style-type: none"> • ru - Russian; • en - English; • cn - Chinese.
redirectLang	no	Redirection language to BMM after payment. Possible values: <ul style="list-style-type: none"> • ru - Russian; • en - English; • zh - Chinese; • uz - Uzbek; • ky - Kyrgyz; • fr - French.
returnUrl	no	Return address after payment.

Response format

```
{
  "data": {
    "orderId": string,
    "paymentFormURL": string,
  },
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }]
}
```

```

    }],
    "success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
orderId	Order ID
paymentFormURL	Payment link Format: <a href="https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=<order_ID>&hash=<hash>&systemId=<system_id>&language=<language>">https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=<order_ID>&hash=<hash>&systemId=<system_id>&language=<language>
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Request and response examples

Description	Request example	Response example
-------------	-----------------	------------------

<p>Mandatory parameter referenceNumber is not passed</p>	<pre>{ "surname": "Test" }</pre>	<pre>{ "errors": [{ "ibeCode": 1008, "description": "Data validation error.", "messages": [{ "code": "ReferenceNumber", "text": "'Reference Number' should not be empty." }] }], "success": false }</pre>
<p>Invalid value of passengerType is passed</p>	<pre>{ 2, "referenceNumber": "passengerType": 1 }</pre>	<pre>{ "errors": [{ "ibeCode": 1082, "description": "Passenger not found in reservation", "messages": [] }], "success": false }</pre>

<p>Validation when birthdate is invalid for chosen passenger type</p>	<pre>{ "referenceNumber": 2, "passengerType": 0, "birthDate": "2019- 02-01" }</pre>	<pre>{ "errors": [{ "ibeCode": 1008, "description": "Data validation error.", "messages": [{ "text": "Passenger age is incorrect" }] }], "success": false }</pre>
---	---	---

Valid request	<p>Passenger in booking</p> <pre>{ "id": 1, "hasInfant": false, "referenceNumber": "2", "firstName": "TEST", "middleName": "TEST", "surname": "TEST", "passengerType": 0, "birthDate": "1989-10-20T00:00:00", "sex": 1, "nationality": "RUS", "documentNumber": "7894561231", "documentExpiryDate": "2031-12-26T00:00:00", "loyaltyCard": "1000761775", "email": "test.test@test.test.test", "phone": "79112111193", "documentType": 0, "isAdult": true, "isChanged": false, "checkedIn": { "1": false }, "checkInHasStarted": { "1": false }, "hasPassengersDataErrors": false, "passengerDataErrors": [] }</pre> <p>Request</p> <pre>{ "passengerType": "0", "referenceNumber": "2", "documentNumber": "7894561232", "email": "test@example.com", "phone": "423365788", "sex": "Male", "birthDate": "2000-12-26", "lang": "en", "redirectLang": "en" }</pre>	<pre>{ "data": { "id": 1, "hasInfant": false, "referenceNumber": "2", "firstName": "ALEKSANDRA", "middleName": "KONSTANTINOVNA", "surname": "TESTOVA", "passengerType": 0, "birthDate": "1988-08-13T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "14152220011", "documentExpiryDate": "2029-07-16T00:00:00", "email": "TESTOVA@GMAIL.COM", "phone": "79510002233", "documentType": 0, "isAdult": true, "isChanged": true }, "errors": [], "success": true }</pre>
---------------	--	---

Online check-in service

API service	Checkin, Checkin/GetPassengerChekinStatus
Short description	Allows to check online check-in for your flight from the booking management module and, if available, go to TAIS
Supported HTTP-(S) methods	GET
Used methods of related systems	TAIS: CheckInGetData

GET Checkin method

Description: allows to get data on the possibility of registration with the CheckInGetData service of the TAIS system.

If check-in is available, refunds the time remaining until the end of check-in for the departure of the first outbound/return flight.

Request format /api/apiVersion/checkin

Request parameters

Parameter	Description
lastName	Last name of one of the passengers added to the booking. The parameter is passed in latin.
pnrNumber	PNR number for which information is needed
ticketNumber	Passenger ticket number

Response format

```
{
  "data": {
    "canRegister": true,
    "remainingTime": 5905,
    "passengers": [
      {
        "ticketNumber": "2622439494689",
        "registered": false
      },
      {
        "ticketNumber": "2622439494692",
```



```

        "registered": false
    }
]
},
"errors": [],
"success": true
}

```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
data	Online check-in information
canRegister	Sign of the availability of online registration
remainingTime	Time left before the end of online registration in seconds
passengers	Passenger list
ticketNumber	Passenger ticket number
registered	Sign of registration
errors	Information about errors that occurred during the operation of the service
ibeCode	Error code, see Attachment 1 IBE error codes
description	Error description, see Attachment 1 IBE error codes
messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by field ibeCode.
code	Amadeus error code
text	Amadeus error text

Examples of successful responses

Response when check-in is open and the second passenger is already checked-in

```

{
  "data": {
    "canRegister": true,
    "remainingTime": 5905,
    "passengers": [

```

```

    {
      "ticketNumber": "2622439494689",
      "registered": false
    },
    {
      "ticketNumber": "2622439494692",
      "registered": true
    }
  ]
},
"errors": [],
"success": true
}

```

Response when registration is not open yet:

```

{
  "data": {
    "canRegister": false,
    "remainingTime": 115380,
    "passengers": []
  },
  "errors": [],
  "success": true
}

```

Response when registration is already closed:

```

{
  "data": {
    "canRegister": false,
    "remainingTime": 0,
    "passengers": []
  },
  "errors": [],
  "success": true
}

```

An example of an unsuccessful response

```

{
  "errors": [
    {
      "ibeCode": 1700,
      "messages": [
        {
          "code": "NOT_FOUND",
          "text": ""
        }
      ]
    }
  ]
}

```

```

    ]
  }
],
"success": false
}

```

GET Checkin/GetPassengerChekinStatus method

Description: allows to get data on the possibility of registration in the context of each segment from the CheckInGetData service of the TAIS system.

Request format /api/apiVersion/Checkin/GetPassengerChekinStatus

Request parameters:

Parameter	Description
lastName	Last name of one of the passengers added to the booking. The parameter is passed in latin.
pnrNumber	PNR number for which information is needed
ticketsNumberCommaSeprated	An array of ticket numbers of all passengers in the booking

Response format

```

{
  "data": [
    {
      "ticketNumber": string,
      "registered": boolean
    }
  ],
  "errors": [{
    "ibeCode": string,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
}
],
"success": boolean
}

```

Response parameters

Parameter	Description
data	An array of objects with information about the availability of passenger check-in
ticketNumber	Passenger ticket number
registered	Sign of availability of registration
errors	Information about errors that occurred during the operation of the service
ibeCode	Error code, see Attachment 1 IBE error codes
description	Error description, see Attachment 1 IBE error codes
messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by field ibeCode.
code	Amadeus error code
text	Amadeus error text
success	Sign of the success of the request. If the errors array is empty, then the field value will be true

Cancellation service

API service	annul
Short description	Allows to cancel tickets and receive information on canceled bookings
Supported HTTP-(S) methods	GET, POST
Используемые методы смежных систем	<p>Amadeus:</p> <p>Getting information on booking - PNR_Retrieve, Ticket_RetrieveListOfTSM, Ticket_ProcessEDoc, PNR_AddMultiElements</p> <p>Cancel reservation - PNR_Cancel</p>

POST api/annul method

Description: allows to cancel tickets for all passengers in the booking

Request format: /api/apiVersion/Annul

Request parameters: X-Session is passed in the header

Response format:

```
{
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
}],
  "success": boolean
}
```

Response parameters:

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true

Parameter		Description
errors		Information about errors that occurred during the operation of the service.
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from Amadeus' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
	code	Amadeus error code
	text	Amadeus error text

GET api/annul method

Description: allows to get information on booking cancellation

Request parameters: X-Session is passed in the header

Parameter	Obligation	Description
pnrNumber	yes	PNR number, unique six-character booking ID

Response format:

```
{
  "data": [{
    "pnrNumber": string,
    "createdDate": datetime
  }],
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }],
  "success": boolean
}
```

Response parameters:

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
data	The data returned by the request
pnrNumber	PNR number
createdDate	Date and time when the cancellation was issued
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code, see Attachment 1. IBE error codes
description	Error description, see Attachment 1. IBE error codes
messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Amadeus error code
text	Amadeus error text

Service for obtaining a catalog of services without PNR

API service	CatalogueStandalone
Short description	The service allows to receive information about available additional services without creating a booking
Supported HTTP-(S) methods	POST
Used methods of related systems	Amadeus: Service_StandaloneCatalogue

POST CatalogueStandalone method

Description: receives information about the catalog of available additional services without a pre-made reservation

Request format: api/apiVersion/CatalogueStandalone

Request body:

```
{
  "flights": {
    "outboundFlights": [{
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "arrivalDate": datetime,
      "flightNumber": string,
      "classOfService": string,
      "marketingCarrier": string,
      "operatingCarrier": string,
      "flightIndex": integer
    }
  ],
  "returnFlights": [{
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "arrivalDate": datetime,
    "flightNumber": string,
    "classOfService": string,
    "marketingCarrier": string,
    "operatingCarrier": string,
    "flightIndex": integer
  }
],
  "fares": {
    "outboundFareFamily": {
      "code": string,
```

```

        "category": string,
        "isSubsidized": boolean
    },
    "returnFareFamily": {
        "code": string,
        "category": string,
        "isSubsidized": boolean
    },
    "fares": [{
        "passengerType": string,
        "flightIndex": integer,
        "fareBasis": string,
        "isBreakpoint": boolean
    }
    ],
    "passengers": [{
        "id": integer,
        "firstName": string,
        "middleName": string,
        "surname": string,
        "passengerType": string,
        "birthDate": datetime,
        "sex": string,
        "loyaltyCard": string,
        "disability": {
            "isDeaf": boolean,
            "isBlind": boolean,
            "hasGuideDog": boolean,
            "assistanceType": integer,
            "wheelchairType": integer,
            "commentary": string
        },
        "hasInfant": boolean,
        "nationality": string
    }
    ],
    "currency": string
}

```

Request parameters

Parameter	Description
flights	Object with information about flights
outboundFlights	An array of StandaloneInputFlight objects with information about flights in the "outbound" direction
returnFlights	An array of StandaloneInputFlight objects with information about flights in the "return" direction

Parameter	Description
fares	Tariff information object
outboundFareFamily	Object with information on fares for "outbound" directions
code	Fare family code. Data is taken from api/flights response: data.fareFamilies.code
category	The category of the tariff family. Data is taken from api/flights response: data.fareFamilies.category
isSubsidized	Sign of a subsidized tariff.
returnFareFamily	Object with information on fares for the "return" route
code	Fare family code. Data is taken from api/flights response: data.fareFamilies.code
category	The category of the tariff family. Data is taken from api/flights response: data.fareFamilies.category
isSubsidized	Sign of a subsidized tariff.
fares	Array with information about tariffs for directions "outbound" and "return"
passengerType	Passenger type. Possible types are shown in the guide "Passenger Types" in Attachment 2.
flightIndex	Segment number for a specific flight (numbered from 1)
fareBasis	Base rate name. The data is taken from the api/flights response: data.recommendations.outboundFareDetails.fareBasis and data.recommendations.returnFareDetails.fareBasis
isBreakpoint	An indication of whether the tariff expires on the current segment. The data is taken from the api/flights response: data.recommendations.outboundFareDetails.isBreakpoint and data.recommendations.returnFareDetails.isBreakpoint

Parameter	Description
passengers	Object with passenger information
id	Passenger's serial number (numbered from 1) (required parameter)
firstName	Passenger's name
middleName	Passenger's patronymic
surname	Passenger's surname
passengerType	Passenger type (required parameter) Possible values: dictionary of passenger types
birthDate	Date of birth
sex	Passenger gender Possible values: dictionary "gender of passengers"
loyaltyCard	Loyalty card number "Wings"
disability	Information about people with disabilities
isDeaf	Hearing impairment
isBlind	Visual impairment
hasGuideDog	A guide dog must be accompanied, must be sent only together with the isBlind sign
assistanceType	Passenger escort type Possible values: dictionary of maintenance types
wheelchairType	Wheelchair type Possible Values: Wheelchair type dictionary
commentary	Comment (optional)
hasInfant	An indication that this passenger has a baby
nationality	Pax nationality
currency	Search currency

StandaloneInputFlight

Parameter	Description
origin	Departure airport
destination	Arrival airport
departureDate	<p>Date and time of departure.</p> <p>The data is taken from the api/flights response:</p> <p>data.outboundFlights.flights.departureDateTime</p> <p>or</p> <p>data.returnFlights.flights.departureDateTime</p> <p>Format: yyyy-MM-ddTHH:mm:ss</p> <p>Example: 2018-06-20T01:00:00</p>
arrivalDate	<p>Date and time of arrival.</p> <p>The data is taken from the api/flights response:</p> <p>data.outboundFlights.flights.arrivalDateTime</p> <p>or</p> <p>data.returnFlights.flights.arrivalDateTime</p> <p>Format: yyyy-MM-ddTHH:mm:ss</p> <p>Example: 2018-06-20T01:00:00</p>
flightNumber	<p>Flight number.</p> <p>The data is taken from the api/flights response:</p> <p>data.outboundFlights.flights.flightNumber</p> <p>or</p> <p>data.returnFlights.flights.flightNumber</p>

Parameter	Description
classOfService	Information about the selected tariff. The data is taken from the api/flights response: data.recommendations.outboundFareDetails.bookingCode or data.recommendations.returnFareDetails.bookingCode
marketingCarrier	Marketing airline For example: U6, S7
operatingCarrier	The airline actually performing the flight For example: U6, OK
flightIndex	Segment number for a specific flight (numbered from 1)

Response format

```
{
  "data": {
    "services": [{
      "code": string,
      "flightReferences": [{
        "referenceNumber": string,
        "passengerPrices": [{
          "referenceNumber": string,
          "amount": number
        }]
      }]
    }],
    "packs": [{
      "includedSeats": [string],
      "excludedSeats": [string],
      "includedServiceCode": string,
      "code": string,
      "flightReferences": [{
        "referenceNumber": string,
        "passengerPrices": [{
          "referenceNumber": string,
          "amount": decimal
        }]
      }]
    }]
  }
}
```

```

    ]
  }
],
"seatServices": [{
  "seatCharacteristics": [string],
  "code": string,
  "flightReferences": [{
    "referenceNumber": string,
    "passengerPrices": [{
      "referenceNumber": string,
      "amount": number
    }
  ]
}
]
}
],
"petsServiceConfiguration": [{
  "petsList": [string],
  "flightReferences": [string]
}
],
"insuranceProducts": [{
  "code": string,
  "amount": number
}
],
"partnerProducts": [{
  "partnerId": "Unknown",
  "partnerType": "Unknown",
  "bookingStatus": "Unknown",
  "code": string,
  "amount": number,
  "currency": string,
  "passengerReference": string,
  "isInfantProduct": boolean,
  "bookingId": string,
  "internalError": integer,
  "internalErrorMessage": string
}
],
"portions": [{
  "type": string,
  "number": string,
  "flightReferenceNumbers": [string]
}
],
"serviceCategories": [{
  "code": string,
  "category": string,
  "order": integer,
  "fullDirection": boolean,
  "onlyForRussianCitizens": bool
}
],
"serviceOptions": [{
  "code": string,

```

```

        "preselect": boolean,
        "flightReferences": [string],
        "category": string,
        "priority": integer
    }
    ],
    "moneyWallCoefficient": string
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string,
        "additionalInfo": {
            "ssrType": string,
            "info": string
        }
    }
    ]
}
]
},
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
data	The data returned by the request
services	An array of objects with information about available airline services
code	Service code Possible values: dictionary of additional services
flightReferences	An array with information about the segments and passengers for which the services are available, as well as the cost of these services.
referenceNumber	Flight segment reference
passengerPrices	An array of objects with information about the passenger and the cost of the service
referenceNumber	Reference to a specific passenger
amount	Service cost for a specific passenger

Parameter		Description
	packs	An array with available service packages
	includedSeats	An array of characteristics of available seats within the service package
	excludedSeats	An array of characteristics of unavailable places within the service package
	includedServiceCode	Included service code
	code	Service package code. Possible values: Attachment 2. Dictionary of service packages
	flightReferences	The arrays with information about the position and passengers, which services are available, as well as the cost of these services.
	referenceNumber	Flight segment reference
	passengerPrices	An array of objects with information about the passenger and the cost of the service
	referenceNumber	Reference to a specific passenger
	amount	Service cost for a specific passenger
	serviceOptions	An array with information about which services should be automatically preselected on the front-end of the booking engine.
	category	Service category
	code	Service package code. Possible values: Attachment 2. Dictionary of service packages
	flightReferences	An array with information about segments for which services are available
	preselect	Service field. If true, then the service should be automatically preselected in the UI. <i>Note: The logic was originally added for use in the Web client.</i>

Parameter		Description
	priority	<p>Service field. Used in conjunction with the preselect field.</p> <p>Indicates the priority of automatically adding a service to the UI, 0 is the highest.</p> <p>For example, if a service with a priority of 0 is not available for pre-selection, then a service with a priority of 1 will be added, etc.</p>
	seatServices	<p>An array of objects with information about the available seats for selection on the flight.</p> <p>IMPORTANT: The api/seatmap service returns more complete information about available seats and should be used to book a seat selection service.</p>
	seatCharacteristics	<p>Array of site characteristics.</p> <p>Possible values: Attachment 2. Dictionary of characteristics of places</p>
	code	Service code, SEAT
	flightReferences	An array with information about the segments and passengers for which the services are available, as well as the cost of these services.
	referenceNumber	Flight segment reference
	passengerPrices	An array of objects with information about the passenger and the cost of the service
	referenceNumber	Reference to a specific passenger
	amount	Service cost for a specific passenger
	petsServiceConfiguration	An array of objects that contain information about the availability of the animal transport service on different segments. It is necessary to control the transportation of animals at the tariffs of the "business" family.
	petsList	An array of strings, contains a list of animals available for transport
	flightReferences	An array of strings, contains a list of segments on which the service is available
	insuranceProducts	An array of objects with information about available insurances for selection

Parameter		Description
	code	Insurance code Possible values: dictionary of additional services (category insurance)
	amount	Insurance cost
serviceCategories		An array of objects, contains information about services and the order in which they are displayed on the site.
	code	Service code Possible values: dictionary of additional services
	category	Service category
	order	The order of the additional services on the page. 1 - top of the page
	fullDirection	Flag indicating whether the service covers the entire flight
partnerProducts		An array of objects, contains information about partner services.
	partnerId	Unique ID of the partner in the IBE engine. Alpha insurance - 1, Sealine insurance - 2, Aeroexpress - 4
	partnerType	Partner type. Insurance - 1, Aeroexpress - 2
	code	Partner service code. Each partner has unique service codes. See the codes for partner services in Attachment 5.
	amount	Partner's service cost
	currency	Tariffication currency.

Parameter		Description
	bookingStatus	Partner service status. Possible values: 0 - unknown, 1 - the service has not been released, 2 - the service has been released, 3 - the release ended with an error, 4 - a return has been made, 5 - the service cannot be released
	passengerReference	Reference to the passenger for whom the service has returned
	isInfantProduct	Flag indicating whether the service is provided to an infant
	bookingId	Unique Id of the partner service order in this booking
	internalError	Error code when receiving partner service
	internalErrorMessage	Error message when receiving partner service
	portions	Service field
	type	Service field
	number	Service field
	flightReferenceNumbers	Service field
	moneyWallCoefficient	Coefficient for calculating the monthly installment payment
	errors	Information about errors that occurred during the operation of the service.
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from Amadeus' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
	code	Amadeus error code
	text	Amadeus error text
	additionalInfo	An array with additional error information
	ssrType	Ssr type
	info	Additional information

Examples of successful requests and responses

Attachment X. Examples of responses from services

Certificate management service

API service	certificate
Short description	The service allows to receive information about certificates and delete them for the current booking
Supported HTTP(S)-methods	GET, DELETE
Methods of related systems used	no

GET api/certificate method

Description: Obtaining information about a certificate

Request format /api/apiVersion/certificate

Request parameters: X-Session is passed in the header

Parameter	Obligation	Description
certNumber	yes	Certificate number EMD-S
surname	yes	Surname of the owner of the certificate

Response format

```
{
  "data": {
    "number": string,
    "surname": string,
    "amount": integer,
    "notValidAfter": datetime
  },
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
}
],
"success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
data	The data returned by the request
number	Certificate number
surname	Surname of the owner of the certificate
amount	Certificate amount in RUB
notValidAfter	Certificate expiration date
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code, see Attachment 1. IBE error codes
description	Error description, see Attachment 1. IBE error codes
messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Amadeus error code
text	Amadeus error text

DELETE api/certificate method

Description: allows to remove the applied certificate.

Request format /api/apiVersion/certificate

Request parameters

None. Cancellation reservations are identified by the session ID passed in the X-Session header.

Response format

{

```

"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }
]
},
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code, see Attachment 1. IBE error codes
description	Error description, see Attachment 1. IBE error codes
messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
code	Amadeus error code
text	Amadeus error text

GET api/certificate/CheckCertificate method

Description: Obtaining information about a certificate

Request format /api/apiVersion/certificate/CheckCertificate

Request parameters:

Parameter	Obligation	Description
certNumber	yes	Certificate number EMD-S
surname	yes	Surname of the owner of the certificate

Response format

```
{
  "data": {
    "documents": [
      {
        "number": string,
        "surname": string,
        "status": string,
        "notValidAfter": "datetime,
        "amount": integer,
        "currency": string
      }
    ]
  },
  "errors": [],
  "success": boolean
}
```

Response parameters

Parameter	Description
data	The data returned by the request
documents	Certificate information
number	Certificate number
surname	Surname of the owner of the certificate
status	Certificate status (open - open, close - closed)
notValidAfter	Date until which the certificate is valid
amount	Certificate amount
currency	Currency (RUB)
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code, see Attachment 1. IBE error codes
description	Error description, see Attachment 1. IBE error codes
messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Amadeus error code
text	Amadeus error text

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true

POST api/certificate method

Description: Creating new certificate.

Request format /api/apiVersion/certificate

Request format

```
{
  "amount": decimal,
  "giver": {
    "firstName": string,
    "lastName": string,
    "phone": string,
    "email": string
  },
  "gifted": {
    "firstName": string,
    "lastName": string,
    "phone": string,
    "email": string
  }
}
```

Request parameters:

Parameter	Description
amount	Certificate amount
giver	Information about the person who buys a certificate.
firstName	First name of the giver of the certificate.
lastName	Last name of the giver of the certificate.
phone	Phone of the giver of the certificate.
email	Email of the giver of the certificate.
gifted	Information about the certificate holder.
firstName	First name of the certificate holder.
lastName	Last name of the certificate holder.

	phone	Phone of the certificate holder.
	email	Email of the certificate holder.

Response format

```
{
  "data": {
    "pnr": string,
    "surname": string,
    "amount": decimal,
    "currency": string,
    "notValidAfter": datetime
  },
  "errors": [],
  "success": boolean
}
```

Response parameters

Parameter		Description	
data		The data returned by the request	
	pnr	Unique booking number	
	surname	Last name of the certificate holder.	
	amount	Certificate amount.	
	currency	Certificate currency.	
	notValidAfter	Date until which the certificate is valid.	
errors		Information about errors that occurred during the operation of the service.	
	ibeCode	Error code, see Attachment 1. IBE error codes.	
	description	Error description, see Attachment 1. IBE error codes.	
	messages	An array of objects containing errors from Amadeus answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.	
		code	Amadeus error code.
		text	Amadeus error text.

success	Sign of the success of the request. If the errors array is empty, then the field value will be true.
---------	---

Service for getting information about the weather

API service	weather
Short description	Service allows to get information about the weather in the city in the request
Supported HTTP(S)-methods	GET

Methods available for the version 2.3

Method POST get/weather

Description: allows to get information about the weather in the city in the request

Request format /api/apiVersion/weather

Request parameters

Parameter	Description
airport	IATA airport code

Request example

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'
/api/v2.3/weather?airport=SVX

Response body

```
{
  "data": {
    "type": string,
    "typeText": string,
    "icon": string,
    "iconImage": string,
    "temp": string,
    "pressure": integer,
    "humidity": integer,
    "wind": {
```

```

    "speed": double,
    "deg": integer,
    "gust": double,
    "direction": string
  },
  "city": {
    "id": string,
    "code": string,
    "serviceName": string
  },
  "success": boolean
},
"errors": [
  {
    "ibeCode": integer,
    "description": string,
    "messages": [
      {
        "text": string
      }
    ]
  }
]
},
"success": boolean
}

```

Response parametres

Параметр	Описание
data	The data returned by the request
type	Weather type
typeText	Textual representation of the weather type
icon	Weather icon
iconImage	Path to icon
temp	Temperature
pressure	Pressure
humidity	Humidity
wind	Wind
speed	Wind speed
deg	Direction In degrees
gust	Gust of wind

Параметр	Описание
direction	Direction
city	City
id	City ID
code	City code
serviceName	City name
success	Sign of success of the request
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code, see Attachments 1. IBE error codes
description	Error description, see Attachments 1. IBE error codes
messages	An array of objects containing errors. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Error code
text	Error text
success	Sign of success of the request If the errors array is empty, then the field value will be true.

Response body example

```
{
  "data": {
    "type": "clouds",
    "typeText": "Облачно",
    "icon": "02d",
    "iconImage": "/images/weather/02d.svg",
    "temp": "20°C",
    "pressure": 1002,
    "humidity": 77,
    "wind": {
      "speed": 2.0,
      "deg": 170,
      "gust": 0.0,
      "direction": "ЮЖНЫЙ"
    },
  },
  "city": {
    "id": "202",
    "code": "SVX",
    "serviceName": "Ekaterinburg"
  },
  "success": true
}
```

```
},  
"errors": [],  
"success": true  
}
```


Service for checking participation of passengers in the loyalty program

API service	services/searchContact
Short description	Service to check the participation of passengers in the loyalty program.
Поддерживаемые HTTP(S)-методы	POST
Используемые методы смежных систем	CRM: get_SearchContact

POST api/services/searchContact method

Description: let to check the participation of the passengers in reservation in the loyalty program.

Request format /api/apiVersion/services/searchContact

Request body:

```
{
  "docNumbers": [string]
}
```

Request parameters

Parameter	Description
docNumbers	The array of document numbers of passengers in reservation.

Response format

```
{
  "documentNumber": string,
  "cardNumber": string,
  "tier": string,
  "email": string,
  "firstName": string,
  "lastName": string,
  "middleName": string
}
```

Response parameters

Parameter	Description
documentNumber	The number of document of the passenger.
cardNumber	The number of loyalty card.
tier	The level of card.
email	E-mail of the passenger.
firstName	The first name of the passenger.
lastName	The last name of the passenger.
middleName	The middle name of the passenger.

Examples of successful requests and responses

```
{
  "data": [{
    "documentNumber": "1234561231",
    "cardNumber": "1000761776",
    "email": "test.test@test.ru",
    "firstName": "Test",
    "lastName": "Test",
    "middleName": "Test"
  }
],
"errors": [],
"success": true
}
```

Examples of failed requests and responses

The empty docNumbers array in the request.

```
{
  "errors": [{
    "ibeCode": 1008,
    "description": "Data validation error.",
    "messages": [{
      "code": "DocNumbers",
      "text": "'Doc Numbers' must not be empty."
    }
  ]
},
],
"success": false
}
```

Services for working with queues (for ADP environment).

Service for placing PNR to queues.

API service	queue/placePNR
Short description	The service allows placing PNR in queues.
Supported HTTP-(S) methods	POST
Used methods of related systems	Amadeus: Queue_PlacePNR

POST queue/placePNR method

Description: placing the PNR in queues

Request format /api/apiVersion/queue/placePNR

Request parameters are the same as request parameters of Amadeus service Queue_PlacePNR

Response format

```
{
  "data": [{
  }],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
}
],
  "success": boolean
}
```

Response parameters are the same as response parameters of Amadeus service Queue_PlacePNR, except the cases when the error appears in IBE and the errors array is consists of detailed information about errors.

Examples of successful requests and responses

Request

```
{
  "Queue_PlacePNR": {
```

```

    "placementOption": {
      "selectionDetails": {
        "option": "QEQ"
      }
    },
    "targetDetails": [{
      "targetOffice": {
        "sourceType": {
          "sourceQualifier1": "3"
        },
        "originatorDetails": {
          "inHouseIdentification1": "SVXU621AA"
        }
      },
      "queueNumber": {
        "queueDetails": {
          "number": "5"
        }
      },
      "categoryDetails": {
        "subQueueInfoDetails": {
          "identificationType": "C",
          "itemNumber": "0"
        }
      }
    }
  ],
  "recordLocator": {
    "reservation": {
      "controlNumber": "{{pnr}}"
    }
  }
}

```

Response

```

{
  "data": {
    "queue_PlacePNRReply": {
      "recordLocator": {
        "reservation": {
          "controlNumber": "U800XI"
        }
      }
    }
  },
  "errors": [],
  "success": true
}

```

Examples of failed requests and responses

Response

```

{
  "data": {
    "queue_PlacePNRReply": {
      "errorReturn": {

```

```

        "errorDefinition": {
            "errorDetails": {
                "errorCode": "91D"
            }
        },
        "errorText": {
            "freeText": [
                "CHECK FORMAT"
            ]
        }
    }
}
},
"errors": [],
"success": true
}

```

The description of Amadeus error codes is on the documentation for Queue_PlacePNR service.

Service for retrieving the list of changes

API service	queue/list
Short description	The service allows retrieving the list of changes from queues.
Supported HTTP-(S) methods	POST
Used methods of related systems	Amadeus: Queue_List

POST queue/list method

Description retrieving the list of changes

Request format /api/apiVersion/queue/list

Request parameters are the same as request parameters of Amadeus service Queue_List

Response format

```

{
  "data": [{
  }],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,

```

```

        "info": string
      }
    ]
  },
  "success": boolean
}

```

Response parameters are the same as response parameters of Amadeus service Queue_List, except the cases when the error appears in IBE and the errors array is consists of detailed information about errors.

Examples of successful requests and responses

Request

```

{
  "Queue_List": {
    "queueNumber": {
      "queueDetails": {
        "number": "1"
      }
    },
    "categoryDetails": {
      "subQueueInfoDetails": {
        "identificationType": "2",
        "itemNumber": "6"
      }
    }
  }
}

```

Response

```

{
  "data": {
    "queue_ListReply": {
      "queueView": {
        "agent": {
          "originatorDetails": {
            "inHouseIdentification1": "SVXU621AA"
          }
        },
        "queueNumber": {
          "queueDetails": {
            "number": 1
          }
        },
        "categoryDetails": {
          "subQueueInfoDetails": {
            "identificationType": "C",
            "itemNumber": "6"
          }
        },
        "date": {
          "timeMode": 1
        }
      }
    }
  }
}

```

```

},
"pnrCount": [{
  "quantityDetails": {
    "numberOfUnit": 30
  }
}, {
  "quantityDetails": {
    "numberOfUnit": 1
  }
}, {
  "quantityDetails": {
    "numberOfUnit": 30
  }
}
],
"item": [{
  "paxName": {
    "paxDetails": {
      "surname": "TEST",
      "type": "0",
      "quantity": 0,
      "quantitySpecified": true
    }
  },
  "recLoc": {
    "reservation": {
      "controlNumber": "T4ALKF"
    }
  },
  "agent": {
    "originatorDetails": {
      "inHouseIdentification1": "SVXU60ARI",
      "inHouseIdentification2": "PI"
    }
  },
  "pnrdates": [{
    "timeMode": "700",
    "dateTime": {
      "year": "2021",
      "month": 8,
      "day": 9
    }
  }, {
    "timeMode": "701",
    "dateTime": {
      "year": "2021",
      "month": 8,
      "day": 9,
      "hour": "00",
      "minutes": "00"
    }
  }, {
    "timeMode": "711",
    "dateTime": {
      "year": "2021",
      "month": 8,
      "day": 9,

```

```

        "hour": "12",
        "minutes": "12"
      }
    ]
  }
},
"errors": [],
"success": true
}

```

Examples of failed requests and responses

Response

```

{
  "data": {
    "queue_ListReply": {
      "errorReturn": {
        "errorDefinition": {
          "errorDetails": {
            "errorCode": "79D"
          }
        }
      }
    }
  },
  "errors": [],
  "success": true
}

```

The description of Amadeus error codes is on the documentation for Queue_List service.

Service for retrieving the count of records

API service	queue/counttotal
Short description	The service allows retrieving the count of records.
Supported HTTP-(S) methods	POST
Used methods of related systems	Amadeus: Queue_CountTotal

POST queue/counttotal method

Description retrieving the count of records

Request format /api/apiVersion/queue/counttotal

Request parameters are the same as request parameters of Amadeus service Queue_CountTotal

Response format

```
{
  "data": [{
    }],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
},
  "success": boolean
}
```

Response parameters are the same as response parameters of Amadeus service Queue_CountTotal, except the cases when the error appears in IBE and the errors array is consists of detailed information about errors.

Examples of successful requests and responses

Request

```
{
  "Queue_CountTotal": {
    "queueingOptions": {
      "selectionDetails": {
        "option": "QT"
      }
    }
  }
}
```

Response

```
{
  "data": {
    "queue_CountTotalReply": {
      "queueCountDisplay": {
        "localDateAndTime": {
          "dateTime": {
            "year": 21,
            "month": "8",
            "day": "18",
            "hour": 12,
            "minutes": 33
          }
        },
        "officeBeingDisplayed": {
          "sourceType": {

```

```

        "sourceQualifier1": "4"
    },
    "originatorDetails": {
        "inHouseIdentification1": "SVXU621AA"
    }
},
"standardQueueCountDisplay": [{
    "queueName": {
        "referenceDetails": {
            "value": "CONFO"
        }
    },
    "queueNumber": {
        "queueDetails": {
            "number": 1
        }
    }
},
"categoryAndCount": [{
    "categoryAndDateRange": {
        "subQueueInfoDetails": {
            "identificationType": "1",
            "itemNumber": "0"
        }
    },
    "queueCount": [{
        "queueDetails": {
            "numberOfItems": 1,
            "status": "TTL"
        }
    }, {
        "queueDetails": {
            "numberOfItems": 0,
            "status": "LQC"
        }
    }, {
        "queueDetails": {
            "numberOfItems": 1,
            "status": "ADD"
        }
    }, {
        "queueDetails": {
            "numberOfItems": 0,
            "status": "IW"
        }
    }
    ]
},
]
}, {
    "queueName": {
        "referenceDetails": {
            "value": "RATES"
        }
    },
    "queueNumber": {
        "queueDetails": {
            "number": 5
        }
    }
}

```

```

    }
  },
  "categoryAndCount": [{
    "categoryAndDateRange": {
      "subQueueInfoDetails": {
        "identificationType": "C",
        "itemNumber": "0"
      }
    },
    "queueCount": [{
      "queueDetails": {
        "numberOfItems": 4,
        "status": "TTL"
      }
    }, {
      "queueDetails": {
        "numberOfItems": 0,
        "status": "LQC"
      }
    }, {
      "queueDetails": {
        "numberOfItems": 7,
        "status": "ADD"
      }
    }, {
      "queueDetails": {
        "numberOfItems": 3,
        "status": "IW"
      }
    }
  ]
}
]
}
]
}
},
"errors": [],
"success": true
}

```

Examples of failed requests and responses

Response

```

{
  "data": {
    "queue_CountTotalReply": {
      "errorReturn": {
        "errorDefinition": {
          "errorDetails": {
            "errorCode": "91A"
          }
        }
      }
    }
  }
}

```

```

    },
    "errors": [],
    "success": true
}

```

The description of Amadeus error codes is on the documentation for Queue_CountTotal service.

Service for removing PNR from the queues

API service	queue/removeItem
Short description	The service allows removing the PNR from the queues.
Supported HTTP-(S) methods	POST
Used methods of related systems	Amadeus: Queue_RemoveItem

POST queue/removeItem method

Description removing the PNR from the queues

Request format /api/apiVersion/queue/removeItem

Request parameters are the same as request parameters of Amadeus service Queue_RemoveItem

Response format

```

{
  "data": [{
  }],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }
  ]
},
],
  "success": boolean
}

```

Response parameters are the same as response parameters of Amadeus service Queue_RemoveItem, except the cases when the error appears in IBE and the errors array is consists of detailed information about errors.

Examples of successful requests and responses

Request

```
{
  "Queue_RemoveItem": {
    "removalOption": {
      "selectionDetails": {
        "option": "QRP"
      }
    },
    "targetDetails": [
      {
        "targetOffice": {
          "sourceType": {
            "sourceQualifier1": "3"
          },
          "originatorDetails": {
            "inHouseIdentification1": "SVXU621AA"
          }
        },
        "queueNumber": {
          "queueDetails": {
            "number": "5"
          }
        },
        "categoryDetails": {
          "subQueueInfoDetails": {
            "identificationType": "c",
            "itemNumber": "0"
          }
        },
        "placementDate": {
          "timeMode": "1"
        },
        "recordLocator": [
          {
            "reservation": {
              "controlNumber": "PNRXXX"
            }
          }
        ]
      }
    ]
  }
}
```

Response

```
{
  "data": {
    "queue_RemoveItemReply": {
      "goodResponse": {
        "responseType": "G",
        "statusCode": "P"
      }
    }
  },
  "errors": [],
}
```

```
    "success": true
  }
```

Examples of failed requests and responses

Response

```
{
  "data": {
    "queue_RemoveItemReply": {
      "errorReturn": {
        "errorDefinition": {
          "errorDetails": {
            "errorCode": "913"
          }
        }
      }
    }
  },
  "errors": [],
  "success": true
}
```

The description of Amadeus error codes is on the documentation for `Queue_RemoveItem` service.

Service for void airtickets

API service	ticket
Short description	Allows to void tickets made by agents before 24:00 of office on the day when ticket was made.
Supported HTTP(S)-methods	DELETE
Conterminous systems methods used	Amadeus: Ticket_CancelDocument

DELETE ticket method

Description: deleting ticket.

Request format /api/apiVersion/ticket

Request headers

X-API-Key - unique GUID of system using IBE API.

Request parameters

ticketNumber - string, voiding ticket number.

Response sample

```
{
  "errors": [
    {
      "ibeCode": "Ibe",
      "description": "string",
      "messages": [
        {
          "code": "string",
          "text": "string",
          "additionalInfo": {
            "ssrType": "string",
```

```

        "info": "string"
    }
} ]
}
], "success": true
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN Attachment 1. IBE error codes
description	Code description, see. EN Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode
code	Amadeus error code
text	Amadeus error text
additionalInfo	Additional info
ssrType	SSR type
info	Additional info text

Response example

```

{
  "success": "true",
  "errors": []
}

```


Service for add remark to reservation /api/version/Reservation/AddRemark

API Service	/Reservation/AddRemark
Short description	Service for add remark via API IBE
Supported HTTP(S)-methods	POST

POST AddRemark method

Note: method is available on ADP

Description: adds remark to reservation

Request format: api/apiVersion/Reservation/AddRemark

Request header

X-Api-Key - unique GUID of system using IBE API.

Request body

```
{
  "reservationNumber": "string",
  "remarks": [
    {
      "passengerReference": "string",
      "segmentReference": "string",
      "freeText": "string"
    }
  ]
}
```

Request parameters description

Параметр	Описание
reservationNumber	unique booking number (PNR).
ssrs	SSR information object
passengerReference	Reference number
segmentReference	Flight segment reference
freeText	Text

Response parameters

```

{
  "data": {
    "journey": {
      "outboundFlights": [
        {
          "aircraft": "string",
          "aircraftIataCode": "string",
          "referenceNumber": "string",
          "origin": "string",
          "destination": "string",
          "departureDate": "2021-12-10T13:26:57.108Z",
          "arrivalDateUtc": "2021-12-10T13:26:57.108Z",
          "departureDateUtc": "2021-12-10T13:26:57.108Z",
          "arrivalDate": "2021-12-10T13:26:57.108Z",
          "flightNumber": "string",
          "classOfService": "string",
          "commercialFamily": "string",
          "operatingCarrier": "string",
          "marketingCarrier": "string",
          "codeshareName": "string",
          "flightDuration": "string",
          "stopDuration": "string",
          "techStops": [
            {
              "arrivalDateTime": "2021-12-10T13:26:57.108Z",
              "departureDateTime": "2021-12-10T13:26:57.108Z",
              "airportCode": "string",
              "stopDuration": "string",
              "flightDurationUntilNextStop": "string"
            }
          ],
          "flightDurationUntilNextStop": "string",
          "arrivalTerminal": "string",
          "departureTerminal": "string",
          "hash": "string",
          "statuses": [
            "string"
          ]
        }
      ],
      "returnFlights": [
        {
          "aircraft": "string",
          "aircraftIataCode": "string",
          "referenceNumber": "string",
          "origin": "string",
          "destination": "string",
          "departureDate": "2021-12-10T13:26:57.108Z",
          "arrivalDateUtc": "2021-12-10T13:26:57.108Z",
          "departureDateUtc": "2021-12-10T13:26:57.108Z",
          "arrivalDate": "2021-12-10T13:26:57.108Z",
          "flightNumber": "string",
          "classOfService": "string",
          "commercialFamily": "string",
          "operatingCarrier": "string",
          "marketingCarrier": "string",

```

```

    "codeshareName": "string",
    "flightDuration": "string",
    "stopDuration": "string",
    "techStops": [
      {
        "arrivalDateTime": "2021-12-10T13:26:57.108Z",
        "departureDateTime": "2021-12-10T13:26:57.108Z",
        "airportCode": "string",
        "stopDuration": "string",
        "flightDurationUntilNextStop": "string"
      }
    ],
    "flightDurationUntilNextStop": "string",
    "arrivalTerminal": "string",
    "departureTerminal": "string",
    "hash": "string",
    "statuses": [
      "string"
    ]
  }
],
"outboundFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": true,
  "isLowCost": true,
  "isUnaccompaniedMinor": true
},
"returnFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": true,
  "isLowCost": true,
  "isUnaccompaniedMinor": true
},
"outboundDuration": "string",
"returnDuration": "string",
"origin": "string",
"destination": "string",
"departureDate": "2021-12-10T13:26:57.108Z",
"returnDate": "2021-12-10T13:26:57.108Z",
"outboundBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult"
  }
],
"returnBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult"
  }
],
"containsInternationalFlights": true

```

```

},
"number": "string",
"reservationDate": "2021-12-10T13:26:57.108Z",
"bookingTimeLimit": "2021-12-10T13:26:57.108Z",
"passengers": [
  {
    "id": 0,
    "hasInfant": true,
    "referenceNumber": "string",
    "disability": {
      "isDeaf": true,
      "isBlind": true,
      "hasGuideDog": true,
      "assistanceType": "None",
      "wheelchairType": "Manual",
      "commentary": "string"
    },
    "firstName": "string",
    "middleName": "string",
    "surname": "string",
    "passengerType": "Adult",
    "birthDate": "2021-12-10T13:26:57.108Z",
    "sex": "Male",
    "nationality": "string",
    "documentNumber": "string",
    "documentExpiryDate": "2021-12-10T13:26:57.108Z",
    "loyaltyCard": "string",
    "email": "string",
    "phone": "string",
    "documentType": "Passport",
    "isAdult": true,
    "isChanged": true,
    "checkedIn": {
      "additionalProp1": true,
      "additionalProp2": true,
      "additionalProp3": true
    },
    "checkInHasStarted": {
      "additionalProp1": true,
      "additionalProp2": true,
      "additionalProp3": true
    },
    "hasPassengersDataErrors": true,
    "passengerDataErrors": [
      {
        "key": "string",
        "value": "string"
      }
    ]
  }
],
"services": [
  {
    "code": "string",
    "status": "string",
    "text": "string",

```

```

    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"seats": [
  {
    "number": 0,
    "title": "string",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"insurances": [
  {
    "number": "string",
    "passengerType": "Adult",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",

```

```

    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"partnerProducts": [
  {
    "partnerId": "Unknown",
    "partnerType": "Unknown",
    "bookingStatus": "Unknown",
    "code": "string",
    "amount": 0,
    "currency": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "isInfantProduct": true,
    "bookingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "orderId": "string",
    "preselectedOption": {
      "preselect": true,
      "passengers": [
        {
          "referenceNumber": "string",
          "passengerType": "Adult"
        }
      ]
    },
    "compatibleProductCodes": [
      "string"
    ],
    "internalError": "UnknownError",
    "internalErrorMessage": "string"
  }
],
"ticketFares": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": 0,
    "totalAmount": 0,
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ]
  }
]

```

```

    ],
    "passengerType": "Adult"
  }
],
"ticketFaresWithoutZapOff": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": 0,
    "totalAmount": 0,
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ],
    "passengerType": "Adult"
  }
],
"fareBasisCodesFlightRefs": [
  {
    "fareBasis": "string",
    "flightReferenceNumber": "string",
    "passengerReferenceNumber": "string"
  }
],
"hasSpecialConditions": true,
"isRefundAvailable": true,
"annulUnavailabilityReason": "FromKyrgyzstan",
"isAnnulAvailable": true,
"isAgentReservation": true,
"accountCode": "string",
"tourCode": "string"
},
"errors": [
  {
    "ibeCode": "Ibe",
    "description": "string",
    "messages": [
      {
        "code": "string",
        "text": "string",
        "additionalInfo": {
          "ssrType": "string",
          "info": "string"
        }
      }
    ]
  }
]
},
"success": true
}

```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	services	Array of objects with information about airline's available services
	code	Service code Possible values: see Attachment 2. Auxiliary services types
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	Reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	serviceOptions	Array of objects with information about which services should be automatically pre-selected in IBE frontend.
	category	Service category
	code	Service code Possible values: see Attachment 2. Auxiliary services types
	flightReferences	Array of objects with information about segments on which services are available
	preselect	If true, service should be automatically pre-selected in UI. <i>Note: Initial logic was applied to Web-client usage</i>
	priority	Priority of automatic service selection, 0 - highest. For instance, if service with priority 0 is unavailable, then service with priority 1 will be added etc.

	seatServices	Array of objects with information about flight's seats available for selection IMPORTANT: api/seatmap service returns more detailed information about available seats and should be used for "seat selection" service booking.
	code	Service code, SEAT
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	petsServiceConfiguration	Array of objects with information about "pet transportation" service availability on different segments. Needed for pet transportation validation on "business" family fares.
	petsList	Array of strings, containing list of pet types available for transportation
	flightReferences	Array of strings, containing list of segments on which service is available
	insuranceProducts	Array of objects with information about insurances available for selection
	code	Insurance code Possible values: see Attachment 2. Auxiliary services types
	amount	Insurance cost
	serviceCategories	Array of objects with information about services and order of its display in website
	code	Service code Attachment 2. Auxiliary services types
	category	Service category
	order	Display index in auxiliary services page. 1 - page top
	partnerProducts	Array of objects with information about partners services

	partnerId	Partners unique ID in IBE. Alfastrakh – 1, Sealine insurance – 2, Aeroexpress – 4
	partnerType	Partner type Insurance – 1, Aeroexpress – 2
	code	Partner's service code. Each partner has its own unique set of codes. See <u>EN Attachment 4. Partner's services.</u>
	amount	Partner's service cost.
	currency	currency
	bookingStatus	Partner's service booking status Possible values: 0 - unknown, 1 - service is not issued, 2 - service is issued, 3 - error issuing of service, 4 - refund performed
	portions	Service field
	type	Service field
	number	Service field
	flightReferenceNumbers	Service field
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode

	code	Amadeus error code
	text	Amadeus error text

Successful request and response examples

EN Attachment X. Request and response examples

Service for add SSR to reservation /api/version/Reservation/AddSSR

Сервис API	/Reservation/AddSSR
Short description	Service for add SSR via API IBE
Supported HTTP(S)-methods	POST

POST AddSSR method

Note: method is available on ADP

Description: adds SSR to reservation

Request format: api/apiVersion/Reservation/AddSSR

Request header

X-API-Key - unique GUID of system using IBE API.

Request body

```
{
  "reservationNumber": "string",
  "ssrs": [
    {
      "code": "string",
      "passengerReference": "string",
      "segmentReference": "string",
      "freeText": "string"
    }
  ]
}
```

Request parameters description

Параметр	Описание
reservationNumber	unique booking number (PNR).
ssrs	SSR information object
code	SSR code(different of selling in IBE): OTHS, WEAP, BIKE, SPEQ, DOCO, DOCA, MEDA, DEAF, BLND, DPNA, ESAN, STCR, WCBD, WCBW, WCHC, WCHR, WCHS, WCMP, WCOB, CTCM, CTCR,
passengerReference	Reference number
segmentReference	Flight segment reference

Параметр	Описание
freeText	Text

Response parameters

```

{
  "data": {
    "journey": {
      "outboundFlights": [
        {
          "aircraft": "string",
          "aircraftIataCode": "string",
          "referenceNumber": "string",
          "origin": "string",
          "destination": "string",
          "departureDate": "2021-12-10T13:26:57.108Z",
          "arrivalDateUtc": "2021-12-10T13:26:57.108Z",
          "departureDateUtc": "2021-12-10T13:26:57.108Z",
          "arrivalDate": "2021-12-10T13:26:57.108Z",
          "flightNumber": "string",
          "classOfService": "string",
          "commercialFamily": "string",
          "operatingCarrier": "string",
          "marketingCarrier": "string",
          "codeshareName": "string",
          "flightDuration": "string",
          "stopDuration": "string",
          "techStops": [
            {
              "arrivalDateTime": "2021-12-10T13:26:57.108Z",
              "departureDateTime": "2021-12-10T13:26:57.108Z",
              "airportCode": "string",
              "stopDuration": "string",
              "flightDurationUntilNextStop": "string"
            }
          ],
          "flightDurationUntilNextStop": "string",
          "arrivalTerminal": "string",
          "departureTerminal": "string",
          "hash": "string",
          "statuses": [
            "string"
          ]
        }
      ],
      "returnFlights": [
        {
          "aircraft": "string",
          "aircraftIataCode": "string",
          "referenceNumber": "string",
          "origin": "string",
          "destination": "string",
          "departureDate": "2021-12-10T13:26:57.108Z",
          "arrivalDateUtc": "2021-12-10T13:26:57.108Z",
          "departureDateUtc": "2021-12-10T13:26:57.108Z",
          "arrivalDate": "2021-12-10T13:26:57.108Z",
          "flightNumber": "string",
          "classOfService": "string",
          "commercialFamily": "string",
          "operatingCarrier": "string",
          "marketingCarrier": "string",

```

```

    "codeshareName": "string",
    "flightDuration": "string",
    "stopDuration": "string",
    "techStops": [
      {
        "arrivalDateTime": "2021-12-10T13:26:57.108Z",
        "departureDateTime": "2021-12-10T13:26:57.108Z",
        "airportCode": "string",
        "stopDuration": "string",
        "flightDurationUntilNextStop": "string"
      }
    ],
    "flightDurationUntilNextStop": "string",
    "arrivalTerminal": "string",
    "departureTerminal": "string",
    "hash": "string",
    "statuses": [
      "string"
    ]
  }
],
"outboundFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": true,
  "isLowCost": true,
  "isUnaccompaniedMinor": true
},
"returnFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": true,
  "isLowCost": true,
  "isUnaccompaniedMinor": true
},
"outboundDuration": "string",
"returnDuration": "string",
"origin": "string",
"destination": "string",
"departureDate": "2021-12-10T13:26:57.108Z",
"returnDate": "2021-12-10T13:26:57.108Z",
"outboundBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult"
  }
],
"returnBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult"
  }
],
"containsInternationalFlights": true

```

```

},
"number": "string",
"reservationDate": "2021-12-10T13:26:57.108Z",
"bookingTimeLimit": "2021-12-10T13:26:57.108Z",
"passengers": [
  {
    "id": 0,
    "hasInfant": true,
    "referenceNumber": "string",
    "disability": {
      "isDeaf": true,
      "isBlind": true,
      "hasGuideDog": true,
      "assistanceType": "None",
      "wheelchairType": "Manual",
      "commentary": "string"
    },
    "firstName": "string",
    "middleName": "string",
    "surname": "string",
    "passengerType": "Adult",
    "birthDate": "2021-12-10T13:26:57.108Z",
    "sex": "Male",
    "nationality": "string",
    "documentNumber": "string",
    "documentExpiryDate": "2021-12-10T13:26:57.108Z",
    "loyaltyCard": "string",
    "email": "string",
    "phone": "string",
    "documentType": "Passport",
    "isAdult": true,
    "isChanged": true,
    "checkedIn": {
      "additionalProp1": true,
      "additionalProp2": true,
      "additionalProp3": true
    },
    "checkInHasStarted": {
      "additionalProp1": true,
      "additionalProp2": true,
      "additionalProp3": true
    },
    "hasPassengersDataErrors": true,
    "passengerDataErrors": [
      {
        "key": "string",
        "value": "string"
      }
    ]
  }
],
"services": [
  {
    "code": "string",
    "status": "string",
    "text": "string",

```



```

    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"seats": [
  {
    "number": 0,
    "title": "string",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"insurances": [
  {
    "number": "string",
    "passengerType": "Adult",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",

```

```

    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"partnerProducts": [
  {
    "partnerId": "Unknown",
    "partnerType": "Unknown",
    "bookingStatus": "Unknown",
    "code": "string",
    "amount": 0,
    "currency": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "isInfantProduct": true,
    "bookingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "orderId": "string",
    "preselectedOption": {
      "preselect": true,
      "passengers": [
        {
          "referenceNumber": "string",
          "passengerType": "Adult"
        }
      ]
    },
    "compatibleProductCodes": [
      "string"
    ],
    "internalError": "UnknownError",
    "internalErrorMessage": "string"
  }
],
"ticketFares": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": 0,
    "totalAmount": 0,
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ]
  }
]

```

```

    ],
    "passengerType": "Adult"
  }
],
"ticketFaresWithoutZapOff": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": 0,
    "totalAmount": 0,
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ],
    "passengerType": "Adult"
  }
],
"fareBasisCodesFlightRefs": [
  {
    "fareBasis": "string",
    "flightReferenceNumber": "string",
    "passengerReferenceNumber": "string"
  }
],
"hasSpecialConditions": true,
"isRefundAvailable": true,
"annulUnavailabilityReason": "FromKyrgyzstan",
"isAnnulAvailable": true,
"isAgentReservation": true,
"accountCode": "string",
"tourCode": "string"
},
"errors": [
  {
    "ibeCode": "Ibe",
    "description": "string",
    "messages": [
      {
        "code": "string",
        "text": "string",
        "additionalInfo": {
          "ssrType": "string",
          "info": "string"
        }
      }
    ]
  }
]
},
"success": true
}

```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	services	Array of objects with information about airline's available services
	code	Service code Possible values: see Attachment 2. Auxiliary services types
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	Reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	serviceOptions	Array of objects with information about which services should be automatically pre-selected in IBE frontend.
	category	Service category
	code	Service code Possible values: see Attachment 2. Auxiliary services types
	flightReferences	Array of objects with information about segments on which services are available
	preselect	If true, service should be automatically pre-selected in UI. <i>Note: Initial logic was applied to Web-client usage</i>
	priority	Priority of automatic service selection, 0 - highest. For instance, if service with priority 0 is unavailable, then service with priority 1 will be added etc.

	seatServices	Array of objects with information about flight's seats available for selection IMPORTANT: api/seatmap service returns more detailed information about available seats and should be used for "seat selection" service booking.
	code	Service code, SEAT
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	petsServiceConfiguration	Array of objects with information about "pet transportation" service availability on different segments. Needed for pet transportation validation on "business" family fares.
	petsList	Array of strings, containing list of pet types available for transportation
	flightReferences	Array of strings, containing list of segments on which service is available
	insuranceProducts	Array of objects with information about insurances available for selection
	code	Insurance code Possible values: see Attachment 2. Auxiliary services types
	amount	Insurance cost
	serviceCategories	Array of objects with information about services and order of its display in website
	code	Service code Attachment 2. Auxiliary services types
	category	Service category
	order	Display index in auxiliary services page. 1 - page top
	partnerProducts	Array of objects with information about partners services

	partnerId	Partners unique ID in IBE. Alfastrakh – 1, Sealine insurance – 2, Aeroexpress – 4
	partnerType	Partner type Insurance – 1, Aeroexpress – 2
	code	Partner's service code. Each partner has its own unique set of codes. See <u>EN Attachment 4. Partner's services.</u>
	amount	Partner's service cost.
	currency	currency
	bookingStatus	Partner's service booking status Possible values: 0 - unknown, 1 - service is not issued, 2 - service is issued, 3 - error issuing of service, 4 - refund performed
	portions	Service field
	type	Service field
	number	Service field
	flightReferenceNumbers	Service field
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode

	code	Amadeus error code
	text	Amadeus error text

Successful request and response examples

EN Attachment X. Request and response examples

Service for prolong reservation

API service	reservation/prolong
Short description	Allows to add MIS segment into reservation for reservation will not archived
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Amadeus: PNR_AddMultiElements

POST reservation/prolong metod

Description: Allows to add MIS segment into reservation for reservation will not archived. Method accept array of PNRs, MIS segment adds to each of elements. In MIS segment adds current date plus year minus 4 days; also adds SVX origin airport

Request format /api/v2.3/Reservation/Prolong

Request parameters

Parameter	Description	
pnrNumbers	Array of PNR	
pnrNumbersPassengerCount	Array of PNR Is used when there are more than 2 passengers in the reservation	
	key	Reservation number (PNR)
	value	Number of passengers

Response format

```
{  
  "data": {  
    "result": [  
      {
```



```

        "pnrNumber": "string",
        "success": true,
        "message": "string"
    }
]
},
"errors": [
    {
        "ibeCode": "Ibe",
        "description": "string",
        "messages": [
            {
                "code": "string",
                "text": "string",
                "additionalInfo": {
                    "ssrType": "string",
                    "info": "string"
                }
            }
        ]
    }
],
"success": true
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Prolongation info
result	PNR array
pnrNumber	Prolonged PNR
success	Sign of query execution success.
message	Message on case of problem
errors	Information about errors
ibeCode	Error code
description	Error description
messages	Objects array with Amadeus errors. Can be empty.
code	Amadeus error code
text	Amadeus error text

Success response example

```
{
  "data": {
    "result": [
      {
        "pnrNumber": "L9WMRI",
        "success": true
      }
    ]
  },
  "errors": [],
  "success": true
}
```


Service for retrieving SSR and remarks information /api/version/Reservation/Advanced

Сервис API	/Reservation/Advanced
Short description	Позволяет получить все SSR и ремарки (с типом RM), которые есть в бронировании, чтобы просмотреть их, а также, чтобы брони с нужными ремарками поставить в очередь, а брони с неинтересными для агента ремарками - удалить из очереди
Supported HTTP(S)-methods	GET

POST AddSSR method

Note: method is available on ADP

Description: adds SSR to reservation

Request format: api/apiVersion/Reservation/Advanced

Request header

X-API-Key - unique GUID of system using IBE API.

Request parameters description

Параметр	Описание
PnrNumber	Reservation number.
TicketNumber	Ticket number, if one is used instead of PNR number, to retrieve reservation.
LastName	Passenger last name.
LastNameHash	Passenger last name hash.

Response parameters

```

{
  "data": {
    "journey": {
      "outboundFlights": [
        {
          "aircraft": "string",
          "aircraftIataCode": "string",
          "referenceNumber": "string",
          "origin": "string",
          "destination": "string",
          "departureDate": "2021-12-10T13:26:57.108Z",
          "arrivalDateUtc": "2021-12-10T13:26:57.108Z",
          "departureDateUtc": "2021-12-10T13:26:57.108Z",
          "arrivalDate": "2021-12-10T13:26:57.108Z",
          "flightNumber": "string",
          "classOfService": "string",
          "commercialFamily": "string",
          "operatingCarrier": "string",
          "marketingCarrier": "string",
          "codeshareName": "string",
          "flightDuration": "string",
          "stopDuration": "string",
          "techStops": [
            {
              "arrivalDateTime": "2021-12-10T13:26:57.108Z",
              "departureDateTime": "2021-12-10T13:26:57.108Z",
              "airportCode": "string",
              "stopDuration": "string",
              "flightDurationUntilNextStop": "string"
            }
          ],
          "flightDurationUntilNextStop": "string",
          "arrivalTerminal": "string",
          "departureTerminal": "string",
          "hash": "string",
          "statuses": [
            "string"
          ]
        }
      ],
      "returnFlights": [
        {
          "aircraft": "string",
          "aircraftIataCode": "string",
          "referenceNumber": "string",
          "origin": "string",
          "destination": "string",
          "departureDate": "2021-12-10T13:26:57.108Z",
          "arrivalDateUtc": "2021-12-10T13:26:57.108Z",
          "departureDateUtc": "2021-12-10T13:26:57.108Z",
          "arrivalDate": "2021-12-10T13:26:57.108Z",
          "flightNumber": "string",
          "classOfService": "string",
          "commercialFamily": "string",
          "operatingCarrier": "string",
          "marketingCarrier": "string",

```

```

    "codeshareName": "string",
    "flightDuration": "string",
    "stopDuration": "string",
    "techStops": [
      {
        "arrivalDateTime": "2021-12-10T13:26:57.108Z",
        "departureDateTime": "2021-12-10T13:26:57.108Z",
        "airportCode": "string",
        "stopDuration": "string",
        "flightDurationUntilNextStop": "string"
      }
    ],
    "flightDurationUntilNextStop": "string",
    "arrivalTerminal": "string",
    "departureTerminal": "string",
    "hash": "string",
    "statuses": [
      "string"
    ]
  }
],
"outboundFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": true,
  "isLowCost": true,
  "isUnaccompaniedMinor": true
},
"returnFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": true,
  "isLowCost": true,
  "isUnaccompaniedMinor": true
},
"outboundDuration": "string",
"returnDuration": "string",
"origin": "string",
"destination": "string",
"departureDate": "2021-12-10T13:26:57.108Z",
"returnDate": "2021-12-10T13:26:57.108Z",
"outboundBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult"
  }
],
"returnBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult"
  }
],
"containsInternationalFlights": true

```

```

},
"number": "string",
"reservationDate": "2021-12-10T13:26:57.108Z",
"bookingTimeLimit": "2021-12-10T13:26:57.108Z",
"passengers": [
  {
    "id": 0,
    "hasInfant": true,
    "referenceNumber": "string",
    "disability": {
      "isDeaf": true,
      "isBlind": true,
      "hasGuideDog": true,
      "assistanceType": "None",
      "wheelchairType": "Manual",
      "commentary": "string"
    },
    "firstName": "string",
    "middleName": "string",
    "surname": "string",
    "passengerType": "Adult",
    "birthDate": "2021-12-10T13:26:57.108Z",
    "sex": "Male",
    "nationality": "string",
    "documentNumber": "string",
    "documentExpiryDate": "2021-12-10T13:26:57.108Z",
    "loyaltyCard": "string",
    "email": "string",
    "phone": "string",
    "documentType": "Passport",
    "isAdult": true,
    "isChanged": true,
    "checkedIn": {
      "additionalProp1": true,
      "additionalProp2": true,
      "additionalProp3": true
    },
    "checkInHasStarted": {
      "additionalProp1": true,
      "additionalProp2": true,
      "additionalProp3": true
    },
    "hasPassengersDataErrors": true,
    "passengerDataErrors": [
      {
        "key": "string",
        "value": "string"
      }
    ]
  }
],
"services": [
  {
    "code": "string",
    "status": "string",
    "text": "string",

```

```

    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"seats": [
  {
    "number": 0,
    "title": "string",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"insurances": [
  {
    "number": "string",
    "passengerType": "Adult",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",

```



```

    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2021-12-10T13:26:57.108Z"
  }
],
"partnerProducts": [
  {
    "partnerId": "Unknown",
    "partnerType": "Unknown",
    "bookingStatus": "Unknown",
    "code": "string",
    "amount": 0,
    "currency": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "isInfantProduct": true,
    "bookingId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "orderId": "string",
    "preselectedOption": {
      "preselect": true,
      "passengers": [
        {
          "referenceNumber": "string",
          "passengerType": "Adult"
        }
      ]
    },
    "compatibleProductCodes": [
      "string"
    ],
    "internalError": "UnknownError",
    "internalErrorMessage": "string"
  }
],
"ticketFares": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": 0,
    "totalAmount": 0,
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ]
  }
]

```

```

    ],
    "passengerType": "Adult"
  }
],
"ticketFaresWithoutZapOff": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": 0,
    "totalAmount": 0,
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ],
    "passengerType": "Adult"
  }
],
"fareBasisCodesFlightRefs": [
  {
    "fareBasis": "string",
    "flightReferenceNumber": "string",
    "passengerReferenceNumber": "string"
  }
],
"hasSpecialConditions": true,
"isRefundAvailable": true,
"annulUnavailabilityReason": "FromKyrgyzstan",
"isAnnulAvailable": true,
"isAgentReservation": true,
"accountCode": "string",
"tourCode": "string"
},
"errors": [
  {
    "ibeCode": "Ibe",
    "description": "string",
    "messages": [
      {
        "code": "string",
        "text": "string",
        "additionalInfo": {
          "ssrType": "string",
          "info": "string"
        }
      }
    ]
  }
]
},
"success": true
}

```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	services	Array of objects with information about airline's available services
	code	Service code Possible values: see <u>Attachment 2. Auxiliary services types</u>
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	Reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	serviceOptions	Array of objects with information about which services should be automatically pre-selected in IBE frontend.
	category	Service category
	code	Service code Possible values: see <u>Attachment 2. Auxiliary services types</u>
	flightReferences	Array of objects with information about segments on which services are available
	preselect	If true, service should be automatically pre-selected in UI. <i>Note: Initial logic was applied to Web-client usage</i>
	priority	Priority of automatic service selection, 0 - highest. For instance, if service with priority 0 is unavailable, then service with priority 1 will be added etc.

	seatServices	Array of objects with information about flight's seats available for selection IMPORTANT: api/seatmap service returns more detailed information about available seats and should be used for "seat selection" service booking.
	code	Service code, SEAT
	flightReferences	Array of objects with information about segments and passengers to whom services are available as well as service's prices
	referenceNumber	reference to flight segment
	passengerPrices	Array of objects with information about passenger and service price
	referenceNumber	Reference to passenger
	amount	Price of service
	petsServiceConfiguration	Array of objects with information about "pet transportation" service availability on different segments. Needed for pet transportation validation on "business" family fares.
	petsList	Array of strings, containing list of pet types available for transportation
	flightReferences	Array of strings, containing list of segments on which service is available
	insuranceProducts	Array of objects with information about insurances available for selection
	code	Insurance code Possible values: see Attachment 2. Auxiliary services types
	amount	Insurance cost
	serviceCategories	Array of objects with information about services and order of its display in website
	code	Service code Attachment 2. Auxiliary services types
	category	Service category
	order	Display index in auxiliary services page. 1 - page top
	partnerProducts	Array of objects with information about partners services

	partnerId	Partners unique ID in IBE. Alfastrakh – 1, Sealine insurance – 2, Aeroexpress – 4
	partnerType	Partner type Insurance – 1, Aeroexpress – 2
	code	Partner's service code. Each partner has its own unique set of codes. See <u>EN Attachment 4. Partner's services.</u>
	amount	Partner's service cost.
	currency	currency
	bookingStatus	Partner's service booking status Possible values: 0 - unknown, 1 - service is not issued, 2 - service is issued, 3 - error issuing of service, 4 - refund performed
	portions	Service field
	type	Service field
	number	Service field
	flightReferenceNumbers	Service field
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode

	code	Amadeus error code
	text	Amadeus error text

Successful request and response examples

EN Attachment X. Request and response examples

Service for removing segment from the reservation

API service	reservation/CancelSegments
Short description	Allows to remove segment from reservation
Поддерживаемые HTTP(S)-методы	POST

GET api/reservation/CancelSegments method

Description: Can remove chosen segment from the reservation

Request format /api/apiVersion/Reservation/CancelSegments

Request shema

```
{
  "segmentsReferences": [
    "string"
  ]
}
```

Response format

```
{
  "errors": [
    {
      "ibeCode": "Ibe",
      "description": "string",
      "messages": [
        {
          "code": "string",
          "text": "string",
          "additionalInfo": {
            "ssrType": "string",
            "info": "string"
          }
        }
      ]
    }
  ],
  "success": true
}
```

Response parameters

Параметр	Описание
----------	----------

success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text
additionalInfo	Additional error
ssrType	SSR type
info	Description

Response example

```
{
  "errors": [],
  "success": true
}
```


Obtain tickets for rebook /api/version/Exchange/Search

API service	/Exchange/Search
Short description	Obtain list of available tickets for rebook
Supported HTTP(S)-methods	POST
Conterminous systems methods used	SIG: SIG_AirReShopRQ

Method POST Exchange/Search

Description: method is only available on ADP

Note: allow to get information about the tickets available for rebook

Request format: api/apiVersion/Exchange/Search

Request headers

X-API-Key - unique GUID for the systems, using IBE API.

Request body

```
{
  "pnrNumber": string,
  "lastName": string,
  "originDestinations": [
    {
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "segmentReference": string
    }
  ],
  "passengerReferences": [
    string
  ]
}
```

Request parameters

Parameter	Description
reservationNumber	Number of a reservation
lastName	Surname of a passenger
originDestination	

	An array of data about the part of the light to be exchanged
origin	IATA-code of origin airport
destination	IATA-code of destination airport
departureDate	Departure date
segmentReference	Reference to the segment to be exchanged
passengerReference	Reference to the segment whose tickets to be exchanged

Response format

```
{
  "data": {
    "flights": [flightGroup],
    "fareFamilies": [
      {
        "fareFamilyId": integer,
        "code": string,
        "category": string,
        "isSubsidized": boolean,
        "options": [
          {
            "code": string,
            "category": string,
            "order": integer,
            "showInPopup": boolean,
            "showInSummary": boolean,
            "showInUpsell": boolean,
            "showInFareIncluded": boolean,
            "showForAdult": boolean,
            "showForInfant": boolean,
            "showInBookingCart": boolean
          }
        ],
        "order": integer,
        "commercialFamily": string,
        "canRefund": boolean,
        "hasUpsale": boolean,
        "isLowCost": boolean
      }
    ],
    "recommendations": [
      {
        "totalPrice": float,
        "totalAddCollectPrice": float,
        "breakdowns": [
          {
            "ticketNumber": string,
            "passengerReference": string,
            "paxType": integer,

```

```

        "baseFareAmount": float,
        "taxAmount": float,
        "totalAmount": float,
        "addCollectAmount": float
    }
],
"flightGroupDetails": [
    {
        "flightGroupId": integer,
        "fareFamilyId": integer,
        "seatsLeft": integer,
        "isExchange": boolean,
        "baggageAllowances": [
            {
                "quantity": integer,
                "weight": integer,
                "passengerType": integer
            }
        ],
        "fareDetails": [
            {
                "bookingCode": string,
                "fareBasis": string,
                "isBreakpoint": boolean,
                "paxType": integer,
                "paxCount": integer,
                "flightIndex": integer,
                "ruleKey": string
            }
        ],
        "flightGroupHash": string
    }
]
},
"airShopSessionId": string,
"currency": string,
"foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string,
    "commercialFamily": string,
    "translation": {
        "ru": string,
        "en": string,
        "zh": string
    }
}
],
"moneyWallCoefficient": float
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
}
]

```

```

    ]
  }
},
"success": true
}

```

flightGroup object

```

{
  "flightGroupId": integer,
  "flights": [{
    "flightId": integer,
    "departureAirport": string,
    "departureTerminal": string,
    "arrivalAirport": string,
    "arrivalTerminal": string,
    "departureDateTime": datetime,
    "excludedNationalities": [
      string
    ]
    "arrivalDateTime": datetime,
    "aircraft": string,
    "aircraftIataCode": string,
    "techStops": [{
      "arrivalDateTime": datetime,
      "departureDateTime": datetime,
      "airportCode": string,
      "stopDuration": string,
      "flightDurationUntilNextStop": string
    }
  ],
  "flightDuration": string,
  "flightDurationUntilNextStop": string,
  "stopDuration": string,
  "flightNumber": string,
  "marketingCarrier": string,
  "operatingCarrier": string,
  "codeshareName": string,
  "hash": string
}
],
"flightGroupDuration": string,
"hash": string
}

```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true

Parameter	Description
data	Returned data
airShopSessionId	SIG search session ID
currency	currency in which prices are represented
moneyWallCoefficient	Coefficient for calculating the monthly installment payment
flights	Array of flightGroup objects with information about flights
fareFamilies	Array of objects with information about fare families
fareFamilyId	Fare family ID
code	Fare family code, for ex: ECOLIGHTRU/ECOFLEXRU/ECOPREMURU etc.
category	Fare family category, for ex: ECOLIGHT/ECOFLEX/ECOPREM Possible values: see Attachment 2, Main fare family categories
order	Fare families display order
commercialFamily	f\Fare family type for ex: U6ECONOMY or U6BUSINESS
isSubsidized	Sign of subsidised tariff
canRefund	Sign of refundable tariff
hasUpsale	Sign indentifying whether present tariff can be upgraded to next one in price category. Used for management of Upsale block display in frontend.
options	Array of fareFamiliesOptions objects with information about tariff options. Possible values: see Attachment 2, tariff options
foodOnBoards	Array of objects with information about meals available on the flight
departureAirport	Airport of departure
arrivalAirport	Airport of arrival

Parameter		Description
	flightNumber	Flight number
	commercialFamily	Fare family type (different type of meals for economy and business can be available for one flight)
	translation	Object with translation options
	ru	Options translation to russian
	en	Options translation to english
	zh	Option translation chinese
recommendations		Array of recommendations objects for flights
	totalPrice	Total ticket price including all applied discounts
	totalAddCollectPrice	Amount to be paid in case of rebook
breakdowns		Array of tickets available for exchange
	ticketNumber	Number of issued ticket
	passengerReference	Reference to a passenger
	paxType	Type of a passenger
	baseFareAmount	Base price
	taxAmount	Tariffs and fees
	totalAmount	Full ticket price including tariffs and fees
	AddCollectAmount	Amount to be paid in case of rebook
errors		Information about errors appeared during query execution.
	ibeCode	Error code, see EN Attachment 1. IBE error codes
	description	Code description, see. EN Attachment 1. IBE error codes
messages		Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Amadeus error code
	text	Amadeus error text

flightGroup object

Contain information about flights

data.flights	
flightGroupId	Index number used in recommendations
flightGroupDuration	Total flight time: summed up of duration values of each <u>transfer</u> and technical stop.
hash	Service field
flights	Array of objects with information <u>about transfers</u>
flightId	Flight Id
departureAirport	Airport of departure
departureTerminal	Terminal in airport of departure
arrivalAirport	Airport of arrival
departureDateTime	Departure date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T17:25:00
arrivalDateTime	Arrival date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T20:50:00
aircraft	Full aircraft title For ex.: Airbus A320
aircraftIataCode	IATA aircraft code
flightDuration	Duration of flight
flightNumber	flight number
marketingCarrier	Marketing carrier Example: U6, S7
operatingCarrier	Operating carrier (one actually performed flight) Example: U6, OK
hash	service field

Service of tickets rebook /api/version/Exchange

API service	/Exchange
Short description	Service of tickets exchange
Supported HTTP(S)-methods	POST
Conterminous systems method used	Amadeus: Ticket_RepricePNRWithBookingClass, Ticket_ReissueConfirmedPricing

Метод POST Exchange

Note: method is available on ADP

Description: allow to make a rebook of ticket

Request format: api/apiVersion/Exchange

Request headers

X-Api-Key - unique GUID for systems using IBE API.

Request body

```
{
  "pnrNumber": string,
  "lastName": string,
  "travelParts": [
    {
      "segmentReference": string,
      "segmentReferenceLast": string,
      "fareFamily": {
        "code": string,
        "category": string,
        "isSubsidized": boolean,
        "isLowCost": boolean,
        "isUnaccompaniedMinor": boolean
      },
    },
    "addFlights": [
      {
        "origin": string,
        "destination": string,
        "departureDate": datetime,
        "arrivalDate": datetime,
        "flightNumber": string,
        "classOfService": string,
        "fares": [
          {
            "fareBasis": string,
```



```

        "isBreakpoint": boolean,
        "paxType": string
    }
    ]
}
],
"passengerReferences": [
    string
]
}

```

Request parameters

Parameter		Description
pnrNumber		Number of reservation
lastName		Passenger's surname
travelParts		Array with information about the flight segment
	fareFamily	Object with information about the fare family
	code	Tariff code
	category	Tariff category
	isSubsidized	Sign of subsidized tariff
	isLowCost	Sign of lowcost flight
	isUnaccompaniedMinor	Sign of unaccompanied child
addFlights		An array with information about the segment to be exchanged
	origin	Departure airport
	destination	Destination airport
	departureDate	Departure date
	arrivalDate	Arrival date
	flightNumber	Number of flight
	classOfService	Information about selected tariff
	fares	Tariff information object
	fareBasis	Base fare name

		isBreakpoint	Flag identifying whether tariff end on present segment
		paxType	Passenger type
passengerReference			Reference to the passenger whose ticket to be rebooked

Response format

```
{
  "data": {
    "tickets": [{
      "number": string,
      "passengerReference": string,
      "isInfantTicket": boolean,
      "issueDate": datetime
    }
  ],
  "fareFamilies": [
    {
      "fareFamilyId": integer,
      "code": string,
      "category": string,
      "isSubsidized": boolean,
      "options": [
        {
          "code": string,
          "category": string,
          "order": integer,
          "showInPopup": boolean,
          "showInSummary": boolean,
          "showInUpsell": boolean,
          "showInFareIncluded": boolean,
          "showForAdult": boolean,
          "showForInfant": boolean,
          "showInBookingCart": boolean
        }
      ],
      "order": integer,
      "commercialFamily": string,
      "canRefund": boolean,
      "hasUpsale": boolean,
      "isLowCost": boolean
    }
  ],
  "appliedDiscounts": [],
  "baseFarePrices": [
    {
      "currency": string,
      "passengerReference": string,
      "isInfantFare": boolean,
      "totalPrice": number,
      "basePrice": number,
      "priceHidden": boolean
    }
  ]
}
```

```

    }
  ],
  "spentBonusesByPassengers": [{
    "amount": integer,
    "passengerReference": string
  }
  ],
  "appliedCertificates": [{
    "amount": integer,
    "number": string
  }
  ],
  "fareBasisCodesFlightRefs": {[fareBasisCodesFlightRefs object]}],
  "familyTrip": boolean,
  "originOfficeId": string,
  "originOfficeType": integer,
  "isPayForbiddenForAgency": boolean,
  "foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string
    "commercialFamily": string,
    "ru": string,
    "en": string,
    "zh": string
  }
  ],
  "reservationRetrieveStatus": integer,
  "ticketCouponStatuses": [{
    "ticketNumber": string,
    "couponStatuses": {
      flightReference: boolean
    }
  }
  ],
  "engineType": integer,
  "isGroupReservation": boolean,
  "isCompanyGroupReservation": boolean,
  "isReceiptAvailable": boolean,
  "isInvoluntaryRefundAvailable": boolean,
  "isVoluntaryRefundAvailable": boolean,
  "journey": {
    "outboundFlights": [fullReservationFlight],
    "outboundFareFamily": {
      "code": string,
      "category": string
    },
    "outboundDuration": string,
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "outboundBaggageAllowances": [{
      "quantity": integer,
      "weight": integer,
      "passengerType": integer
    }
  ],
  "returnBaggageAllowances": [{

```

```

        "quantity": integer,
        "weight": integer,
        "passengerType": integer
    }
    ],
    "containsInternationalFlights": boolean
},
"number": string,
"reservationDate": datetime,
"passengers": [[passengerObject]],
"services": [],
"seats": [],
"luggageSeats": [],
"insurances": {},
"partnerProducts": [],
"hasSpecialConditions": boolean,
"isRefundAvailable": boolean,
"isAnnulAvailable": boolean,
"isAgentReservation": boolean
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ],
}
],
"success": boolean
}

```

fullReservationFlight object

```

{
    "aircraft": string,
    "aircraftIataCode": string,
    "referenceNumber": string,
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "arrivalDate": datetime,
    "arrivalDateUtc": datetime,
    "departureDateUtc": datetime,
    "flightNumber": string,
    "classOfService": string,
    "marketingCarrier": string,
    "operatingCarrier": string,
    "flightDuration": string,
    "stopDuration": string,
    "techStops": [{
        "arrivalDateTime": datetime,
        "departureDateTime": datetime,
        "airportCode": string,
        "stopDuration": string,
    }
    ],
}

```

```

        "flightDurationUntilNextStop": string
    }
],
"flightDurationUntilNextStop": string,
"arrivalTerminal": string,
"departureTerminal": string,
"hash": string,
"isCancelledByAirline": boolean
}

```

Response parameters

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true.
data	The data returned by the request.
number	Unique booking number (PNR).
reservationDate	Date and time of booking creation Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T15:44:00
passengers	An array of the passengerData objects with passenger information
services	Array of objects with information about auxiliary services.
seats	Array of objects with information about selected seats.
insurances	Object with information about selected insurance.
luggageSeats	An array of objects with information about luggage with separate air ticket and ssr CBBG.
referenceNumber	The reference to CBBG luggage.
passengerReference	The reference to passenger who added CBBG luggage. Associated with field data → passengers[n] → referenceNumber.

	partnerProducts	An array of objects with information about partner services.
	journey	Flight information for current booking
	outboundFlights	An array of the fullReservationFlight objects with details of outbound flights
	outboundFareFamily	Service field, contains the fareFamily code
	returnFlights	An array of fullReservationFlight objects with detailed information about the "return" flights
	origin	Initial airport of departure
	destination	Initial airport of destination
	departureDate	Departure date
	returnDate	Return date
	ticketFares	An array of objects with information about tariffs after applying the discount
	baseAmount	Base rate
	passengerReferences	An array of objects with links to passengers
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "PA"
	passengerType	Passenger type Possible values: see Attachment 2. Passenger types .
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "TST"
	totalAmount	The total amount of the tariff
	currency	Reservation currency
	ticketFaresWithoutZapOff	An array of objects with information about tariffs without applying a discount
	baseAmount	Base rate
	passengerReferences	An array of objects with links to passengers
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "PA"

	passengerType	Passenger type Possible values: see Attachment 2. Passenger types
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "TST"
	totalAmount	The total amount of the tariff
	currency	Reservation currency
	hasSpecialConditions	Sign of the additional conditions
	isRefundAvailable	Sign of the availability of the return
	annulUnavailabilityReason	Reason for unavailability of booking cancellation
	isAnnulAvailable	Indication of availability of booking cancellation
errors		Information about errors that occurred during the operation of the service
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from Amadeus' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
	code	Amadeus error code
	text	Amadeus error text

fullReservationFlight object

Parameter	Description
aircraftIataCode	IATA aircraft code
referenceNumber	Reference to the current flight segment
origin	Departure airport for the current segment
destination	Destination airport for the current segment

departureDate	Departure date and time for the current segment
arrivalDate	Arrival date and time for the current segment
arrivalDateUtc	Date and time of arrival for the current segment in UTC
departureDateUtc	Departure date and time for the current segment in UTC
flightNumber	Flight number
classOfService	Information about the selected tariff
marketingCarrier	Marketing airline For example: U6, S7
operatingCarrier	The airline actually performing the flight For example: U6, OK
techStops	An array with information about technical landings
arrivalDateTime	Date and time of arrival at the pick-up point
departureDateTime	Date and time of departure from the pick-up point
airportCode	Airport code of the stop
stopDuration	Duration of the stop
flightDurationUntilNextStop	Duration of the flight to the next stop
flightDurationUntilNextStop	Duration of the flight to the next stop
arrivalTerminal	Terminal number at the point of arrival
departureTerminal	Terminal number at the point of departure
hash	Unique hash of the flight
isCancelledByAirline	Sign of cancellation of flights by the airline

passengerData object

Parameter	Description
id	Passenger's serial number (numbered from 1)
hasInfant	An sign that this passenger has a baby
referenceNumber	Reference to this passenger in the booking

firstName	Passenger's name	
middleName	Passenger's patronymic (may be absent)	
surname	Passenger's surname	
passengerType	Passenger type Possible values: see Attachment 2. Passenger types	
isAdult	Whether the passenger is an adult true for ADT/SRC/YTH type passengers false for CHD/INF type passengers	
birthDate	Date of birth	
sex	Passenger gender Possible values: see dictionary "gender of passengers"	
nationality	Nationality (country) Possible values: dictionary of countries	
documentNumber	Document number	
documentType	Document type Possible values: dictionary of document types	
documentExpiryDate	Date until which the document is valid Mandatory only for a foreign passport. For other documents, +10 years from the current date is automatically set	
email	Email	
phone	Phone	
checkedIn	Passenger check-in on flights	
checkInHasStarted	Starting check-in on flights	
hasPassengersDataErrors	The sign of errors in passengers data.	
passengerDataErrors	Array with detailed information of errors in passengers data.	
	key	The key.
	value	The value.

Service API methods description

Prolong Amadeus session service

API service	renew (service)
Short description	Allows to prolong Amadeus session for preliminary booking.
Supported HTTP(S)-methods	POST

POST renew method

Description: prolongs Stateful Amadeus session for preliminary booking. Method should be used if user needs more time to fill in contact data.

There's no need to call this method if booking is created with all necessary data at once: flights, passengers and tariffs.

Request format /api/apiVersion/renew

Request parameters

No parameters. Necessary information about Amadeus session is present in booking context.

Request format

```
{
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
},
]
"success": boolean
}
```

Request parameters

Parameter	Description
-----------	-------------

success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Successful and failed request and response example

Unable to find session parameters in context

```
{
  "errors": [
    {
      "ibeCode": 1001,
      "description": "Operation can not be performed due to invalid
reservation state.",
      "messages": []
    }
  ],
  "success": false
}
```

Retrieve airports list service

API service	airports (service)
Short description	Allows to retrieve list of airports to which U6 company performs flights
Supported HTTP(S)-methods	GET

GET airports methods

Description: returns list of airports and its iata codes.

Request format /api/apiVersion/airports

Request parameters

No parameters.

Response format

```
{
  "data": [{
    "country": string,
    "cityRu": string,
    "cityEn": string,
    "iataCity": string,
    "airportRu": string,
    "airportEn": string,
    "iataAirport": string
  }
],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
},
]
"success": boolean
}
```

Response parameters

Parameter	Description
-----------	-------------

success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
country	Country where city is located
cityRu	City name (russian locale)
cityEn	City name (english locale)
iataCity	IATA-code of the city
airportRu	Airport name (russian locale)
airportEn	Airport name (english locale)
iataAirport	IATA-code of the airport
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

IBE health check service

API service	health (service)
Short description	Allows to retrieve status of work of IBE services
Supported HTTP(S)-methods	GET

GET health method

Description: returns information about services and percent values of warnings/errors which appeared for them for specified period of time. Also contains information about amount of created and payed bookings for the same period of time.

At present period is set to 10 minutes.

Request format /api/health

Pay attention to that it is not needed to specify API version for calling this service.

Request parameters

No parameters.

Response format

```
{
  "data": {
    "endpointsHealth": [{
      "endpoint": string,
      "severity": integer,
      "sessionsCount": integer,
      "errorsPercentAvg": integer,
      "warningsPercentAvg": integer,
      "state": integer
    }
  ],
  "bookingStats": {
    "retrievedBookingsCount": integer,
    "createdBookingsCount": integer
  }
},
"errors": [],
"success": boolean
}
```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	endpointsHealth	Array of objects with information about services
	endpoint	Service name List of services: Attachment 2. Dictionaries
	severity	Service importance 1 – low, 2 – medium, 3 – high Services importance: Attachment 2. Dictionaries Services of high importance is responsible for booking creation.
	sessionsCount	Number of requests of this service for the period
	errorsPercentAvg	Average percent of errors for the service
	warningsPercentAvg	Average percent of warnings for the service

	state	<p>Service state</p> <p>Possible service states:</p> <p>1 – OK, normal service state.</p> <p>2 – Warning, there are non-critical problems in a service.</p> <p>State 2 is set for a service in following cases:</p> <ol style="list-style-type: none"> 1. Warnings percentage exceeds 25% 2. Number of sessions is less than 10 and error percentage is higher then 25% 3. Number of sessions is greater than 10 and error percentage is in 25%-75% interval <p>3 – Error, service work is broken</p> <p>State 3 is set for a service in case when there are more than 10 session and error percentage is above 75%</p>
	bookingStats	Object with booking information
	retrievedBookingsCount	Number of payed bookings for the period
	createdBookingsCount	Number of created bookings for the period
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Amadeus error code
	text	Amadeus error text

Detailed service work description

1. api/health service queries information via RES API for Azure Application Insights. 10 seconds timeout is set for the service. After this time IBE either returns expected

information about services state or ibeCode = 1009 error. After receiving of such error it is needed to request information from service again.

2. api/health returns information about errors/warnings in percents due to following reason: there might appear situation when error occurs for one client (in one booking session) and he reproduces it multiple times. For instance, clicks on "proceed" button though received refresh booking error already.

This might negatively impact on overall statistics as other clients might not experience such type of errors at all. Therefore percent of errors for each service is returned instead of error's count. At first errors for each service gets grouped by user's session and then average error percent is calculated.

1. Information about service work is queried for certain period of time. At present this period is set to 10 minutes. For instance if service was queried at 11:17:10 then information will be requested since 11:07:10.
2. Due to information passed in Application insights not instantly, it is recommended to query api/health service approximately once in 2-3 minutes. In case of appeared problem more frequent querying of service is possible.
3. If api/health returns errors with 5xx code then it means that IBE is completely inoperative.

Send error report service

API service	problemNotification (service)
Short description	Allows users to send error reports
Supported HTTP(S)-methods	POST

POST problemNotification method

Description: allows users to report about problems. Report sending form is available in frontend when "general technical error" gets received with 1000 code.

Request format /api/apiVersion/problemNotification

Request body

```
{
  "from": "string",
  "description": "string",
  "details": "string"
}
```

Request parameters

Parameter	Description
from	Sender. Any string identifying user.
description	Description of the problem.
details	Technical details of the problem. For example, session id.

Response format

```
{
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Request and response examples

Successful sending of information about error

```
{  
  "errors": [],  
  "success": true  
}
```

Get exchange rates service

API service	exchangerates (service)
Short description	Allows to receive information about currency exchange rates
Supported HTTP(S) methods	GET

GET exchangerates method

Description: retrieves information about currency exchange rates

Request format /api/apiVersion/exchangerates

Request parameters

No parameters

Response format

```
{
  "data": [{
    "name": string,
    "code": string,
    "value": number
  }],
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
name	Currency name, for instance, «Azerbaijan Manat»

	code	Currency code, for instance, «AZN»
	value	current exchange rate to RUR
errors		Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
messages		Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Amadeus error code
	text	Amadeus error text

Response example

```
{
  "data": [{
    "name": "Azerbaijan Manat",
    "code": "AZN",
    "value": 39.857316
  }, {
    "name": "Armenia Dram",
    "code": "AMD",
    "value": 0.13939014
  }, {
    "name": "US Dollar",
    "code": "USD",
    "value": 67.617942
  }, {
    "name": "Euro",
    "code": "EUR",
    "value": 76.895148
  }, {
    "name": "Indian Rupee",
    "code": "INR",
    "value": 0.9637674
  }, {
    "name": "Kazakhstan Tenge",
    "code": "KZT",
    "value": 0.18109692
  }, {
    "name": "Kyrgyzstan Som",
    "code": "KGS",
    "value": 0.96873888
  }, {
    "name": "China Yuan",
    "code": "CNY",
    "value": 9.8162046
  }, {

```

```
    "name": "Tajikistan Ruble",
    "code": "TJS",
    "value": 7.1774442
  }, {
    "name": "Uzbekistan Sum",
    "code": "UZS",
    "value": 0.0081475458
  }, {
    "name": "Czech Koruna",
    "code": "CZK",
    "value": 2.9621310
  }, {
    "name": "Japanese Yen",
    "code": "JPY",
    "value": 0.59577894
  }
],
"errors": [],
"success": true
}
```

Get Cyrillic transliteration rules service

API service	settings/transliteration (service)
Short description	Allows to receive Cyrillic transliteration rules
Supportes HTTP(S) methods	GET

GET settings/transliteration method

Description: requests Cyrillic transliteration rules

Request format /api/apiVersion/settings/transliteration

Request parameters

No parameters.

Response format

```
{
  "data": {
    "nativeLetter": string,
    "latinLetter": string
  },
  "errors": [
    {
      "ibeCode": number,
      "description": string,
      "messages": [
        {
          "code": string,
          "text": string,
          "additionalInfo": {
            "ssrType": string,
            "info": string
          }
        }
      ]
    }
  ],
  "success": boolean
}
```

Response parameters

Parameter	Description
-----------	-------------

success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
nativeLetter	Cyrillic symbol
latinLetter	Latin symbol
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN_Attachment 1. IBE error codes
description	Code description, see. EN_Attachment 1. IBE error codes
messages	Array of objects, containing errors from Amadeus response. Array can be empty, it is recommended to analyse errors by ibeCode.
code	Amadeus error code
text	Amadeus error text

Response example

```
{
  "data": [
    {
      "nativeLetter": "А",
      "latinLetter": "A"
    },
    {
      "nativeLetter": "а",
      "latinLetter": "a"
    },
    {
      "nativeLetter": "Б",
      "latinLetter": "B"
    },
    {
      "nativeLetter": "б",
      "latinLetter": "b"
    },
    {
      "nativeLetter": "Я",
      "latinLetter": "Ia"
    },
    {
      "nativeLetter": "я",
      "latinLetter": "ia"
    }
  ]
}
```



```
"errors": [],  
"success": true  
}
```

Service for monitoring the status of the backup engine contour

API service	system/Availability
Short description	Allows to monitor the status of the backup engine contour in IT Grad
Supported HTTP(S)-methods	GET

Description: The method is analogous to the already existing healthcheck method. Made specifically for checking the status of services on the backup engine circuit.

Check:

1. DB - connection availability
2. Redis - connection availability
3. Amadeus - getting a session by login/password
4. SIG - authorization, flight request
5. Rabbit - connection availability
6. CRM - availability of connection, three endpoints: discounts, authorization, rest
7. Partners API - availability Aeroexpress, Alfastrah, Sealine
8. GOI - connection availability
9. BPC - connection availability
10. Artsofte - airport dictionary, getHash

Request format: /api/system/Availability

Request parameters: None.

Response format:

```
{
  "data": [
    {
      "service": integer,
      "severityLevel": string,
      "isAvailable": boolean
    }
    ....
  ]
  "isAvailable": boolean
}
```

Response parameters:

Parameter	Description
service	service name
severityLevel	importance of service Low = 1, Medium = 2, High = 3
isAvailable	service availability sign

The last field isAvailable shows the overall health of the engine and is true only when all services are available with severityLevel = 3

Attachments

Attachment 1. IBE error codes

Error code	Description returned by IBE	Error Description
IBE errors		
1000	Unknown IBE error	Unhandled exception, reason unknown.
1001	Operation can not be performed due to invalid reservation state	Error gets raised if certain API method was called at improper moment in booking flow. For instance if api/services method was called right after api/session (booking was not created yet)
1002	IBE session pool waiting timeout exceeded	Error might appear when IBE attempts to obtain Amadeus session but session pool is not responding
1003	IBE session pool stateful sessions limit exceeded	Error appears at the moment when max amount of Stateful session exceeded. Stateful sessions are used for preliminary bookings
1004	Session pool cleaning operation error	Error deleting of expired session.
1005	Session pool is unable to acquire new session from Amadeus	Unable to get new Amadeus session.
1006	Failed to resolve timezone	Error might appear during failed attempt to refresh timezone dictionary
1007	Session pool is unable to acquire stateful session as it's expired	The engine tries to get information about the session in the Session Pool, but there is no such session.
1008	Data validation error	Validation error, might appear on all booking steps. For instance when invalid passenger data is being sent

1009	IBE health check failed	Unable to receive information from api/health method. Method is used by Azure for retrieving of data.
1010	Promotion is not found	Promo code is not found
1011	Operation can not be performed due to invalid payment order status	Error appears in case of attempt to modify booking within same session after receiving of payment link in order to avoid existing of two concurrent orders.
1012	Unable to cancel additional service with issued ticket	The error occurs when trying to cancel services using the api/services service, if the canceled service was paid for and a ticket was issued for it.
1013	Unable to load IBE reservation context	It is not possible to get information on the open session ID of the booking engine.
1014	IBE authentication error	Error appears in case of concurrent passing of Authorization header and X-API-Key.
1015	File type not allowed	Error appears in case of passing invalid file type in IBE API methods
1016	File size is too large	The error occurs when an attempt is made to transfer large files to the booking engine methods.
1017	Request body is too large	An error occurs when the request body is too large.
1018	Unable to change more then maximum allowed symbols	The error occurs when an attempt is made to change more characters in the passenger information than allowed. At the moment it is used for checking when changing the last name and first name

1019	Unable to change element twice	<p>The error occurs if you try to change passenger data again.</p> <p>At the moment it is used for checking when changing the last name and first name</p>
1020	Failed to update or retrieve search log data	Service error. Occurs when an attempt to write logs to the database is unsuccessful.
1021	Failed to update or retrieve reservation log data	Service error. Occurs when an attempt to read logs from the database is unsuccessful.
1022	Failed connect to redis	Service error. Occurs when an attempt to connect to Redis is unsuccessful.
1030	Unknown Refund error	Unknown error in init refund process.
1031	Refund sessions store has failed to process an operation	Error appears in case of failed read/write attempt of refund information to database.
1032	Refund process is locked by another process	Error appears in case of concurrent attempt to refund for the same PNR.
1033	There are incomplete refund processes	Error appears in case if one of refund processes for PNR hangs in non-terminal status. See Voluntary tickets and services refund service: api/refund
1040	The ticket refund info was not found	Error appears in case when ticket state in IBE database differs from PNR in Amadeus.
1041	Some passengers are not exist in reservation	An error occurs if the passengers transferred in the request were not found in the booking.
1042	The remaining passengers are of the wrong type	Appears in case when remaining passengers in booking has incorrect type. For instance attempt to refund booking for adult passenger in 1 adult + 1 child booking.
1043	The PNR was created not in IBE	Error appears in case when booking was created without usage of IBE API.

1044	One of the specified refund items is not found or not available for the confirmation	Error appears in case when it is not possible to confirm refund for one of the tickets (GOI-proxy service, PUT refund/pnr)
1050	The partner refund process returned error	The error occurs when an unsuccessful attempt to cancel partner services.
1060	The refund session has a problem	Error appears in case of refund process hanged in non-terminal state.
1080	OrderInfo client returned error	Error appears in case of original GOI returns error.
1084	Any Fare wasn't found by given reservation params	The error appears if the sent fare doesn't return in recommendations.
1085	Session is locked by another request	The error appears if the session is locked by another request.
1086	Reservation can't be annulled	The error appears if a reservation couldn't be annuled.
1092	Passenger documents in incorrect format	The error appears if passengers documents are sent in the incorrect format.
1093	Passenger gender is a required field	The error appears if the passengers gender is missed.
1094	Passenger birthdate is required field	The error appears if the passengers birthday is missed.
1095	Passenger firstname and surname in incorrect format	The error appears if the passengers first name or last name is sent in the incorrect format.
1096	Reservation prolongation is unavailable	The error appears if adding of MIS-segment isn't available.
1097	Passenger with infant is child	The error appears if there are no adults in the request on reservation with infants.
1098	Airport code not found	The error appears if the airport code not found.
1140	Reservation has no passenger information	An error occurs if there is no information about passengers in the booking.

1141	Some passenger in reservation has no document information	The error occurs if one of the passengers does not have information about the documents.
1500	Failed to receive exchange rates.	Unable to receive information for api/exchangerates. Method uses API CB RF for receiving of data.
1600	Send grid error	Error sending email about technical problem via Send Grid.
1800	Unknown social network error	Unhandled exception, reason unknown.
1801	Social network connectivity error	Error might appear in case of authorisation attempt via social networks.
1802	Social network connection timeout exceeded	An error can occur when trying to log in through social networks, there was a timeout in accessing the systems.
1810	External system temporary unavailable	The error occurs when the SIG request circuit breaker is triggered.
1811	Total request process did not complete within the timeout	The error occurs when the total requests processing did not complete within the timeout.
2000	Unknown Connector error	Unhandled exception, unknown cause.
2200	Airports service error	Unable to receive information from api/airports method. Method uses API u6.ru for receiving of data.
2201	Airports cache is empty	api/airports method did not return data
2202	Airports API returned invalid result	api/airports method returned invalid data
3000	Unknown Soap receiver service error	Unknown error in loading of flights-exceptions via IBE Soap receiver.
5000	Certificate is invalid	The api/Certificate/CheckCertificate method returned information that the requested certificate was not found. For example, if the certificate number or the owner's surname is incorrectly specified

6100	Weather service return error.	Occurs when an error is received from the weather service.
8000	Request to BPC did not complete	The error occurs if the request to BPC didn't complete.
9003	Seat map is not available due to incorrect billing	The error occurs if the place map returned from Amadeus without a price.
9004	Unknown IBE error. Null Ok response.	The error occurs if the response to the request to create a booking is returned empty.
Amadeus errors		
1100	Unknown Amadeus error	Unhandled exception, unknown cause.
1101	Amadeus connectivity error	Occurs when there is an error connecting to the Amadeus services.
1102	Amadeus connection timeout exceeded	Occurs when the Amadeus service is not responsible for the specified timeout.
1103	Amadeus Service_IntegratedCatalogue function returned error	Occurs when an error occurred while receiving a list of additional services from Amadeus.
1104	Amadeus Insurance_Pricing function returned error	Occurs when there was an error in obtaining information on the insurance entered in Amadeus.
1105	Amadeus Fare_PricePNRWithBookingClass function returned error	Occurs when there was an error in confirming the ticket price in the booking (pricing).
1106	Amadeus Service_IntegratedPricing function returned error	Occurs when there was an error in confirming the price of additional services in the booking.
1107	Amadeus Insurance_Booking function returned error	Occurs when there was an error booking insurance, entered into Amadeus.
1108	Amadeus Ticket_CreateTSTFromPricing function returned error	Error creating TST for booking.
1109	Amadeus Ticket_CreateTSMFromPricing function returned error	Error creating TSM for booking.
1110	Amadeus PNR_AddMultiElements function returned error	Error adding/deleting information to booking.

1111	Amadeus PNR_Retrieve function returned error	Error getting booking information.
1112	Amadeus PNR_Cancel function returned error	Cancellation error.
1113	Amadeus Air_RetrieveSeatMap function returned error	Error in getting seating information on board.
1114	Amadeus Ticket_RetrieveListOfTSM function returned error	Error in obtaining information about additional services of the airline in the booking.
1117	Amadeus Fare_PricePNRWithBookingClass function return error. No fare for booking code	The error occurs when trying to book a fare for which there are no tickets.
1118	Amadeus Ticket_DisplayTSMP function returned error	An error occurred in the Ticket_DisplayTSMP method
1119	Amadeus Ticket_UpdateTSMP function returned error	An error occurred in the Ticket_UpdateTSMP method
1120	Amadeus service returned fault of type Session	Invalid Amadeus session was used
1121	Amadeus PNR_Split function returned error	An error occurred in the PNR_Split method
1122	Unable to split PNR if children are left in PNR	Service error. At present is not used anywhere.
1130	Unknown Refund error	Voluntary refunds are not available as one of the passengers has successfully checked in for the flight.
1131	Unable to refund	Can't return tickets
1132	Amadeus DocRefund_IgnoreRefund function returned error	An error occurred in the DocRefund_IgnoreRefund method (the method is called after calculating information to be returned via Amadeus).
1133	No ATC tickets to refund	Error appears in case of attempt to initiate voluntary refund for booking with not issued tickets.

1150	Amadeus PNR not found. No match for record locator.	Error appears in case of attempt to retrieve information about not existing booking. Relates to 1931 "Application NO MATCH FOR RECORD LOCATOR" error.
CRM Errors		
1200	Unknown CRM error	Unhandled exception, unknown cause.
1201	CRM connectivity error	Occurs when there is an error connecting to CRM services.
1202	CRM connection timeout exceeded	Occurs when the CRM service is not responsible for the set timeout.
1203	CRM HoldPromoCode function returned error	Promo code holding error.
1204	CRM AgreePersonalInfo function returned error	Service error sending subscription information to the newsletter.
1205	CRM CheckPromoCode function returned error	Promo code verification error.
1206	CRM CalculateDiscount function returned error	Error of the service for obtaining a discount.
1207	CRM REST API getUser method returned error	CRM authorization service error.
1208	CRM CalculateDiscountList function returned error	Error in the service for obtaining a list of discounts.
1209	On this phone already created involuntary refund	The error occurs if the forced refund has already been initiated using the given phone number.
1210	Unknown Involuntary refund error	Unknown error during a forced checkout.
1211	Involuntary refund not found	No forced refund found.
1215	CRM food on board service has wrong result	Error receiving food on board.
1400	CRM receipt service returned error.	An error occurred in the itinerary receipt service.

1401	CRM receipt send email service returned error.	Error in the service of sending the itinerary receipt to email.
1402	CRM receipt timeout exceeded.	Timeout from the service for receiving/sending itinerary receipts.
BPC errors		
1300	Unknown Payment gate error	Unhandled exception, unknown cause.
1301	Payment gate hash value is incorrect	Wrong hash on order.
1302	Unknown Payment History error	Error appears in case when it is not possible to receive information about order history from BPC. Service is used in BMM in services upsale.
1303	Payment History Login returned error	The error occurs if it was not possible to log in to the BPC service.
1304	Payment History By Pnr returned error	The error occurs if an attempt to get the history of orders for a reservation is unsuccessful.
1306	Url for perform payment method is empty	The error occurs if there is no link to perform a payment.
1307	OrderId for perform payment method is empty	The error occurs if there is no order number for perform a payment.
TAIS errors		
1700	Unknown Tais error	Unhandled exception, unknown cause.
1701	Tais Connection Timeout	Occurs when the TAIS service is not responsible for the set timeout.
1702	Tais Connectivity Error	Error connecting to TAIS services.
1703	Tais Ftp server return error.	Error sending a report to the TAIS FTP-server
1755	Bad Request. Check parameters of request	SIG request error.
Errors in getting partner services		
1900	Aeroexpress services return error	Aeroexpress service returned an error.
1901	Aeroexpress connection timeout exceeded	Occurs when the Aeroexpress service is not responsible for the set timeout.

1902	Aeroexpress connectivity error	Error connecting to Aeroexpress services.
1950	Unknown Partner	An error occurs if the partner ID is unknown.
1951	Empty partner products list	Service error. Appears in case when it is not possible to receive list of partner's products from booking during report upload to SOFI.
1952	Partner product wasn't issued	Error appears when partners product's tickets were not issued and user attempts to initiate their resending.
1953	Unknown partner product booking ID	Error appears when it is not possible to find partner's service in booking via ID.
1957	A partner type is not available for current App Type	The error occurs when trying to add a partner service to a booking for an application type for which it is not available. For example, Alpha insurance for a mobile application.
DMS errors		
2100	Unknown DMS error	Unhandled exception, unknown cause.
2101	DMS connection timeout exceeded	Occurs when the DMS service is not responsible for the specified timeout.
OrderInfo (GOI proxy-service) errors		
4000	OrderInfo proxy service has failed to process request	Occurs when the proxy was unable to process the incoming request.
4001	OrderInfo proxy service has not found requested reservation or failed to load it	The requested reservation could not be found or the information on it was loaded.
4002	OrderInfo proxy service has failed to prepare partners products for issue operation	The error occurs if it was not possible to prepare partner services for release.
4003	OrderInfo proxy service has failed to issue partners products	Failed to release partner services.
4004	OrderInfo proxy service has failed to decrease fop amount on insurance amount	An error occurs if it was not possible to reduce the cost of booking by the amount of partner services.

4100	OrderInfo service has failed to process request	The error occurs if the GOI-proxy was unable to fulfill the request.
------	---	--

Adding services to booking error codes (only for API version 2.1 and above)

Error code	IBE description	Error description
1000	Unknown error	Unknown error. Any error not handled by IBE.
1001	Invalid passenger or segment reference	Error appears when attempting to add service for passenger or segment not existing in booking.
1007	Passenger reference cannot be empty if IsInfantProduct is true	Error appears when attempting to add alfamedicine to an infant without specifying a passengerReference
2001	Seat is unavailable for this passenger type.	Error appears when attempting to book seat for a child in emergency row or attempting to book seat for adult with an infant not in central row.
2002	Seat is unavailable	Error appears when attempting to book already booked seat.
3001	Unknown insurance code	Error appears when attempting to add insurance with invalid code to booking.
3002	Catalogue of insurance services is unavailable	Error appears when Amadeus insurance service is not available during adding of insurance to booking.
4001	Catalogue of services is unavailable	Error appears when Amadeus Catalogue service is not available during adding of services to booking.
4002	Invalid service code	The error occurs when you try to add a service with an incorrect code to your booking.
5000	Invalid product code	The error occurs when you try to add a partner service with an incorrect code to your booking.
5001	A partner type is not available for current App Type.	The error occurs when trying to add a partner service to a booking for an application type for which it is not available.
5006	Cannot be sold to non-citizens of the Russian Federation	Error appears when attempting to add a telemedicine service for a non-citizens of the Russian Federation.
5010	Unable to purchase the service after registering for a destination.	An error occurs if additional purchase of the service is not available after registration.

6007	Alfa Insurance Health services return error.	Alfamedicine services error
------	--	-----------------------------

Attachment 2. Dictionaries

Passenger types

paxType field value	Decryption	Passenger type
0	ADT, adult	Conventional
1	SRC, pensioner	Subsidised
2	YTH, youth	Subsidised
3	CHD, child	Conventional
4	INF, infant	Conventional
5	CNN, child	Subsidised
6	UNN, child	Conventional, Unaccompanied minors
7	STR, adult	Subsidised for Far East Region
8	JNN, child	Subsidised for Far East Region/family trips
9*	MIL, adult	Military
10*	DIS, adult	Disabled passenger
11*	MDP, adult	Military relative
12*	YCD, pensioner	Senior citizen, instead of SRC
13*	INS, infant	Infant with a seat
14*	SNN, pensioner	Senior citizen, instead of SRC
15	GCF, pensioner/adult	Subsidised for Far East Region/family trips
16	JCB, youth	Subsidised for Far East Region/family trips

* this type is not used when creating a reservation, it can be returned in a GET/reservation response.

Passenger sex

sex field value	Decryption
0	Male
1	Female

Document types

documentType field value	Decryption
0	Passport (general civil)
1	International passport
2	Birth certificate
3	Other

Passenger assistance types

assistanceType field value	Decryption
0	Without assistance
1	Wheelchair is needed on the way to/from aircraft
2	Need assistance in getting on and off board
3	Both points 1 and 2

Wheelchair types

wheelchairType field value	Decryption
0	Manual type
1	Electric type with dry battery
2	Electric type with liquid elements battery

Auxiliary services types

services.code field value	Category	Decryption
ABAG	luggage	First additional bag
BBAG	luggage	Second additional bag
CBAG	luggage	Third additional bag
DBAG	luggage	Additional bag 10 kg
HBAG	luggage	Extension of additional bag or default luggage (if allowed)
CHML	meal	Child meal
HTML	meal	Hot meat dish

MOML	meal	Muslim halal meal
SNML	meal	Sandwich meal
SFML	meal	Hot seafood
VGML	meal	Vegeterian meal
CAML	meal	Dessert
KSML	meal	Kosher
BGML	meal	Burger
AVML	meal	Vegetarian Hindu
HNML	meal	Indian Cuisine
VJML	meal	Vegetarian jain
TCML	meal	Hot breakfast
GFML	meal	Gluten free
BBML	meal	Baby meal
CPML	meal	Cheese platter
MPML	meal	Meat platter
GRML	meal	Greek salad
MTML	meal	Meat Steak Meal
UDML	meal	Seafood Udon Meal
FTML	meal	Fish Steak Meal
ECML	meal	English Breakfast
FCML	meal	French Breakfast
ACML	meal	American Breakfast
RCML	meal	Russian Breakfast
CSML	meal	Caucasian Cuisine
RUML	meal	Russian Cuisine
ORML	meal	Asian Cuisine
MEML	meal	Italian Cuisine
GEML	meal	German Cuisine
ASML	meal	Central Asian Cuisine

PETC	pets	Pet transportation
URFL	insurance	Insurance, avia bundle
URTR	insurance	Insurance, travel bundle
SEAT	seat	Seat selection
BLOU	room	Business room
VLOU	room	VIP room
CVID	covidExpress	Rapid analysis for covid-19
XBLL	handLuggage	Hand luggage up to 10 kg in the cabin (luggage in the cabin)
XBLS	handLuggage	Additional piece of carry-on baggage up to 10 kg in the cabin
NYML	meal	New Year meal

Types of additional services supported by the engine (not sold in the engine)

Значение поля serviceCategories.code	Категория	Расшифровка
CBBG	luggage	Luggage in the aircraft cabin with a separate seat
BIKE	luggage	Bicycle in the luggage compartment
SKI	luggage	Ski equipment
BULK	luggage	Additional carry-on baggage
XBAG	luggage	Oversized baggage
HBGS	luggage	Excess baggage
HBGT	luggage	Excess baggage
DBGS	luggage	Increase in baggage weight up to 32 kg
DBGT	luggage	Increase in baggage weight up to 32 kg
FRAG	luggage	Fragile luggage
WBAG	luggage	registration of payment for excess weight of checked baggage from 10 to 23 kg
SRXBLL	luggage	registration of payment for excess weight of hand luggage from 5 to 10 kg, not exceeding dimensions 55*40*20

Значение поля serviceCategories.code	Категория	Расшифровка
SRXBLS	luggage	registration of payment for an additional piece of hand luggage weighing no more than 5 kg and not exceeding dimensions 55*40*20
MEQT	luggage	Medical equipment
WEAP	luggage	Weapon
GOLF	luggage	Golf equipment
SPEQ	luggage	Sport equipment
RVML	meal	Raw vegetarian meal
VLML	meal	Vegetarian lacto-ovo meal
VOML	meal	Vegetarian oriental meal
DBML	meal	Diabetic food
BLML	meal	Diet food
LCML	meal	Low-calorie food
FPML	meal	Fruit nutrition
LFML	meal	Low-fat meals
NLML	meal	Low lactose meals
LSML	meal	Low salt diet
NFML	meal	Fish-free meals
SPML	meal	Special meals
JPML	meal	Japanese food
CFML	meal	Natural coffee cappuccino
CCML	meal	Natural coffee americano
AVIH	pets	Animal in the hold
SVAN	pets	Animal in the cabin
EXST	seat	Additional free space nearby
STRQ	seat	SEAT REQUEST, Seat
GPST	seat	Group seat request, specify number and preferred location, Seat

Значение поля serviceCategories.code	Категория	Расшифровка
RQST	seat	Seat request-include seat number or preference, Seat
CHNM	data	CHANGE NAME, Name change
NAMM	data	Payment of the fee for changing the name of the passenger on the ticket
NAME	data	Name - when airline holds reservation under a different name, Passanger's name change
PFEE	data	Payment of the fee for renewal at the call center
CKIN	data	Registration service
BLTW	data	Blocking the nearest seat
UMNR	other	Unaccompanied child
BUSS	other	Transfer by bus
TIME	other	Deferred payment service
WCHR	other	Wheelchair to/from the plane
WCHS	other	Wheelchair to climb/descend the airplane ladder
WCHC	other	Wheelchair for movement on the apron and the cabin of the aircraft
URFO	insurance	Avia
NSP1	insurance	Cancellation of a trip with insurance coverage of RUB 5,000
NSP2	insurance	Cancellation of a trip with insurance coverage of RUB 10000
NSP3	insurance	Cancellation of a trip with insurance coverage of RUB 20000
NSP4	insurance	Cancellation of a trip with insurance coverage of RUB 30000
NSP5	insurance	Cancellation of a trip with insurance coverage of RUB 40000

Package of services

packs.code field value	packs.includedServiceCode field value	packs.includedSeats field value	Name	Decryption
PAKA	VGML	COMF SEAT, [L, E]	SEAT & MEAL	Comfort seat and Vegeterian meal
PAKB	CHML	COMF SEAT, [L]	SEAT & MEAL	Comfort seat and Child meal
PAKC	PETC	CH SEAT, [CH, W, A]	SEAT & PETS	Standard paid seat and Pet Carriage
PAKD	ABAG	COMF SEAT, [L,E]	SEAT & LUGGAGE	Comfort seat and Extra baggage, 23 kg
PAKE	ABAG	CH SEAT, [CH, W, A]	SEAT & LUGGAGE	Standard paid seat and Extra baggage, 23 kg
PAKF	HTML	COMF SEAT, [L,E]	SEAT & MEAL	Comfort seat and Hot meal
PKDP	ABAG	COMF SEAT, [L,E]	SEAT & LUGGAGE	Comfort seat and Extra baggage, 23 kg for PROMO tariff
PKEP	ABAG	CH SEAT, [CH, W, A]	SEAT & LUGGAGE	Standard paid seat and Extra baggage, 23 kg for PROMO tariff
PAKR	RUML	COMF SEAT, [L,E]	SEAT & MEAL	Comfort seat and Russian Cuisine
PAKZ	ECML	COMF SEAT, [L,E]	SEAT & MEAL	Comfort seat and English breakfast

Types of animals

services.text field value	Decryption
DOG	Dog
CAT	Cat
PARROT	Parrot
CANARY	Canary

Most widely spread Seatmap seat characteristics

characteristics field value	Decryption
A	Aisle seat
CH	Chargeable seat
E	Exit row seat
I	Seat suitable for adult with an infant
L	Leg space seat
V	Seat to be left vacant or offered last
8	No seat at this location

Full list of characteristic can be found in Amadeus documentation for Air_RetrieveSeatMap 14.2. service

Promocode statuses

promotionStatus field value	Decryption
0	New
1	Used
2	Sent
3	Saved
4	Received
5	Hold

Tariff options

Option code	Category	Decryption
bestSeats	availablePaid	first rows in cabin of economy class (up to 15th row) if available at the moment of registration
upTo5bonuses	bonus	Bonus Accrual up to 5%
upTo10bonuses	bonus	Bonus Accrual up to 10%
upTo15bonuses	bonus	Bonus Accrual up to 15%
flexibilityOnChangesAndCancellations	ffServices	Possibility of ticket change and cancellation
refundExchangeProhibited	ffServices	Ticket refund is prohibited
handLuggage5kg	handBag	Hand luggage 5 kg
handLuggage15kg	handBag	Hand luggage 15 kg
oneSuitcase10kg	luggage	One suitcase 10 kg
oneSuitcase23kg	luggage	One suitcase 23 kg
oneSuitcase32kg	luggage	One suitcase 32 kg
twoSuitcase32kg	luggage	Two suitcase 32 kg
mealsOnBoard	menu	Standard meal
boarding	priorityBoarding	Priority boarding
checkin	priorityCheckIn	Priority checkin
businessClassCatering	specialMenu	Business class catering
upgradeToBusinessClass	upgradeToBusiness	Upgrade to business class
vipLounges	vipLounge	Vip lounge
twoSeatsOnRow	vipSeats	Two seats on row
vipSeats	vipSeats	Vip seats
noMealsOnBoard	menu	No meals

Main fare family categories

Amadeus category	SIG category	Decryption
ECOLIGHT	PR	«Promo» tariff

ECOFLEX	EC	«Econom» tariff
ECOPREM	FL	«Premium-Econom» tariff
BUSFLEX	BS	«Business» tariff
COMFORT	CF	«Comfort» tariff
PRLIGHT	PL	«Promo-lightt» tariff
BAGLESS	LC1	«Bagless» tariff
LUGGAGE	LC2	«Luggage» tariff
SUBSIDIZED	SUBSY	«Subsidised» tariff

«Business-light» (BUSLIGHT) tariff was deleted

IBE services severity

№	Service name	Code	Decryption
1	Airports/Get	1	Low
2	Calendar/Get	2	Medium
3	Catalogue/Get	2	Medium
4	Discounts/Get	2	Medium
5	Flights/Get	3	High
6	Households/Get	1	Low
7	Payment/Get	2	Medium
8	Payment/Post service version for API 1.0	3	High
9	Payment/PostV2 service version for API 1.1	3	High
10	Pricing/Post	3	High
11	Promotions/Post	1	Low
12	Promotions/Delete	1	Low
13	Receipt/Get	2	Medium
14	Receipt/Post	2	Medium

15	Renew/Post	2	Medium
16	Reservation/Get	3	High
17	Reservation/Post	3	High
18	Reservation/Delete	2	Medium
19	Seatmap/Get	2	Medium
20	Services/Post	2	Medium
21	ServicesPricing/Post	2	Medium
22	Token/Get	3	High
23	Token/Post	2	Medium
24	User/Post	2	Medium

High importance has services used in booking creation process.

Amadeus office types

№	Amadeus office ID	Office type	Description
1	SVXU6040A (deprecated), SVXU608MP (current)	1	Mobile office, booking via mobile application
2	SVXU608AZ, SVXU608AA	2	Web-office, booking via website
3	SVXU608SS	2	Web-office, booking made with redirect from Skyscanner
4	SVXU608AS	2	Web-office, booking made with redirect from Aviasales
5	MOWU608AK	3	IVR, all-center office
6	SVXU608YA	2	Web-office, booking created by transfer from Yandex.Avia
7	SVXU60072, MOWU608BL, SVXU608KN	5	Call-center offices, reservations created in the call center service
8	MOWU621AC	4	ADP-office, booking created by MyAgent
9	MOWU621OA, MOWU621MI, MOWU621CA, MOWU621MA	4	ADP-office, booking created by AviaCenter agent
10	SVXU621AA	4	ADP-office, booking through the website

11	SVXU621AB	4	ADP-office for the OrderInfo service
12	MOWU621TT	4	ADP-office, booking created by Tutu.ru
13	TJMU621TC	4	ADP-office, booking created by TQABC
14	LEDU621KB	4	ADP-office, booking created by Kupibilet
15	LEDU621IB	4	ADP-office, booking created by VIPSERVICE / INTERNET BILET (same as Kupibilet)
16	SVXU621BC		BPC
17	FRUU621KG	4	ADP-office, Kochevnik
18	MOWU621CT	4	ADP-office, Chabooka
19	MOWU621WT	4	ADP-office, WhiteTravel
20	MOWU621AT	4	ADP-office, AllianceTravel
21	KHVU621BD	4	ADP-office, BiletDV
22	CEKU621PT	4	ADP-office, PegasTour
23	LEDU621SL	4	ADP-office, Starliner
24	SKDU621AA	4	ADP-office, AtlantaAir
25	SVXU621CW	4	ADP-office, ClubWings
26	SVXU621BC	4	ADP-office, RBS
27	TASU621RT	4	ADP-office, RaiusTravel
28	MCXU621MT	4	ADP-office, MarwaTour
29	LEDU621DG	4	ADP-office, YesGC
30	SVXU621VA	4	ADP-office, VselennayaAvia
31	KRRU621AT	4	ADP-office, Azimuth2000
32	KJAU621RR	4	ADP-office, RazletRu
33	FRUU621BS	4	ADP-office, BiletStandartUSD
34	FRUU621SB	4	ADP-office, BiletStandartKGS
35	MOWU621TN	4	ADP-office, TeleTrainVIPService
36	HTAU621ES	4	ADP-office, EastSiberianSTC
37	KGDU621YK	4	ADP-office, YK
38	TASU621AT	4	ADP-office, AerodromTashkent

39	ROVU621RT	4	ADP-office, ReinaTour
40	FRUU621KC	4	ADP-office, KyrgyzConcept
41	KGDU621TT	4	ADP-office, TurneTrans
42	KGDU621WB	4	ADP-office, WebBooking
43	KGDU621TB	4	ADP-office, TTBooking
44	KGDU621VR	4	ADP-office, Vector
45	KGDU621TG	4	ADP-office, TransServiceGroup
46	AERU621AU	4	ADP-office, AviaTourne
47	MOWU621SA	4	ADP-office, AmigoS
48	MOWU621TR	4	ADP-office, TeleTrainVIPService1
49	LEDU621AT	4	ADP-office, AeroTour
50	MOWU621IM	4	ADP-office, InnovativeMobility
51	ROVU621RL	4	ADP-office, Liner
52	MOWU621TI	4	ADP-office, Talarii
53	KGDU621YA	4	ADP-office, AmberCruiseKorobova
54	KGDU621YB	4	ADP-office, AmberCruiseKvitka
55	KGDU621YC	4	ADP-office, AmberCruiseMikhasenok
56	SVXU621CP	4	ADP-office, PegasTourCP
57	SVXU621AC	4	ADP-office, PegasTourAC
58	CEKU621FM	4	ADP-office, PegasTourFM
59	CEKU621IT	4	ADP-office, PegasTourIT
60	CSYU621CH	4	ADP-office, PegasTourCH
61	SVXU621IM	4	ADP-office, PegasTourIM
62	SVXU621ST	4	ADP-office, PegasTourST
63	MOWU621AT	4	ADP-office, PegasTourAT
64	CEKU621PK	4	ADP-office, PegasTourPK
65	CEKU621PM	4	ADP-office, PegasTourPM
66	GDZU621PE	4	ADP-office, PegasTourPE
67	SVXU621PB	4	ADP-office, PegasTourPB

68	SVXU621SA	4	ADP-office, PegasTourSA
69	CEKU621TK	4	ADP-office, PegasTourTK
70	MOWU621AS	4	ADP-office, PegasTourAS
71	CEKU621LT	4	ADP-office, PegasTourLT
72	DYUU621FA	4	ADP-office, Faeton, currency TJS
73	EVNU621EA	4	ADP-office, ErebuniAir, currency USD
74	EVNU621CD	4	ADP-office, SVLand, currency USD
75	TASU621BK	4	ADP-office, Bookit, currency USD
76	LBDU621KT	4	ADP-office, KorvonT, currency TJS
77	SKDU621BT	4	ADP-office, BaseTravel, currency UZS
78	SVXU621PR	4	ADP-office, Piligrim94PR, currency RUB
79	SVXU621M9	4	ADP-office, Piligrim94M9, currency RUB
80	SVXU621P4	4	ADP-office, Piligrim94P4, currency RUB
81	MOWU621ZA	4	ADP-office, AgentRu, currency RUB
82	LEDU608BX	2	Web-office, VipService
83	MOWU621VP	4	ADP-офис, PRTB/ТА
84	KJAU621TT	4	ADP-офис, BiletTime
85	MOWU621MN	4	ADP-офис, Mirana

Countries

Country code	Russian name	English name
ABW	Аруба	Aruba
AFG	Афганистан	Afghanistan
AGO	Ангола	Angola
AIA	Ангилья	Anguilla
ALA	Аландские острова	Åland Islands
ALB	Албания	Albania
AND	Андорра	Andorra

ARE	ОАЭ	United Arab Emirates
ARG	Аргентина	Argentina
ARM	Армения	Armenia
ASM	Американское Самоа	American Samoa
ATA	Антарктида	Antarctica
ATF	Французские Южные и Антарктические Территории	French Southern Territories
ATG	Антигуа и Барбуда	Antigua and Barbuda
AUS	Австралия	Australia
AUT	Австрия	Austria
AZE	Азербайджан	Azerbaijan
BDI	Бурунди	Burundi
BEL	Бельгия	Belgium
BEN	Бенин	Benin
BES	Бонэйр Синт-Эстатиус и Саба	Bonaire Sint Eustatius and Saba
BFA	Буркина-Фасо	Burkina Faso
BGD	Бангладеш	Bangladesh
BGR	Болгария	Bulgaria
BHR	Бахрейн	Bahrain
BHS	Багамы	Bahamas
BIH	Босния и Герцеговина	Bosnia and Herzegovina
BLM	Сен-Бартелеми	Saint Barthélemy
BLR	Белоруссия	Belarus
BLZ	Белиз	Belize
BMU	Бермуды	Bermuda
BOL	Боливия	Bolivia
BRA	Бразилия	Brazil
BRB	Барбадос	Barbados
BRN	Бруней	Brunei Darussalam

BTN	Бутан	Bhutan
BVT	Остров Буве	Bouvet Island
BWA	Ботсвана	Botswana
CAF	ЦАР	Central African Republic
CAN	Канада	Canada
ССК	Кокосовые острова	Cocos (Keeling) Islands
CHE	Швейцария	Switzerland
CHL	Чили	Chile
CHN	КНР (Китайская Народная Республика)	China
CIV	Кот-д'Ивуар	Cote D'Ivoire
CMR	Камерун	Cameroon
COD	Демократическая Республика Конго	Congo the Democratic Republic of the
COG	Республика Конго	Congo
СОК	Острова Кука	Cook Islands
COL	Колумбия	Colombia
COM	Коморы	Comoros
CPV	Кабо-Верде	Cape Verde
CRI	Коста-Рика	Costa Rica
CUB	Куба	Cuba
CUW	Кюрасао	Curaçao
CXR	Остров Рождества	Christmas Island
СУМ	Острова Кайман	Cayman Islands
СУР	Кипр	Cyprus
CZE	Чехия	Czech Republic
DEU	Германия	Germany
DJI	Джибути	Djibouti
DMA	Доминика	Dominica
DNK	Дания	Denmark

DOM	Доминиканская Республика	Dominican Republic
DZA	Алжир	Algeria
ECU	Эквадор	Ecuador
EGY	Египет	Egypt
ERI	Эритрея	Eritrea
ESH	САДР	Western Sahara
ESP	Испания	Spain
EST	Эстония	Estonia
ETH	Эфиопия	Ethiopia
FIN	Финляндия	Finland
FJI	Фиджи	Fiji
FLK	Фолклендские острова	Falkland Islands (Malvinas)
FRA	Франция	France
FRO	Фареры	Faroe Islands
FSM	Микронезия	Micronesia Federated States of
GAB	Габон	Gabon
GBR	Великобритания	United Kingdom
GEO	Грузия	Georgia
GGY	Гернси	Guernsey
GHA	Гана	Ghana
GIB	Гибралтар	Gibraltar
GIN	Гвинея	Guinea
GLP	Гваделупа	Guadeloupe
GMB	Гамбия	Gambia
GNB	Гвинея-Бисау	Guinea-Bissau
GNQ	Экваториальная Гвинея	Equatorial Guinea
GRC	Греция	Greece
GRD	Гренада	Grenada
GRL	Гренландия	Greenland

GTM	Гватемала	Guatemala
GUF	Гвиана	French Guiana
GUM	Гуам	Guam
GUY	Гайана	Guyana
HKG	Гонконг	Hong Kong
HMD	Херд и Макдональд	Heard Island and McDonald Islands
HND	Гондурас	Honduras
HRV	Хорватия	Croatia
HTI	Гаити	Haiti
HUN	Венгрия	Hungary
IDN	Индонезия	Indonesia
IMN	Остров Мэн	Isle of Man
IND	Индия	India
IOT	Британская территория в Индийском океане	British Indian Ocean Territory
IRL	Ирландия	Ireland
IRN	Иран	Iran Islamic Republic of
IRQ	Ирак	Iraq
ISL	Исландия	Iceland
ISR	Израиль	Israel
ITA	Италия	Italy
JAM	Ямайка	Jamaica
JEY	Джерси	Jersey
JOR	Иордания	Jordan
JPN	Япония	Japan
KAZ	Казахстан	Kazakhstan
KEN	Кения	Kenya
KGZ	Киргизия	Kyrgyzstan
KHM	Камбоджа	Cambodia

KIR	Кирибати	Kiribati
KNA	Сент-Китс и Невис	Saint Kitts and Nevis
KOR	Республика Корея	Korea Republic of
KWT	Кувейт	Kuwait
LAO	Лаос	Lao People's Democratic Republic
LBN	Ливан	Lebanon
LBR	Либерия	Liberia
LBY	Ливия	Libyan Arab Jamahiriya
LCA	Сент-Люсия	Saint Lucia
LIE	Лихтенштейн	Liechtenstein
LKA	Шри-Ланка	Sri Lanka
LSO	Лесото	Lesotho
LTU	Литва	Lithuania
LUX	Люксембург	Luxembourg
LVA	Латвия	Latvia
MAC	Макао	Macao
MAF	Сен-Мартен	Saint Martin (French part)
MAR	Марокко	Morocco
MCO	Монако	Monaco
MDA	Молдавия	Moldova Republic of
MDG	Мадагаскар	Madagascar
MDV	Мальдивы	Maldives
MEX	Мексика	Mexico
MHL	Маршалловы Острова	Marshall Islands
MKD	Македония	Macedonia the Former Yugoslav Republic of
MLI	Мали	Mali
MLT	Мальта	Malta
MMR	Мьянма	Myanmar

MNE	Черногория	Montenegro
MNG	Монголия	Mongolia
MNP	Северные Марианские Острова	Northern Mariana Islands
MOZ	Мозамбик	Mozambique
MRT	Мавритания	Mauritania
MSR	Монтсеррат	Montserrat
MTQ	Мартиника	Martinique
MUS	Маврикий	Mauritius
MWI	Малави	Malawi
MYS	Малайзия	Malaysia
MYT	Майотта	Mayotte
NAM	Намибия	Namibia
NCL	Новая Каледония	New Caledonia
NER	Нигер	Niger
NFK	Остров Норфолк	Norfolk Island
NGA	Нигерия	Nigeria
NIC	Никарагуа	Nicaragua
NIU	Ниуэ	Niue
NLD	Нидерланды	Netherlands
NOR	Норвегия	Norway
NPL	Непал	Nepal
NRU	Науру	Nauru
NZL	Новая Зеландия	New Zealand
OMN	Оман	Oman
PAK	Пакистан	Pakistan
PAN	Панама	Panama
PCN	Острова Питкэрн	Pitcairn
PER	Перу	Peru
PHL	Филиппины	Philippines

PLW	Палау	Palau
PNG	Папуа — Новая Гвинея	Papua New Guinea
POL	Польша	Poland
PRI	Пуэрто-Рико	Puerto Rico
PRK	КНДР (Корейская Народно-Демократическая Республика)	Korea Democratic People's Republic of
PRT	Португалия	Portugal
PRY	Парагвай	Paraguay
PSE	Государство Палестина	Palestinian Territory Occupied
PYF	Французская Полинезия	French Polynesia
QAT	Катар	Qatar
REU	Реюньон	Reunion
ROU	Румыния	Romania
RUS	Россия	Russian Federation
RWA	Руанда	Rwanda
SAU	Саудовская Аравия	Saudi Arabia
SDN	Судан	Sudan
SEN	Сенегал	Senegal
SGP	Сингапур	Singapore
SGS	Южная Георгия и Южные Сандвичевы Острова	South Georgia and the South Sandwich Islands
SHN	Острова Святой Елены Вознесения и Тристан-да-Кунья	Saint Helena
SJM	Шпицберген и Ян-Майен	Svalbard and Jan Mayen
SLB	Соломоновы Острова	Solomon Islands
SLE	Сьерра-Леоне	Sierra Leone
SLV	Сальвадор	El Salvador
SMR	Сан-Марино	San Marino
SOM	Сомали	Somalia
SPM	Сен-Пьер и Микелон	Saint Pierre and Miquelon

SRB	Сербия	Serbia
SSD	Южный Судан	South Sudan
STP	Сан-Томе и Принсипи	Sao Tome and Principe
SUR	Суринам	Suriname
SVK	Словакия	Slovakia
SVN	Словения	Slovenia
SWE	Швеция	Sweden
SWZ	Свазиленд	Swaziland
SXM	Синт-Мартен	Sint Maarten (Dutch part)
SYC	Сейшельские Острова	Seychelles
SYR	Сирия	Syrian Arab Republic
TCA	Теркс и Кайкос	Turks and Caicos Islands
TCD	Чад	Chad
TGO	Того	Togo
THA	Таиланд	Thailand
TJK	Таджикистан	Tajikistan
TKL	Токелау	Tokelau
TKM	Туркмения	Turkmenistan
TLS	Восточный Тимор	Timor-Leste
TON	Тонга	Tonga
TTO	Тринидад и Тобаго	Trinidad and Tobago
TUN	Тунис	Tunisia
TUR	Турция	Turkey
TUV	Тувалу	Tuvalu
TWN	Китайская Республика	Taiwan
TZA	Танзания	Tanzania United Republic of
UGA	Уганда	Uganda
UKR	Украина	Ukraine

UMI	Внешние малые острова (США)	United States Minor Outlying Islands
URY	Уругвай	Uruguay
USA	США	United States of America
UZB	Узбекистан	Uzbekistan
VAT	Ватикан	Holy See (Vatican City State)
VCT	Сент-Винсент и Гренадины	Saint Vincent and the Grenadines
VEN	Венесуэла	Venezuela
VGB	Виргинские Острова (Великобритания)	Virgin Islands British
VIR	Виргинские Острова (США)	Virgin Islands U.S.
VNM	Вьетнам	Viet Nam
VUT	Вануату	Vanuatu
WLF	Уоллис и Футуна	Wallis and Futuna
WSM	Самоа	Samoa
XKX	Косово	Kosovo
YEM	Йемен	Yemen
ZAF	ЮАР	South Africa
ZMB	Замбия	Zambia
ZWE	Зимбабве	Zimbabwe

Attachment 3. Remarks/service SSR

IBE adds PNR remarks for technical purposes. Their type, format and purpose is displayed below.

Service SSR for discounts

SSR with information about applied discount

Description: SSR is added in tariff booking with discount

Format: DISCOUNT/DSNN/UsageType/CampaignId/LESSNN/SegmentRefernece/PassengerReference

Examples:

DISCOUNT/DS30/0/1-45RT6KP/LESS1972/S2/P2

DISCOUNT/DS07/0/1-45S5YUL/LESS363/S1/P2

Field	Description	Comment
DISCOUNT	Constant	-
DSNN	DS - constant NN - discount percent	Examples: DS05, DS15
UsageType	Discount application type	Fare = 0, Order = 1,
CampaignId	CRM discount campaign unique ID	Example: 1-45RT6KP
LESSNN	NN - discount amount (in RUR)	Example: LESS1972
SegmentRefernece	Reference to segment	Discount applies to all segment or not applies at all. Separate SSR is added for each segment.
PassengerReference	Reference to passenger	-

SSR with information about "Wings" card

Description: SSR is added in tariff booking with discount

Format: DISCOUNTCARD/CardNumber

CardNumber - wings card

Example: DISCOUNTCARD/1000444608

Service SSR for promo codes

Description: SSR is added when promo code is applied

Format of SSR for promo code for

flight: PROMO/Promocode/CampaignId/DSNN/LESSNN/SegmentRefernece/PassengerReference

Example: PROMO/IBE21/1-45H4OIC/DS06/LESS1264/S1/P2

Field	Description	Comment
PROMO	Constant	-
Promocode	Applied promo code	Example: IBE21
CampaignId	CRM promo campaign unique ID	Example: 1-45H4OIC
DSNN	DS - constant NN - discount percent	Example: DS05, DS15
LESSNN	NN - discount amount (in RUR)	Example: LESS1264
SegmentReference	Reference to segment	Discount applies to all segment or not applies at all. Separate SSR is added for each segment.
PassengerReference	Reference to passenger	-

Format of SSR for promo code for

service: PROMOCODE/ServiceCode/Promo/CampaignId/DSNN/LESSNN/SegmentRefernece/PassengerReference

Example: PROMOCODE/CHML/CHML01/1-45SADSN/DS07/LESS35/S1/P2

Field	Description	Comment
PROMOCODE	Constant	-
ServiceCode	Service code	Possible codes see EN Attachment 2. Dictionaries
Promo	Applied promo code	-
CampaignId	CRM promo campaign unique ID	Example: 1-45H4OIC
DSNN	DS - constant NN - discount percent	Example: DS05, DS15
LESSNN	NN - discount amount (in RUR)	Example: LESS1264
SegmentRefernece	Reference to segment	Discount applies to all segment or not applies at all. Separate SSR is added for each segment.
PassengerReference	Reference to passenger	-

Remark for tariff category

Description: Remark is being added for possibility of tariff option display in BMM after payment.

Format: outboundFare/returnFare

Example:

For One way: "ECOFLEXRU/"

For Return Trip: "ECOFLEXRU/BUSFLEXRU"

Postfix "RU" might differ depending on selected tariff.

Main fare family categories can be found in [Attachment 2. Dictionaries](#)

Utility SSRs for partner services

Description: SSR is added when adding a partner service to the booking to send information on the partner service in an sbr file

Format (similar to remark format):

PTR/PartnerId/ServiceId/Code/OrderId/Currency/Amount/Status/RefundAmount/PassengerReference/InfantFlag

Remark for partner's service

Description: remark is added for passing information to BPC/CRM about booked partner's service.

Format:

PTR/PartnerId/ServiceId/Code/OrderId/Currency/Amount/Status/RefundAmount/PassengerReference/InfantFlag/SegmentReference/IssuedOffice

Example (Aeroexpress):

"PTR/4/65241A17717E4BEEB378B82763795A03/STANDARTOW/520501/RUB/500.00/1/0.00/P2///SVXU608AZ"

Field	Description	Comment
PTR	Constant	-
PartnerId	Partner's unique identifier	Possible types described in <u>Attachment 2. Dictionaries.</u>
ServiceId	Unique order's id	GUID
Code	Partner's service code	Possible types described in <u>Attachment 2. Dictionaries.</u>
OrderId	Order's id in partner's system	-
Currency	Currency	RUB
Amount	Service cost	Float number

Status	<p>Service status.</p> <p>Possible values:</p> <p>NotIssued = 1, service is added to booking, awaits issuing</p> <p>Issued = 2, service is issued, obtained number of report document</p> <p>IssueFailed = 3, service is added to booking, but error appeared during service issuing</p> <p>Refunded = 4, service is cancelled, refund initiated</p>	<p>Status code number is used in remark</p> <p>If service status is changing old remark is deleted first, then new remark is added with new status code.</p>
RefundAmount	Refund amount	If refund was not initiated then field value is set to 0.00
PassengerReference	Reference to passenger for whom service is provided.	If service relates to particular passengers, and not for the whole booking, then references to all corresponding passengers are added.
InfantFlag	Obtains value “I”, if service is provided for an infant.	For Aeroexpress: never filled in
SegmentReference	Reference to segment	<p>A link to a segment is not added if the partner service is added to all segments. Currently only used for baggage packing partner.</p> <p>Example: S1</p>
IssuedOffice	Office under which the partner was added to the booking	-

Remark for applied markup

Description: Remark is added when markup is applied for certain airline's services.

Markups gets loaded from DMS.

Format:

PRICE/ServiceCode/ChargeAmount/SegmentReference/PassengerReference/ServiceReference/
ChargeType

Example: PRICE/ABAG/107/S1/P2/SSR12/1

Field	Description	Comment
PRICE	Constant	-
ServiceCode	Service code	Possible codes see <u>EN Attachment 2.</u> <u>Dictionaries</u>
ChargeAmount	Charge Amount (RUR)	Integer value
SegmentReference	Reference to segment on which service with markup was added	-
PassengerReference	Reference to passenger to whom service with markup is provided	-
ServiceReference	Reference to the additional service for which a markup has been added	-
ChargeType	Type of added markup	Possible values: 1 - DMS markup 2 - dynamic markup obtained from getContactMarkup

Attachment 4. Partner's services.

Partner codes

Partner code	Title
1	Alfastrah
2	Sealine insurance
3	Service
4	Aeroexpress
5	Ostrovok
6	TUI
7	Telemedicine
8	Baggage packing
9	Transfer
10	Car rent
11	Alfastrah (Anticovid)
12	Sealine (Insurance AntiVirus)
13	Business room
21	Alfamedicine

Partner types

Partner type	Decryption
1	Insurance
2	Aeroexpress
3	Medicine

Aeroexpress service codes

№	Service code	Tariff decryption
1	STANDARTOW	Standard
2	STANDARTRT	Round trip

Tariff description can be found on official Aeroexpress website: <https://aeroexpress.ru/aero/prices.html>

Alfastrah service codes

№	Service code	Tariff decryption
1	AVIA	Avia
2	TRAVEL	Travel
3	LUGGAGE	Luggage
4	SPORT	Sport
5	NOFLIGHT	No flight
6	AVIADYNAMIC	Avia with dynamic cost calculation
7	AVIADYNAMICBUS	Avia for Business fares with dynamic cost calculation

Tariff description <https://www.uralairlines.ru/passengers-info/partner-services/insurance/>

Alfastrah service codes (Anticovid)

№	Service code	Tariff decryption
1	ANTICOVID19	Anticovid Insurance for disclosure of COVID-19

Sealine service codes

№	Service code	Tariff decryption	Availability
1	SLAVIA	Avia	Yes
2	SLAVIAPLUS	Avia+	Yes
3	SLSPORT	Sport	Yes
4	SLPROMO	Promo	Yes
5	SLTRAVEL	Travel	Yes

Sealine service codes (COVID insurance)

№	Service code	Tariff decryption
1	SLANTICOVID	Страхование АнтиВирус

Sberhealth service codes

№	Service code	Tariff decryption	Availability
1	SBERDOC	Therapist and Pediatrician Online	Yes
2	SBERVET	Veterinarian Online	Yes

Alfahealth service codes

№	Service code	Tariff decryption	Availability
1	AFLADOC	Therapist and Pediatrician Online	Yes

Business rooms service codes

№	Service code	Tariff decryption	Availability
1	BLN	Business room	Yes
2	VLN	VIP room	Yes

Attachment X. Request and response examples

Examples of using of flightDurationUntilNextStop field

Request:

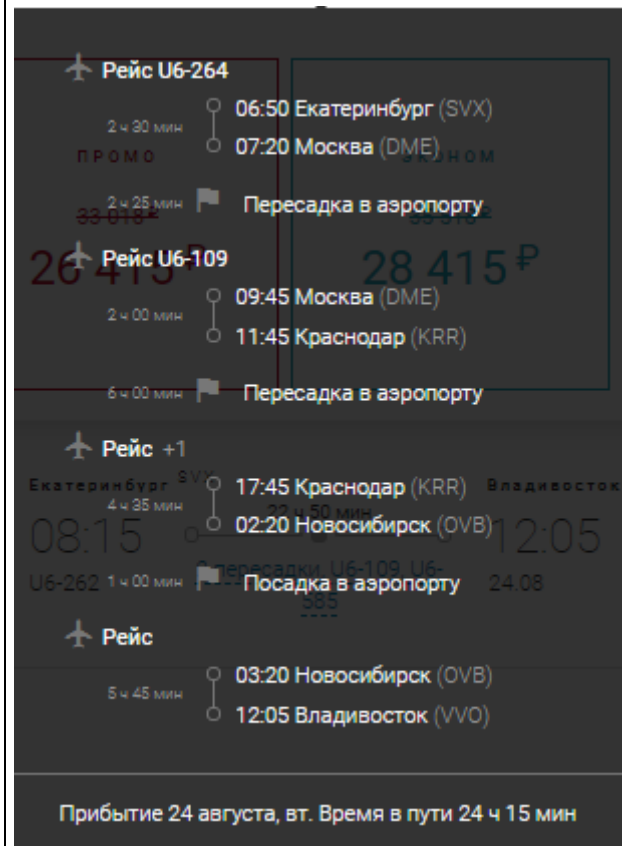
`/api/flights?orig=SVX&dest=VVO&dep=20210823&ret=&adt=1&chd=0&inf=0&src=0&yth=0`

Lets analyse one of the outboundFlights array elements:


```

{
  "flightGroupId": 0,
  "flights": [
    {
      "flightId": 0,
      "departureAirport": "SVX",
      "arrivalAirport": "DME",
      "departureDateTime": "2021-08-23T08:15:00",
      "arrivalDateTime": "2021-08-23T08:45:00",
      "aircraft": "Airbus A320",
      "aircraftIataCode": "320",
      "flightDuration": "2:30",
      "stopDuration": "1:00",
      "flightNumber": "262",
      "marketingCarrier": "U6",
      "operatingCarrier": "U6",
      "hash": "08/23/2021 08:15:00-262-SVX-DME"
    },
    {
      "flightId": 1,
      "departureAirport": "DME",
      "arrivalAirport": "KRR",
      "departureDateTime": "2021-08-23T09:45:00",
      "arrivalDateTime": "2021-08-23T11:45:00",
      "aircraft": "Airbus A320",
      "aircraftIataCode": "320",
      "flightDuration": "2:00",
      "stopDuration": "6:00",
      "flightNumber": "109",
      "marketingCarrier": "U6",
      "operatingCarrier": "U6",
      "hash": "08/23/2021 09:45:00-109-DME-KRR"
    },
    {
      "flightId": 2,
      "departureAirport": "KRR",
      "arrivalAirport": "VVO",
      "departureDateTime": "2021-08-23T17:45:00",
      "arrivalDateTime": "2021-08-

```



<pre> 24T12:05:00", "aircraft": "Airbus A320", "aircraftIataCode": "320", "techStops": [{ "arrivalDateTime": "2021-08- 24T02:20:00", "departureDateTime": "2021-08- 24T03:20:00", "airportCode": "OVB", "stopDuration": "1:00", "flightDurationUntilNextStop": "5:45" }], "flightDuration": "11:20", "flightDurationUntilNextStop": "4:35", "flightNumber": "585", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "08/23/2021 17:45:00-585-KRR- VVO" }], "flightGroupDuration": "22:50", "hash": "08/23/2021 08:15:00-262-SVX- DME,08/23/2021 09:45:00-109-DME- KRR,08/23/2021 17:45:00-585-KRR-VVO" } </pre>	
---	--

First and last airports in list matches flights[0] -> departureAirport (SVX) and flights[1] -> arrivalAirport (VVO).

The first and last airports on the list correspond to flights[0] -> departureAirport (LED) и flights[1] -> arrivalAirport (VVO).

The initial departure time corresponds in the same way flights[0] -> departureDateTime (08:15) и flights[1] -> arrivalDateTime (11:45) for these airports.

If the flight does not contain technical stops, then its duration is taken from the field flights -> flightDuration (as for the first flight, "flightDuration": "2:30"), and the time of transfer from flights -> stopDuration ("stopDuration": "1:00").

The thert flight contains a technical stop, so the value "flightDuration": "11:20" we get, if we add all the values flightDurationUntilNextStop (flight time to next stop) и stopDuration (stop time).

Flight time KRR – VVO taken from flights[1] -> flightDurationUntilNextStop = 4:35, which is the same as the calculated value: flights[1] -> techStops -> arrivalDateTime (17:45) subtract flights[1] -> departureDateTime (02:20), that is 02:20 – 17:45 = 14:45 minus ten hours difference equals 4:35.

Then, the stop time is taken from the field flights[1] -> techStops -> stopDuration.

The next flight time is taken from the field flights[1] -> techStops -> flightDurationUntilNextStop.

Value "flightGroupDuration": "22:50" is coincides with the sum of the duration of all flights and stops: 2:30 + 1:00 + 2:00 + 6:00 + 11:20.

Service reservation, POST method. Creating a pre-booking

One way flight	
Request, sending flight data	Response

```

{
  "flights": {
    "outboundFlights": [
      {
        "origin": "LED",
        "destination": "DME",
        "departureDate": "2021-07-
31T16:00:00.000Z",
        "flightNumber": "60",
        "classOfService": "A"
      }
    ],
    "returnFlights": null,
    "adults": 1,
    "children": 0,
    "infants": 0,
    "pensioners": 0,
    "youths": 0,
    "unaccompaniedMinors": 0,
    "strAdults": 0,
    "jnnChildren": 0,
    "origin": "LED",
    "farEastPensioners": 0,
    "farEastYouths": 0,
    "destination": "MOW",
    "passengersCount": 1
  },
  "fares": {
    "outboundFareFamily": {
      "category": "ECOFLEX",
      "code": "ECOFLEXRU",
      "isSubsidized": false
    },
    "returnFareFamily": null,
    "fares": [
      {
        "fareBasis": "AECOW",
        "isBreakpoint": true,
        "passengerType": 0,
        "flightDirection": 0,
        "flightIndex": 0
      }
    ],
    "sessionMarker": null
  },
  "expectedPrice": 3569,

```

```

{
  "data": {
    "journey": {
      "outboundFlights": [
        {
          "aircraftIataCode": "32N",
          "referenceNumber": "1",
          "origin": "LED",
          "destination": "DME",
          "departureDate": "2021-07-31T16:00:00",
          "arrivalDate": "2021-07-31T17:30:00",
          "flightNumber": "60",
          "classOfService": "A",
          "operatingCarrier": "U6",
          "marketingCarrier": "U6",
          "techStops": [],
          "departureTerminal": "1",
          "hash": "07/31/2021 16:00:00-60-LED-
DME"
        }
      ],
      "outboundFareFamily": {
        "code": ""
      },
      "origin": "LED",
      "destination": "DME",
      "departureDate": "2021-07-31T16:00:00",
      "outboundBaggageAllowances": [],
      "returnBaggageAllowances": []
    },
    "passengers": [],
    "services": [],
    "seats": [],
    "insurances": [],
    "partnerProducts": [],
    "hasSpecialConditions": false,
    "isRefundAvailable": false,
    "isAnnulAvailable": true,
    "isAgentReservation": false
  },
  "errors": [],
  "success": true
}

```

"searchPrice": 3569 }	
Request, sending passenger data (1 adult)	Response

<pre> { "flights": null, "passengers": { "passengers": [{ "id": 1, "firstName": "Test", "middleName": "Testovich", "surname": "Testov", "passengerType": 0, "birthDate": "2000-01-01", "sex": 0, "nationality": "RUS", "documentNumber": "1234567890", "documentType": 0, "documentExpiryDate": null, "loyaltyCard": "1234567890", "loyaltyCardValidTo": null, "phone": "71234567890", "email": "test@test.test", "disability": null }], "userAgreement": true }, "fares": { "outboundFareFamily": { "category": "ECOFLEX", "code": "ECOFLEXRU", "isSubsidized": false }, "returnFareFamily": null, "fares": [{ "fareBasis": "AECOW", "isBreakpoint": true, "passengerType": 0, "flightDirection": 0, "flightIndex": 0 }], "sessionMarker": null }, "expectedPrice": 3569, "searchPrice": 3569 } </pre>	<pre> { "data": { "journey": { "outboundFlights": [{ "aircraftIataCode": "32N", "referenceNumber": "1", "origin": "LED", "destination": "DME", "departureDate": "2021-07-31T16:00:00", "arrivalDate": "2021-07-31T17:30:00", "flightNumber": "60", "classOfService": "A", "operatingCarrier": "U6", "marketingCarrier": "U6", "hash": "07/31/2021 16:00:00-60-LED-DME" }], "outboundFareFamily": { "code": "ECOFLEXRU" }, "origin": "LED", "destination": "DME", "departureDate": "2021-07-31T16:00:00" }, "number": "PNRNUMBER", "reservationDate": "2021-07-24T05:08:00", "bookingTimeLimit": "2021-07-24T13:10:00", "passengers": [{ "id": 1, "hasInfant": false, "referenceNumber": "2", "firstName": "TEST", "middleName": "TESTOVICH", "surname": "TESTOV", "passengerType": 0, "birthDate": "2000-01-01T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "1234567890", "documentExpiryDate": "2031-07-24T00:00:00", "loyaltyCard": "1234567890", "email": "TEST@TEST.TEST", </pre>
---	--

	<pre> "phone": "71234567890", "documentType": 0, "isAdult": true, "isChanged": false }], "services": [], "seats": [], "insurances": [], "partnerProducts": [], "ticketFares": [{ "referenceNumber": "1", "referenceQualifier": "TST", "baseAmount": 1470.0, "totalAmount": 3569.0, "passengerReferences": [{ "referenceNumber": "2", "referenceQualifier": "PA" }] }, "passengerType": 0 },], "ticketFaresWithoutZapOff": [{ "referenceNumber": "1", "referenceQualifier": "TST", "baseAmount": 1470.0, "totalAmount": 3569.0, "passengerReferences": [{ "referenceNumber": "2", "referenceQualifier": "PA" }] }, "passengerType": 0 },], "hasSpecialConditions": false, "isRefundAvailable": false, "isAnnulAvailable": true, "isAgentReservation": false }, "errors": [], </pre>
--	---

	<pre>"success": true } "number": "PNRNUMBER" – the response will contain the real number PNR</pre>
Round-trip flight	
Request, sending flight data	Response


```

{
  "flights": {
    "outboundFlights": [
      {
        "origin": "LED",
        "destination": "DME",
        "departureDate": "2021-08-18T16:00:00.000Z",
        "flightNumber": "60",
        "classOfService": "A"
      }
    ],
    "returnFlights": [
      {
        "origin": "DME",
        "destination": "LED",
        "departureDate": "2021-08-25T13:35:00.000Z",
        "flightNumber": "59",
        "classOfService": "A"
      }
    ],
    "adults": 1,
    "children": 1,
    "infants": 0,
    "pensioners": 0,
    "youths": 0,
    "unaccompaniedMinors": 0,
    "strAdults": 0,
    "jnnChildren": 0,
    "origin": "LED",
    "farEastPensioners": 0,
    "farEastYouths": 0,
    "destination": "MOW",
    "passengersCount": 2
  },
  "fares": {
    "outboundFareFamily": {
      "category": "ECOFLEX",
      "code": "ECOFLEXRU",
      "isSubsidized": false
    },
    "returnFareFamily": {
      "category": "ECOFLEX",
      "code": "ECOFLEXRU",
      "isSubsidized": false
    }
  }
}

```

```

{
  "data": {
    "journey": {
      "outboundFlights": [
        {
          "aircraftIataCode": "32N",
          "referenceNumber": "1",
          "origin": "LED",
          "destination": "DME",
          "departureDate": "2021-08-18T16:00:00",
          "arrivalDate": "2021-08-18T17:30:00",
          "flightNumber": "60",
          "classOfService": "A",
          "operatingCarrier": "U6",
          "marketingCarrier": "U6",
          "techStops": [],
          "departureTerminal": "1",
          "hash": "08/18/2021 16:00:00-60-LED-DME"
        }
      ],
      "returnFlights": [
        {
          "aircraftIataCode": "32N",
          "referenceNumber": "2",
          "origin": "DME",
          "destination": "LED",
          "departureDate": "2021-08-25T13:35:00",
          "arrivalDate": "2021-08-25T14:55:00",
          "flightNumber": "59",
          "classOfService": "A",
          "operatingCarrier": "U6",
          "marketingCarrier": "U6",
          "techStops": [],
          "arrivalTerminal": "1",
          "hash": "08/25/2021 13:35:00-59-DME-LED"
        }
      ],
      "outboundFareFamily": {
        "code": ""
      },
      "origin": "LED",
      "destination": "DME",
      "departureDate": "2021-08-18T16:00:00",
      "returnDate": "2021-08-25T13:35:00",
    }
  }
}

```

<pre> }, "fares": [{ "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 0, "flightDirection": 0, "flightIndex": 0 }, { "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 5, "flightDirection": 0, "flightIndex": 0 }, { "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 0, "flightDirection": 1, "flightIndex": 0 }, { "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 5, "flightDirection": 1, "flightIndex": 0 }], "sessionMarker": null }, "expectedPrice": 11347, "searchPrice": 11347 } </pre>	<pre> "outboundBaggageAllowances": [], "returnBaggageAllowances": [] }, "passengers": [], "services": [], "seats": [], "insurances": [], "partnerProducts": [], "hasSpecialConditions": false, "isRefundAvailable": false, "isAnnulAvailable": true, "isAgentReservation": false }, "errors": [], "success": true } </pre>
<p>Request, sending passenger data (1 adult and 1 child)</p>	<p>Response</p>

<pre> { "flights": null, "passengers": { "passengers": [{ "id": 1, "firstName": "Test", "middleName": null, "surname": "Testov", "passengerType": 0, "birthDate": "2000-01-01", "sex": 0, "nationality": "RUS", "documentNumber": "1234567890", "documentType": 0, "documentExpiryDate": null, "loyaltyCard": null, "loyaltyCardValidTo": null, "phone": "71234567890", "email": "test@test.test", "disability": null }, { "id": 2, "firstName": "Test", "middleName": null, "surname": "Testov", "passengerType": 5, "birthDate": "2010-01-01", "sex": 1, "nationality": "RUS", "documentNumber": "1234567890", "documentType": 2, "documentExpiryDate": null, "loyaltyCard": null, "loyaltyCardValidTo": null, "phone": null, "email": null, "disability": null }], "userAgreement": true }, "fares": { "outboundFareFamily": { "category": "ECOFLEX", </pre>	<pre> { "data": { "journey": { "outboundFlights": [{ "aircraftIataCode": "32N", "referenceNumber": "1", "origin": "LED", "destination": "DME", "departureDate": "2021-08-18T16:00:00", "arrivalDate": "2021-08-18T17:30:00", "flightNumber": "60", "classOfService": "A", "operatingCarrier": "U6", "marketingCarrier": "U6", "hash": "08/18/2021 16:00:00-60-LED-DME" }], "returnFlights": [{ "aircraftIataCode": "32N", "referenceNumber": "2", "origin": "DME", "destination": "LED", "departureDate": "2021-08-25T13:35:00", "arrivalDate": "2021-08-25T14:55:00", "flightNumber": "59", "classOfService": "A", "operatingCarrier": "U6", "marketingCarrier": "U6", "hash": "08/25/2021 13:35:00-59-DME-LED" }], "outboundFareFamily": { "code": "ECOFLEXRU" }, "returnFareFamily": { "code": "ECOFLEXRU" }, "origin": "LED", "destination": "DME", "departureDate": "2021-08-18T16:00:00", "returnDate": "2021-08-25T13:35:00" } }, </pre>
---	---

<pre> "code": "ECOFLEXRU", "isSubsidized": false }, "returnFareFamily": { "category": "ECOFLEX", "code": "ECOFLEXRU", "isSubsidized": false }, "fares": [{ "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 0, "flightDirection": 0, "flightIndex": 0 }, { "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 5, "flightDirection": 0, "flightIndex": 0 }, { "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 0, "flightDirection": 1, "flightIndex": 0 }, { "fareBasis": "AECRT", "isBreakpoint": true, "passengerType": 5, "flightDirection": 1, "flightIndex": 0 }], "sessionMarker": null }, "expectedPrice": 11347, "searchPrice": 11347 } </pre>	<pre> "number": "PNRNUMBER", "reservationDate": "2021-07-24T05:17:00", "bookingTimeLimit": "2021-07-24T13:18:00", "passengers": [{ "id": 1, "hasInfant": false, "referenceNumber": "3", "firstName": "TEST", "middleName": "", "surname": "TESTOV", "passengerType": 0, "birthDate": "2000-01-01T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "1234567890", "documentExpiryDate": "2031-07-24T00:00:00", "email": "TEST@TEST.TEST", "phone": "71234567890", "documentType": 0, "isAdult": true, "isChanged": false }, { "id": 2, "hasInfant": false, "referenceNumber": "2", "firstName": "TEST", "middleName": "", "surname": "TESTOV", "passengerType": 5, "birthDate": "2010-01-01T00:00:00", "sex": 1, "nationality": "RUS", "documentNumber": "1234567890", "documentExpiryDate": "2031-07-24T00:00:00", "email": "TEST@TEST.TEST", "phone": "71234567890", "documentType": 0, "isAdult": false, "isChanged": false }], "services": [], </pre>
--	---

	<pre> "seats": [], "insurances": [], "partnerProducts": [], "ticketFares": [{ "referenceNumber": "1", "referenceQualifier": "TST", "baseAmount": 1410.0, "totalAmount": 5274.0, "passengerReferences": [{ "referenceNumber": "2", "referenceQualifier": "PA" }], "passengerType": 5 }, { "referenceNumber": "2", "referenceQualifier": "TST", "baseAmount": 1875.0, "totalAmount": 6073.0, "passengerReferences": [{ "referenceNumber": "3", "referenceQualifier": "PA" }], "passengerType": 0 }], "ticketFaresWithoutZapOff": [{ "referenceNumber": "1", "referenceQualifier": "TST", "baseAmount": 1410.0, "totalAmount": 5274.0, "passengerReferences": [{ "referenceNumber": "2", "referenceQualifier": "PA" }], "passengerType": 5 }, { </pre>
--	--

	<pre> "referenceNumber": "2", "referenceQualifier": "TST", "baseAmount": 1875.0, "totalAmount": 6073.0, "passengerReferences": [{ "referenceNumber": "3", "referenceQualifier": "PA" }], "passengerType": 0 }], "hasSpecialConditions": false, "isRefundAvailable": false, "isAnnulAvailable": true, "isAgentReservation": false }, "errors": [], "success": true } </pre>
--	---

Reservation service, GET method. Retrieving booking information

Booking with additional services and discounts. One way flight

Request string: /api/v2.3/Reservation?pnrNumber=JJ94XM&lastName=TESTOV

Response:

```

{
  "data": {
    "tickets": [
      {
        "number": "2622475454680",
        "passengerReference": "2",
        "isInfantTicket": false
      }
    ],
    "outboundFareFamily": {
      "fareFamilyId": 0,
      "code": "ECOFLEXRU",
      "category": "ECOFLEX",
      "isSubsidized": false,
      "options": [
        {
          "code": "flexibilityOnChangesAndCancellations",
          "category": "ffServices",
          "order": 5,

```

```

    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": true,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "mealsOnBoard",
    "category": "menu",
    "order": 3,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "upTo10bonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": false,
    "showInSummary": false,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": false,
    "showForInfant": false,
    "showInBookingCart": false
  },
  {
    "code": "handLuggage5kg",
    "category": "handBag",
    "order": 2,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "oneSuitcase23kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": true,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": false,
    "showInBookingCart": true
  },
  {

```

```

    "code": "oneSuitcase10kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": false,
    "showForInfant": true,
    "showInBookingCart": false
  },
  {
    "code": "canPayByBonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": false,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": false,
    "showInBookingCart": true
  },
  {
    "code": "upTo5bonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "noMealsOnBoard",
    "category": "menu",
    "order": 3,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  }
],
"order": 250,
"commercialFamily": "U6ECONOMY",
"canRefund": true,
"hasUpsale": true,
"isLowCost": false
},
"appliedDiscounts": [
  {
    "flightReference": "1",

```



```

    "amount": 713,
    "id": "1-4N9MCYL",
    "type": "DISCOUNT",
    "passengerReference": "2",
    "serviceCode": "",
    "usageType": 1,
    "promotionType": 0
  }
],
"baseFarePrices": [
  {
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantFare": false,
    "totalPrice": 2856.0,
    "basePrice": 757.0,
    "discountAmount": 713.0,
    "priceHidden": false
  }
],
"spentBonusesByPassengers": [],
"appliedCertificates": [],
"originOfficeId": "SVXU608AZ",
"originOfficeType": 2,
"isPayForbiddenForAgency": false,
"foodOnBoards": [],
"reservationRetrieveStatus": 0,
"engineType": 1,
"isGroupReservation": false,
"journey": {
  "outboundFlights": [
    {
      "aircraft": "Airbus A319",
      "aircraftIataCode": "319",
      "referenceNumber": "1",
      "origin": "LED",
      "destination": "DME",
      "departureDate": "2021-08-18T17:45:00",
      "arrivalDateUtc": "2021-08-18T16:15:00Z",
      "departureDateUtc": "2021-08-18T14:45:00Z",
      "arrivalDate": "2021-08-18T19:15:00",
      "flightNumber": "80",
      "classOfService": "A",
      "operatingCarrier": "U6",
      "marketingCarrier": "U6",
      "flightDuration": "1:30",
      "techStops": [],
      "departureTerminal": "1",
      "hash": "08/18/2021 17:45:00-80-LED-DME"
    }
  ]
},
"outboundFareFamily": {
  "code": "ECOFLEXRU",
  "category": "ECOFLEX"
},
"outboundDuration": "1:30",
"origin": "LED",

```

```

"destination": "DME",
"departureDate": "2021-08-18T17:45:00",
"outboundBaggageAllowances": [
  {
    "quantity": 1,
    "weight": 23,
    "passengerType": 0
  }
],
"returnBaggageAllowances": [],
"containsInternationalFlights": false
},
"number": "JJ94XM",
"reservationDate": "2021-07-24T05:26:00",
"passengers": [
  {
    "id": 1,
    "hasInfant": false,
    "referenceNumber": "2",
    "firstName": "TEST",
    "middleName": "",
    "surname": "TESTOV",
    "passengerType": 0,
    "birthDate": "2000-01-01T00:00:00",
    "sex": 0,
    "nationality": "RUS",
    "documentNumber": "1234567890",
    "documentExpiryDate": "2031-07-24T00:00:00",
    "email": "TEST@TEST.TEST",
    "phone": "71234567890",
    "documentType": 0,
    "isAdult": true,
    "isChanged": false,
    "checkedIn": {
      "1": false
    },
    "checkInHasStarted": {
      "1": false
    }
  }
],
"services": [
  {
    "code": "BGML",
    "status": "HK",
    "category": "meal",
    "passengerReference": "2",
    "flightReference": "1",
    "referenceNumber": "13",
    "ticketNumber": "2624552537101",
    "finalPrice": 450.0,
    "price": 450.0,
    "discountAmount": 0.0
  },
  {
    "code": "ABAG",
    "status": "HK",

```

```

        "category": "luggage",
        "passengerReference": "2",
        "flightReference": "1",
        "referenceNumber": "14",
        "ticketNumber": "2624552537102",
        "finalPrice": 2200.0,
        "price": 2200.0,
        "discountAmount": 0.0
    }
],
"seats": [
    {
        "number": 5,
        "title": "B",
        "code": "RQST",
        "status": "HK",
        "passengerReference": "2",
        "flightReference": "1",
        "referenceNumber": "15",
        "ticketNumber": "2624552537100",
        "finalPrice": 300.0,
        "price": 300.0,
        "discountAmount": 0.0
    }
],
"insurances": [],
"partnerProducts": [],
"hasSpecialConditions": false,
"isRefundAvailable": true,
"isAnnulAvailable": true,
"isAgentReservation": false
},
"errors": [],
"success": true
}

```

Round trip booking

Request string: /api/v2.3/Reservation?pnrNumber=JJANVQ&lastName=TESTOV

Response:

```

{
  "data": {
    "tickets": [
      {
        "number": "2622475454681",
        "passengerReference": "2",
        "isInfantTicket": false
      }
    ],
    "outboundFareFamily": {
      "fareFamilyId": 0,

```

```

"code": "ECOFLEXRU",
"category": "ECOFLEX",
"isSubsidized": false,
"options": [
  {
    "code": "flexibilityOnChangesAndCancellations",
    "category": "ffServices",
    "order": 5,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": true,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "mealsOnBoard",
    "category": "menu",
    "order": 3,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "upTo10bonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": false,
    "showInSummary": false,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": false,
    "showForInfant": false,
    "showInBookingCart": false
  },
  {
    "code": "handLuggage5kg",
    "category": "handBag",
    "order": 2,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "oneSuitcase23kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,

```

```

    "showInSummary": true,
    "showInUpsell": true,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": false,
    "showInBookingCart": true
  },
  {
    "code": "oneSuitcase10kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": false,
    "showForInfant": true,
    "showInBookingCart": false
  },
  {
    "code": "canPayByBonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": false,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": false,
    "showInBookingCart": true
  },
  {
    "code": "upTo5bonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "noMealsOnBoard",
    "category": "menu",
    "order": 3,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  }
],
"order": 250,

```

```

"commercialFamily": "U6ECONOMY",
"canRefund": true,
"hasUpsale": true,
"isLowCost": false
},
"returnFareFamily": {
  "fareFamilyId": 0,
  "code": "ECOFLEXRU",
  "category": "ECOFLEX",
  "isSubsidized": false,
  "options": [
    {
      "code": "flexibilityOnChangesAndCancellations",
      "category": "ffServices",
      "order": 5,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": true,
      "showInFareIncluded": true,
      "showForAdult": true,
      "showForInfant": true,
      "showInBookingCart": true
    },
    {
      "code": "mealsOnBoard",
      "category": "menu",
      "order": 3,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": false,
      "showInFareIncluded": false,
      "showForAdult": true,
      "showForInfant": true,
      "showInBookingCart": true
    },
    {
      "code": "upTo10bonuses",
      "category": "bonus",
      "order": 4,
      "showInPopup": false,
      "showInSummary": false,
      "showInUpsell": false,
      "showInFareIncluded": false,
      "showForAdult": false,
      "showForInfant": false,
      "showInBookingCart": false
    },
    {
      "code": "handLuggage5kg",
      "category": "handBag",
      "order": 2,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": false,
      "showInFareIncluded": true,
      "showForAdult": true,
      "showForInfant": true,

```

```

    "showInBookingCart": true
  },
  {
    "code": "oneSuitcase23kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": true,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": false,
    "showInBookingCart": true
  },
  {
    "code": "oneSuitcase10kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": false,
    "showForInfant": true,
    "showInBookingCart": false
  },
  {
    "code": "canPayByBonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": false,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": false,
    "showInBookingCart": true
  },
  {
    "code": "upTo5bonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true,
    "showInBookingCart": true
  },
  {
    "code": "noMealsOnBoard",
    "category": "menu",
    "order": 3,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,

```

```

        "showInFareIncluded": false,
        "showForAdult": true,
        "showForInfant": true,
        "showInBookingCart": true
    }
],
"order": 250,
"commercialFamily": "U6ECONOMY",
"canRefund": true,
"hasUpsale": true,
"isLowCost": false
},
"appliedDiscounts": [
    {
        "flightReference": "1",
        "amount": 1214,
        "id": "1-4N9MCYL",
        "type": "DISCOUNT",
        "passengerReference": "2",
        "serviceCode": "",
        "usageType": 1,
        "promotionType": 0
    },
    {
        "flightReference": "2",
        "amount": 1214,
        "id": "1-4N9MCYL",
        "type": "DISCOUNT",
        "passengerReference": "2",
        "serviceCode": "",
        "usageType": 1,
        "promotionType": 0
    }
],
"baseFarePrices": [
    {
        "currency": "RUB",
        "passengerReference": "2",
        "isInfantFare": false,
        "totalPrice": 4859.0,
        "basePrice": 661.0,
        "discountAmount": 1214.0,
        "priceHidden": false
    }
],
"spentBonusesByPassengers": [],
"appliedCertificates": [],
"originOfficeId": "SVXU608AZ",
"originOfficeType": 2,
"isPayForbiddenForAgency": false,
"foodOnBoards": [],
"reservationRetrieveStatus": 0,
"engineType": 1,
"isGroupReservation": false,
"journey": {
    "outboundFlights": [
        {

```



```

    "aircraft": "Airbus A319",
    "aircraftIataCode": "319",
    "referenceNumber": "1",
    "origin": "LED",
    "destination": "DME",
    "departureDate": "2021-08-18T17:45:00",
    "arrivalDateUtc": "2021-08-18T16:15:00Z",
    "departureDateUtc": "2021-08-18T14:45:00Z",
    "arrivalDate": "2021-08-18T19:15:00",
    "flightNumber": "80",
    "classOfService": "A",
    "operatingCarrier": "U6",
    "marketingCarrier": "U6",
    "flightDuration": "1:30",
    "techStops": [],
    "departureTerminal": "1",
    "hash": "08/18/2021 17:45:00-80-LED-DME"
  }
],
"returnFlights": [
  {
    "aircraft": "Airbus A319",
    "aircraftIataCode": "319",
    "referenceNumber": "2",
    "origin": "DME",
    "destination": "LED",
    "departureDate": "2021-08-25T10:10:00",
    "arrivalDateUtc": "2021-08-25T08:40:00Z",
    "departureDateUtc": "2021-08-25T07:10:00Z",
    "arrivalDate": "2021-08-25T11:40:00",
    "flightNumber": "79",
    "classOfService": "A",
    "operatingCarrier": "U6",
    "marketingCarrier": "U6",
    "flightDuration": "1:30",
    "techStops": [],
    "arrivalTerminal": "1",
    "hash": "08/25/2021 10:10:00-79-DME-LED"
  }
],
"outboundFareFamily": {
  "code": "ECOFLEXRU",
  "category": "ECOFLEX"
},
"returnFareFamily": {
  "code": "ECOFLEXRU",
  "category": "ECOFLEX"
},
"outboundDuration": "1:30",
"returnDuration": "1:30",
"origin": "LED",
"destination": "DME",
"departureDate": "2021-08-18T17:45:00",
"returnDate": "2021-08-25T10:10:00",
"outboundBaggageAllowances": [
  {
    "quantity": 1,

```

```

        "weight": 23,
        "passengerType": 0
    }
],
"returnBaggageAllowances": [
    {
        "quantity": 1,
        "weight": 23,
        "passengerType": 0
    }
],
"containsInternationalFlights": false
},
"number": "JJANVQ",
"reservationDate": "2021-07-24T05:32:00",
"passengers": [
    {
        "id": 1,
        "hasInfant": false,
        "referenceNumber": "2",
        "firstName": "TEST",
        "middleName": "",
        "surname": "TESTOV",
        "passengerType": 0,
        "birthDate": "2000-01-01T00:00:00",
        "sex": 0,
        "nationality": "RUS",
        "documentNumber": "1234567890",
        "documentExpiryDate": "2031-07-24T00:00:00",
        "email": "TEST@TEST.TEST",
        "phone": "71234567890",
        "documentType": 0,
        "isAdult": true,
        "isChanged": false,
        "checkedIn": {
            "1": false,
            "2": false
        },
        "checkInHasStarted": {
            "1": false,
            "2": false
        }
    }
],
"services": [
    {
        "code": "BGML",
        "status": "HK",
        "category": "meal",
        "passengerReference": "2",
        "flightReference": "1",
        "referenceNumber": "15",
        "ticketNumber": "2624552537104",
        "finalPrice": 450.0,
        "price": 450.0,
        "discountAmount": 0.0
    }
],

```

```

{
  "code": "BGML",
  "status": "HK",
  "category": "meal",
  "passengerReference": "2",
  "flightReference": "2",
  "referenceNumber": "16",
  "ticketNumber": "2624552537106",
  "finalPrice": 550.0,
  "price": 550.0,
  "discountAmount": 0.0
},
{
  "code": "ABAG",
  "status": "HK",
  "category": "luggage",
  "passengerReference": "2",
  "flightReference": "1",
  "referenceNumber": "17",
  "ticketNumber": "2624552537107",
  "finalPrice": 2200.0,
  "price": 2200.0,
  "discountAmount": 0.0
},
{
  "code": "ABAG",
  "status": "HK",
  "category": "luggage",
  "passengerReference": "2",
  "flightReference": "2",
  "referenceNumber": "18",
  "ticketNumber": "2624552537108",
  "finalPrice": 2200.0,
  "price": 2200.0,
  "discountAmount": 0.0
}
],
"seats": [
  {
    "number": 5,
    "title": "A",
    "code": "RQST",
    "status": "HK",
    "passengerReference": "2",
    "flightReference": "1",
    "referenceNumber": "19",
    "ticketNumber": "2624552537103",
    "finalPrice": 300.0,
    "price": 300.0,
    "discountAmount": 0.0
  },
  {
    "number": 5,
    "title": "A",
    "code": "RQST",
    "status": "HK",
    "passengerReference": "2",

```

```

    "flightReference": "2",
    "referenceNumber": "22",
    "ticketNumber": "2624552537105",
    "finalPrice": 300.0,
    "price": 300.0,
    "discountAmount": 0.0
  }
],
"insurances": [],
"partnerProducts": [],
"hasSpecialConditions": false,
"isRefundAvailable": true,
"isAnnulAvailable": true,
"isAgentReservation": false
},
"errors": [],
"success": true
}

```

Catalog service, GET method. Getting information about services

Request string: /api/v1.1/catalogue

Response:

Services service, POST method. Adding services

Adding services:	
seat selection, insurance, luggage and pet travel for 2 passengers	
Request	Response

<pre> { "services": [{ "code": "PETC", "flightReference": "1", "passengerReference": "2", "text": "dog" }, { "code": "PETC", "flightReference": "2", "passengerReference": "2", "text": "dog" }, { "code": "BLOU", "flightReference": "1", "passengerReference": "2" }, { "code": "BLOU", "flightReference": "2", "passengerReference": "2" }, { "code": "BGML", "flightReference": "1", "passengerReference": "2" }, { "code": "BGML", "flightReference": "2", "passengerReference": "2" }, { "code": "ABAG", "flightReference": "1", "passengerReference": "2" }, { "code": "ABAG", "flightReference": "2", "passengerReference": "2" }, { "code": "XBLL", </pre>	<pre> { "data": { "journey": { "outboundFlights": [{ "aircraftIataCode": "32N", "referenceNumber": "1", "origin": "LED", "destination": "DME", "departureDate": "2021-09-01T16:00:00", "arrivalDate": "2021-09-01T17:30:00", "flightNumber": "60", "classOfService": "W", "operatingCarrier": "U6", "marketingCarrier": "U6", "techStops": [], "departureTerminal": "1", "hash": "09/01/2021 16:00:00-60-LED-DME" }], "returnFlights": [{ "aircraftIataCode": "32N", "referenceNumber": "2", "origin": "DME", "destination": "LED", "departureDate": "2021-09-02T13:35:00", "arrivalDate": "2021-09-02T14:55:00", "flightNumber": "59", "classOfService": "W", "operatingCarrier": "U6", "marketingCarrier": "U6", "techStops": [], "arrivalTerminal": "1", "hash": "09/02/2021 13:35:00-59-DME-LED" }], "outboundFareFamily": { "code": "ECOFLEXRU", "category": "ECOFLEX" }, "returnFareFamily": { "code": "ECOFLEXRU", "category": "ECOFLEX" }, "origin": "LED", </pre>
--	--

<pre> "flightReference": "1", "passengerReference": "2" }, { "code": "XBLL", "flightReference": "2", "passengerReference": "2" }], "seats": [{ "flightReference": "1", "passengerReference": "2", "number": 4, "title": "A" }, { "flightReference": "2", "passengerReference": "2", "number": 4, "title": "A" }] } </pre>	<pre> "destination": "DME", "departureDate": "2021-09-01T16:00:00", "returnDate": "2021-09-02T13:35:00", "outboundBaggageAllowances": [], "returnBaggageAllowances": [] }, "number": "JJARYP", "passengers": [{ "id": 1, "hasInfant": false, "referenceNumber": "2", "firstName": "TEST", "middleName": "", "surname": "TESTOV", "passengerType": 0, "birthDate": "2000-01-01T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "1234567890", "documentExpiryDate": "2031-07-24T00:00:00", "email": "TEST@TEST.TEST", "phone": "71234567890", "documentType": 0, "isAdult": true, "isChanged": false, "checkedIn": {}, "checkInHasStarted": {} }], "services": [{ "code": "PETC", "status": "HK", "text": "DOG 1PC 8KG 45X35X25", "petType": "DOG", "category": "pets", "passengerReference": "2", "flightReference": "1", "referenceNumber": "13", "finalPrice": 3500.0, "price": 3500.0, "discountAmount": 0.0 }, { "code": "PETC", </pre>
--	--

	<pre> "status": "HK", "text": "DOG 1PC 8KG 45X35X25", "petType": "DOG", "category": "pets", "passengerReference": "2", "flightReference": "2", "referenceNumber": "14", "finalPrice": 3500.0, "price": 3500.0, "discountAmount": 0.0 }, { "code": "BLOU", "status": "HK", "category": "room", "passengerReference": "2", "flightReference": "1", "referenceNumber": "15", "finalPrice": 3000.0, "price": 3000.0, "discountAmount": 0.0 }, { "code": "BLOU", "status": "HK", "category": "room", "passengerReference": "2", "flightReference": "2", "referenceNumber": "16", "finalPrice": 3500.0, "price": 3500.0, "discountAmount": 0.0 }, { "code": "BGML", "status": "HK", "category": "meal", "passengerReference": "2", "flightReference": "1", "referenceNumber": "17", "finalPrice": 450.0, "price": 450.0, "discountAmount": 0.0 }, { "code": "BGML", </pre>
--	--

	<pre> "status": "HK", "category": "meal", "passengerReference": "2", "flightReference": "2", "referenceNumber": "18", "finalPrice": 550.0, "price": 550.0, "discountAmount": 0.0 }, { "code": "ABAG", "status": "HK", "category": "luggage", "passengerReference": "2", "flightReference": "1", "referenceNumber": "19", "finalPrice": 2200.0, "price": 2200.0, "discountAmount": 0.0 }, { "code": "ABAG", "status": "HK", "category": "luggage", "passengerReference": "2", "flightReference": "2", "referenceNumber": "20", "finalPrice": 2200.0, "price": 2200.0, "discountAmount": 0.0 }, { "code": "XBLL", "status": "HK", "category": "handLuggage", "passengerReference": "2", "flightReference": "1", "referenceNumber": "21", "finalPrice": 2800.0, "price": 2800.0, "discountAmount": 0.0 }, { "code": "XBLL", "status": "HK", "category": "handLuggage", </pre>
--	--

	<pre> "passengerReference": "2", "flightReference": "2", "referenceNumber": "22", "finalPrice": 2800.0, "price": 2800.0, "discountAmount": 0.0 }], "seats": [{ "number": 4, "title": "A", "code": "RQST", "status": "HK", "passengerReference": "2", "flightReference": "1", "referenceNumber": "23", "finalPrice": 300.0, "price": 300.0, "discountAmount": 0.0 }, { "number": 4, "title": "A", "code": "RQST", "status": "HK", "passengerReference": "2", "flightReference": "2", "referenceNumber": "26", "finalPrice": 300.0, "price": 300.0, "discountAmount": 0.0 }], "partnerProducts": [], "hasSpecialConditions": true }, "errors": [], "success": true } </pre>
--	--

Seatmap service, GET method. Airplane scheme

Request	Response
----------------	-----------------

<pre> /api/v2.3/seatmap? referenceNumber=1& origin=LED& destination=DME& departureDate=2021-09-01& flightNumber=60& classOfService=W& commercialFamily=U6ECONOMY </pre>	<pre> { "data": { "compartmentDetails": [{ "number": 3, "seats": [{ "seat": "A", "number": 3, "description": "W", "characteristics": ["CH", "L", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 3, "description": "9", "characteristics": ["CH", "I", "L"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }] }] }] } } </pre>
---	---

	<pre> "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "L"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "L"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 3, "description": "9", </pre>
--	--

	<pre> "characteristics": ["CH", "I", "L"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 3, "description": "W", "characteristics": ["CH", "L", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 4, "seats": [{ "seat": "A", "number": 4, </pre>
--	--

	<pre> "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 4, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O" </pre>
--	---

	<pre>], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 4, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST",</pre>
--	--

	<pre> "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 4, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 5, "seats": [{ "seat": "A", "number": 5, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ </pre>
--	--

	<pre> "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 5, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }] } </pre>
--	---

	<pre> }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 5, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { </pre>
--	--

	<pre> "seat": "F", "number": 5, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 6, "seats": [{ "seat": "A", "number": 6, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, </pre>
--	--

	<pre> { "seat": "B", "number": 6, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", </pre>
--	--

	<pre> "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 6, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 6, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", </pre>
--	--

	<pre> "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }], }, { "number": 7, "seats": [{ "seat": "A", "number": 7, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 7, "description": "9", "characteristics": ["CH", "I", "O"], }, </pre>
--	---

	<pre> "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, </pre>
--	---

	<pre> "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 7, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 7, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }] }, </pre>
--	--

	<pre> "classDesignator": "ECONOMY" }] }, { "number": 8, "seats": [{ "seat": "A", "number": 8, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 8, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] }] } </pre>
--	---

	<pre>], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 8, </pre>
--	---

	<pre> "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 8, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 9, "seats": [{ "seat": "A", </pre>
--	---

	<pre> "number": 9, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 9, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", </pre>
--	--

	<pre> "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 9, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ </pre>
--	--

	<pre> "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 9, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 10, "seats": [{ "seat": "A", "number": 10, "description": "W", "characteristics": ["1D", "CH", "V", "W"], "occupation": "F", </pre>
--	---

	<pre> "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 10, "description": "9", "characteristics": ["1D", "CH", "I", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["1D", "A", "CH", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", </pre>
--	---

	<pre> "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["1D", "A", "CH", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 10, "description": "9", "characteristics": ["1D", "CH", "I", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, </pre>
--	---

	<pre> "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 10, "description": "W", "characteristics": ["1D", "CH", "V", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }], }, { "number": 11, "seats": [{ "seat": "A", "number": 11, "description": "W", "characteristics": ["1A", "1D", "CH", "E", "IE", "L", "V", "W"], }], </pre>
--	--

	<pre> "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 11, "description": "9", "characteristics": ["1A", "1D", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["1A", "1D", "A", "CH", "E", </pre>
--	---

	<pre> "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["1A", "1D", "A", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 11, "description": "9", "characteristics": [</pre>
--	--

	<pre> "1A", "1D", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 11, "description": "W", "characteristics": ["1A", "1D", "CH", "E", "IE", "L", "V", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" } </pre>
--	--

	<pre>] }, { "number": 12, "seats": [{ "seat": "A", "number": 12, "description": "W", "characteristics": ["1A", "CH", "E", "IE", "L", "V", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 12, "description": "9", "characteristics": ["1A", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST",</pre>
--	---

	<pre> "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["1A", "A", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["1A", "A", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [</pre>
--	--

	<pre> { "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 12, "description": "9", "characteristics": ["1A", "CH", "E", "IE", "L", "V"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 12, "description": "W", "characteristics": ["1A", "CH", "E", "IE", "L", "V", "W" </pre>
--	--

	<pre>], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 1200.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 13, "seats": [{ "seat": "A", "number": 13, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 13, "description": "9", "characteristics": ["CH", "I", </pre>
--	--

	<pre> "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", </pre>
--	---

	<pre>"uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 13, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 13, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false</pre>
--	--

	<pre> }], "classDesignator": "ECONOMY" }] }, { "number": 14, "seats": [{ "seat": "A", "number": 14, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 14, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, </pre>
--	--

	<pre> "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, {</pre>
--	--

	<pre> "seat": "E", "number": 14, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 14, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 15, "seats": [</pre>
--	---

	<pre> { "seat": "A", "number": 15, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 15, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": [</pre>
--	--

	<pre> "A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH", "O"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 15, "description": "9", "characteristics": ["CH", "I", "O"], "occupation": "F", </pre>
--	---

	<pre> "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 15, "description": "W", "characteristics": ["CH", "O", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 300.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 16, "seats": [{ "seat": "A", "number": 16, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", </pre>
--	--

	<pre> "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 16, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] } </pre>
--	---

	<pre> }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 16, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 16, </pre>
--	--

	<pre> "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 17, "seats": [{ "seat": "A", "number": 17, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }], "seat": "B", "number": 17, "description": "9", </pre>
--	---

	<pre> "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", </pre>
--	--

	<pre> "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 17, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 17, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], },</pre>
--	--

	<pre> "classDesignator": "ECONOMY" }] }, { "number": 18, "seats": [{ "seat": "A", "number": 18, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 18, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" </pre>
--	--

	<pre> }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 18, "description": "9", "characteristics": ["CH", "I" </pre>
--	---

	<pre>], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 18, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 19, "seats": [{ "seat": "A", "number": 19, "description": "W", "characteristics": ["CH", "W"], }], </pre>
--	---

	<pre> "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 19, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, </pre>
--	---

	<pre> "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 19, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F",</pre>
--	---

	<pre> "number": 19, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 20, "seats": [{ "seat": "A", "number": 20, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 20, </pre>
--	---

	<pre> "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ </pre>
--	--

	<pre> "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 20, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 20, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] } </pre>
--	--

	<pre>], "classDesignator": "ECONOMY" }] }, { "number": 21, "seats": [{ "seat": "A", "number": 21, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 21, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] }], </pre>
--	--

	<pre> "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 21, "description": "9", "characteristics": ["CH", </pre>
--	--

	<pre> "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 21, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 22, "seats": [{ "seat": "A", "number": 22, "description": "W", "characteristics": ["CH", "W"] }] } </pre>
--	---

	<pre>], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 22, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, </pre>
--	--

	<pre> "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 22, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { </pre>
--	--

	<pre> "seat": "F", "number": 22, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 23, "seats": [{ "seat": "A", "number": 23, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", </pre>
--	--

	<pre> "number": 23, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [</pre>
--	--

	<pre> { "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 23, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 23, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] } </pre>
--	--

	<pre> }], "classDesignator": "ECONOMY" }] }, { "number": 24, "seats": [{ "seat": "A", "number": 24, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 24, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] }] }</pre>
--	---

	<pre>], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 24, "description": "9", "characteristics": [</pre>
--	---

	<pre> "CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 24, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 25, "seats": [{ "seat": "A", "number": 25, "description": "W", "characteristics": ["CH", </pre>
--	---

	<pre> "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 25, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", </pre>
--	--

	<pre> "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 25, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, </pre>
--	--

	<pre> { "seat": "F", "number": 25, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }], }, { "number": 26, "seats": [{ "seat": "A", "number": 26, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }], } </pre>
--	--

	<pre> "seat": "B", "number": 26, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", </pre>
--	--

	<pre> "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 26, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 26, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, </pre>
--	--

	<pre> "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 27, "seats": [{ "seat": "A", "number": 27, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 27, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }] }] } </pre>
--	--

	<pre> }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 27, "description": "9", </pre>
--	--

	<pre> "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 27, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 28, "seats": [{ "seat": "A", "number": 28, "description": "W", "characteristics": [</pre>
--	--

	<pre> "CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 28, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", </pre>
--	---

	<pre> "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 28, "description": "9", "characteristics": ["CH", "I"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" </pre>
--	--

	<pre> }, { "seat": "F", "number": 28, "description": "W", "characteristics": ["CH", "W"], "occupation": "F", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 29, "seats": [{ "seat": "A", "number": 29, "description": "W", "characteristics": ["CH", "W"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }], }</pre>
--	--

	<pre> { "seat": "B", "number": 29, "description": "9", "characteristics": ["CH", "I"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["A", "CH"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["A", "CH"], </pre>
--	---

	<pre> "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 29, "description": "9", "characteristics": ["CH", "I"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 29, "description": "W", "characteristics": ["CH", "W"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, </pre>
--	---

	<pre> "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }, { "number": 30, "seats": [{ "seat": "A", "number": 30, "description": "W", "characteristics": ["1D", "7A", "CH", "V", "W"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "B", "number": 30, "description": "9", "characteristics": ["1D", "7A", "CH", "I", "V"], "occupation": "O", </pre>
--	---

	<pre> "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["1D", "7A", "A", "CH", "V"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["1D", "7A", "A", "CH", "V"], "occupation": "O", "passengerRelatedPrice": [{ </pre>
--	---

	<pre> "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "E", "number": 30, "description": "9", "characteristics": ["1D", "7A", "CH", "I", "V"], "occupation": "O", "passengerRelatedPrice": [{ "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }, { "seat": "F", "number": 30, "description": "W", "characteristics": ["1D", "7A", "CH", "V", "W"], "occupation": "O", "passengerRelatedPrice": [{ </pre>
--	--

	<pre> "name": "TEST", "uniqueCustomerIdentifier": "2", "seatPrice": 100.0, "isSeatUnavailable": false, "hasInfant": false }], "classDesignator": "ECONOMY" }] }], "isEmpty": false, "passengersTotalData": [{ "passengerReference": "2", "passengerPriceLevels": [{ "price": 100.0, "priceLevel": 1 }, { "price": 300.0, "priceLevel": 2 }, { "price": 1200.0, "priceLevel": 3 }] }] }] }, "errors": [], "success": true } </pre>
--	---

Service user, POST method. CRM authorization

Request: /api/v2.3/user

Request body: {"login":"1000761813","password":"tes756"}

Response:

```

{
  "data": {
    "userId": "1-4NGESOI",
    "firstName": "Александр",
    "firstNameLat": "Aleksandr",
    "middleName": "",
    "middleNameLat": "",
    "lastName": "Тестов",
    "lastNameLat": "Testov",
    "birthDate": "1989-10-20T00:00:00",
    "email": "test.test@test.test.ru",
    "cellPhone": "79996664455",
    "homePhone": "",
    "address": {
      "city": "Санкт-Петербург",
      "country": "Россия",
      "region": "-",
      "street": "-",
      "zip": "-"
    },
    "documents": [
      {
        "number": "4446665551",
        "title": "Общегражданский Паспорт",
        "lastUpdateDate": "2020-10-08T16:24:08",
        "documentType": 0,
        "countryCode": "RUS",
        "countryName": "Российская Федерация",
        "docId": "1-4NGESON"
      },
      {
        "number": "456321987",
        "title": "Общеграждан. Загран. Паспорт",
        "lastUpdateDate": "2020-10-08T16:24:08",
        "documentType": 1,
        "countryCode": "RUS",
        "countryName": "Российская Федерация",
        "docId": "1-4NGESOO"
      }
    ],
    "logins": [
      {
        "loginName": "test.test@test.test.ru",
        "loginType": "Электронная почта"
      },
      {
        "loginName": "79996664455",
        "loginType": "Телефон"
      }
    ],
    "loyalties": [
      {
        "memberId": "1-4NGESOR",
        "level": "Синий",
        "program": "Крылья",
        "programArt": "WingsProgramArticul",
        "joinDate": "2020-10-08T16:24:08",

```

```

"currentPoints": 500,
"levelPoints": 0,
"levelBonus": 0,
"levelStartDate": "08.10.2020",
"lastFlightDate": "",
"cards": [
  {
    "active": true,
    "ffn": "1000761812",
    "name": "Aleksandr Testov",
    "cardLevel": "Синий",
    "cardType": "Обычный",
    "issueDate": "2020-10-07T19:00:00",
    "validFromDate": "2020-10-07T19:00:00",
    "validToDate": "2120-10-08T16:24:09"
  }
],
"activeCard": {
  "active": true,
  "ffn": "1000761812",
  "name": "Aleksandr Testov",
  "cardLevel": "Синий",
  "cardType": "Обычный",
  "issueDate": "2020-10-07T19:00:00",
  "validFromDate": "2020-10-07T19:00:00",
  "validToDate": "2120-10-08T16:24:09"
}
},
"sex": 0,
"wingsLoyalty": {
  "memberId": "1-4NGESOR",
  "level": "Синий",
  "program": "Крылья",
  "programArt": "WingsProgramArticul",
  "joinDate": "2020-10-08T16:24:08",
  "currentPoints": 500,
  "levelPoints": 0,
  "levelBonus": 0,
  "levelStartDate": "08.10.2020",
  "lastFlightDate": "",
  "cards": [
    {
      "active": true,
      "ffn": "1000761812",
      "name": "Aleksandr Testov",
      "cardLevel": "Синий",
      "cardType": "Обычный",
      "issueDate": "2020-10-07T19:00:00",
      "validFromDate": "2020-10-07T19:00:00",
      "validToDate": "2120-10-08T16:24:09"
    }
  ]
},
"activeCard": {
  "active": true,
  "ffn": "1000761812",
  "name": "Aleksandr Testov",

```

```

        "cardLevel": "Синий",
        "cardType": "Обычный",
        "issueDate": "2020-10-07T19:00:00",
        "validFromDate": "2020-10-07T19:00:00",
        "validToDate": "2120-10-08T16:24:09"
    }
}
},
"errors": [],
"success": true
}

```

Promotions service, POST method

Flight promo code

Response /api/v2.3/promotions

```

{
  "number": "happyf",
  "origin": "LED",
  "destination": "MOW",
  "departureDate": "2021-07-24",
  "returnDate": null,
  "passengersCount": 1
}

```

Response:

```

{
  "data": {
    "number": "HAPPYF",
    "amount": 100,
    "promotionStatus": 0,
    "promotionType": 0,
    "description": "Скидка на перелет",
    "startFlightDate": "2020-01-01T05:00:00",
    "endFlightDate": "2025-08-01T05:00:00",
    "startDate": "2020-01-01T05:00:00",
    "endDate": "2025-11-30T05:00:00",
    "isUnique": false,
    "campaignId": "1-48R824H",
    "directions": [
      {
        "origin": "LED",
        "destination": "ZIA"
      },
      {
        "origin": "LED",
        "destination": "DME"
      }
    ]
  }
}

```

```

    {
      "origin": "LED",
      "destination": "SVO"
    },
    {
      "origin": "LED",
      "destination": "VKO"
    }
  ],
  "services": [],
  "families": [
    {
      "code": "COMFORT"
    },
    {
      "code": "ECOLIGHT"
    },
    {
      "code": "ECOFLEX"
    },
    {
      "code": "ECOPREM"
    },
    {
      "code": "BUSLIGHT"
    },
    {
      "code": "BUSFLEX"
    },
    {
      "code": "PRLIGHTIN"
    }
  ],
  "hasConflict": false,
  "excludedFlights": [
    {
      "flightNumber": "60",
      "departureAirport": "LED",
      "arrivalAirport": "DME"
    }
  ],
  "excludedPeriods": [
    {
      "startDate": "2021-02-09T00:00:00",
      "endDate": "2021-02-17T00:00:00"
    },
    {
      "startDate": "2020-12-02T00:00:00",
      "endDate": "2020-12-03T00:00:00"
    }
  ],
  "loyaltyCardIsRequired": false
},
"errors": [],
"success": true
}

```

Promo code for additional services

Request /api/v2.3/promotions

```
{
  "number": "happys",
  "origin": "LED",
  "destination": "MOW",
  "departureDate": "2021-07-24",
  "returnDate": null,
  "passengersCount": 1
}
```

Response:

```
{
  "data": {
    "number": "HAPPYS",
    "amount": 10,
    "promotionStatus": 0,
    "promotionType": 1,
    "description": "Дарим Вам скидку на День Рождения",
    "startFlightDate": "2020-12-23T05:00:00",
    "endFlightDate": "2022-02-01T05:00:00",
    "startDate": "2020-02-02T05:00:00",
    "endDate": "2022-10-02T05:00:00",
    "isUnique": false,
    "campaignId": "1-48R824E",
    "directions": [
      {
        "origin": "LED",
        "destination": "ZIA"
      },
      {
        "origin": "LED",
        "destination": "DME"
      },
      {
        "origin": "LED",
        "destination": "SVO"
      },
      {
        "origin": "LED",
        "destination": "VKO"
      }
    ],
    "services": [
      {
        "code": "BBAG"
      },
      {
        "code": "HTML"
      }
    ]
  }
}
```

```

    },
    {
      "code": "ABAG"
    },
    {
      "code": "SEAT"
    },
    {
      "code": "PETC"
    },
    {
      "code": "SNML"
    }
  ],
  "families": [],
  "hasConflict": false,
  "excludedFlights": [
    {
      "flightNumber": "60",
      "departureAirport": "LED",
      "arrivalAirport": "DME"
    }
  ],
  "excludedPeriods": [],
  "loyaltyCardIsRequired": false
},
"errors": [],
"success": true
}

```

An example of a request for a seat map on an airplane without a pre-made reservation for 1 adult with an infant: POST api/SeatmapStandalone

Request:

```

{
  "passengers": [
    {
      "id": 2,
      "firstName": "Ivan",
      "surname": "Petrov",
      "passengerType": "0",
      "sex": "0",
      "birthDate": "1989-10-25",
      "hasInfant": true
    }
  ],
  "origin": "LED",
  "destination": "DME",
  "departureDate": "2020-06-22",
  "flightNumber": "60",
  "classOfService": "X",
  "commercialFamily": "U6ECONOMY"
}

```

Response:

```
{
  "data": {
    "compartmentDetails": [
      {
        "number": 3,
        "seats": [
          {
            "seat": "A",
            "number": 3,
            "description": "W",
            "characteristics": [
              "CH",
              "L",
              "W"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
              {
                "name": "IVAN",
                "type": "ADT",
                "uniqueCustomerIdentifier": "2",
                "seatPrice": 999.0,
                "isSeatUnavailable": true,
                "hasInfant": true
              }
            ],
            "classDesignator": "ECONOMY"
          },
          {
            "seat": "B",
            "number": 3,
            "description": "9",
            "characteristics": [
              "CH",
              "I",
              "L"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
              {
                "name": "IVAN",
                "type": "ADT",
                "uniqueCustomerIdentifier": "2",
                "seatPrice": 999.0,
                "isSeatUnavailable": false,
                "hasInfant": true
              }
            ],
            "classDesignator": "ECONOMY"
          },
          {
            "seat": "C",
            "description": "A",
            "characteristics": [
```



```

    "A",
    "CH",
    "L"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "A",
    "CH",
    "L"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 3,
  "description": "9",
  "characteristics": [
    "CH",
    "I",
    "L"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ]
}

```

```

    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "F",
    "number": 3,
    "description": "W",
    "characteristics": [
      "CH",
      "L",
      "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
],
},
{
  "number": 4,
  "seats": [
    {
      "seat": "A",
      "number": 4,
      "description": "W",
      "characteristics": [
        "C",
        "CH",
        "W"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 299.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 4,
      "description": "9",
      "characteristics": [

```

```

    "C",
    "CH",
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [
    "A",
    "C",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "A",
    "C",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
},

```

```

    "classDesignator": "ECONOMY"
  },
  {
    "seat": "E",
    "number": 4,
    "description": "9",
    "characteristics": [
      "C",
      "CH",
      "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": false,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "F",
    "number": 4,
    "description": "W",
    "characteristics": [
      "C",
      "CH",
      "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
],
},
{
  "number": 5,
  "seats": [
    {
      "seat": "A",
      "number": 5,
      "description": "W",
      "characteristics": [
        "CH",

```

```

    "W"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "B",
  "number": 5,
  "description": "9",
  "characteristics": [
    "CH",
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [
    "A",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{

```

```

"seat": "D",
"description": "A",
"characteristics": [
  "A",
  "CH"
],
"occupation": "F",
"passengerRelatedPrice": [
  {
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 5,
  "description": "9",
  "characteristics": [
    "CH",
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 5,
  "description": "W",
  "characteristics": [
    "CH",
    "W"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}
]
},
{
  "number": 6,
  "seats": [
    {
      "seat": "A",
      "number": 6,
      "description": "W",
      "characteristics": [
        "CH",
        "W"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 299.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 6,
      "description": "9",
      "characteristics": [
        "CH",
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 299.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "C",
      "description": "A",
      "characteristics": [
        "A",
        "CH"
      ]
    }
  ]
}

```

```

],
"occupation": "F",
"passengerRelatedPrice": [
  {
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "A",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 6,
  "description": "9",
  "characteristics": [
    "CH",
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",

```



```

    "number": 6,
    "description": "W",
    "characteristics": [
      "CH",
      "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
],
},
{
  "number": 7,
  "seats": [
    {
      "seat": "A",
      "number": 7,
      "description": "W",
      "characteristics": [
        "CH",
        "W"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 299.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 7,
      "description": "9",
      "characteristics": [
        "CH",
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",

```

```

        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "C",
    "description": "A",
    "characteristics": [
        "A",
        "CH"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "D",
    "description": "A",
    "characteristics": [
        "A",
        "CH"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 7,
    "description": "9",
    "characteristics": [
        "CH",
        "I"
    ],
},

```

```

"occupation": "F",
"passengerRelatedPrice": [
  {
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 7,
  "description": "W",
  "characteristics": [
    "CH",
    "W"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}
],
},
{
  "number": 8,
  "seats": [
    {
      "seat": "A",
      "number": 8,
      "description": "W",
      "characteristics": [
        "CH",
        "W"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 299.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ]
    }
  ]
}

```

```

    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "B",
    "number": 8,
    "description": "9",
    "characteristics": [
      "CH",
      "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": false,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "C",
    "description": "A",
    "characteristics": [
      "A",
      "CH"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "D",
    "description": "A",
    "characteristics": [
      "A",
      "CH"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 8,
    "description": "9",
    "characteristics": [
        "CH",
        "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "number": 8,
    "description": "W",
    "characteristics": [
        "CH",
        "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 9,
    "seats": [
        {
            "seat": "A",
            "number": 9,

```

```

"description": "W",
"characteristics": [
  "CH",
  "W"
],
"occupation": "F",
"passengerRelatedPrice": [
  {
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "B",
  "number": 9,
  "description": "9",
  "characteristics": [
    "CH",
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [
    "A",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],

```

```

    "classDesignator": "ECONOMY"
  },
  {
    "seat": "D",
    "description": "A",
    "characteristics": [
      "A",
      "CH"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "E",
    "number": 9,
    "description": "9",
    "characteristics": [
      "CH",
      "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": false,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "F",
    "number": 9,
    "description": "W",
    "characteristics": [
      "CH",
      "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 10,
    "seats": [
        {
            "seat": "A",
            "number": 10,
            "description": "W",
            "characteristics": [
                "1D",
                "W"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "B",
            "number": 10,
            "description": "9",
            "characteristics": [
                "1D",
                "I"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": false,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "C",
            "description": "A",
            "characteristics": [
                "1D",
                "A"
            ],
        },
    ],
}

```



```

"occupation": "F",
"passengerRelatedPrice": [
  {
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "1D",
    "A"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 10,
  "description": "9",
  "characteristics": [
    "1D",
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 10,
  "description": "W",
  "characteristics": [
    "1D",
    "W"
  ],
  "occupation": "F",

```

```

    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
]
},
{
  "number": 11,
  "seats": [
    {
      "seat": "A",
      "number": 11,
      "description": "W",
      "characteristics": [
        "1A",
        "1D",
        "CH",
        "E",
        "IE",
        "L",
        "V",
        "W"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 999.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 11,
      "description": "9",
      "characteristics": [
        "1A",
        "1D",
        "CH",
        "E",
        "IE",
        "L",
        "V"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [

```

```

    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [
    "1A",
    "1D",
    "A",
    "CH",
    "E",
    "IE",
    "L",
    "V"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "1A",
    "1D",
    "A",
    "CH",
    "E",
    "IE",
    "L",
    "V"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,

```

```

        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 11,
    "description": "9",
    "characteristics": [
        "1A",
        "1D",
        "CH",
        "E",
        "IE",
        "L",
        "V"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 999.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "number": 11,
    "description": "W",
    "characteristics": [
        "1A",
        "1D",
        "CH",
        "E",
        "IE",
        "L",
        "V",
        "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 999.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
},

```

```

    "classDesignator": "ECONOMY"
  }
]
},
{
  "number": 12,
  "seats": [
    {
      "seat": "A",
      "number": 12,
      "description": "W",
      "characteristics": [
        "1A",
        "CH",
        "E",
        "IE",
        "L",
        "V",
        "W"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 999.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 12,
      "description": "9",
      "characteristics": [
        "1A",
        "CH",
        "E",
        "IE",
        "L",
        "V"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "name": "IVAN",
          "type": "ADT",
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 999.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    }
  ]
}

```

```

},
{
  "seat": "C",
  "description": "A",
  "characteristics": [
    "1A",
    "A",
    "CH",
    "E",
    "IE",
    "L",
    "V"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "1A",
    "A",
    "CH",
    "E",
    "IE",
    "L",
    "V"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 12,
  "description": "9",
  "characteristics": [
    "1A",

```

```

    "CH",
    "E",
    "IE",
    "L",
    "V"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 12,
  "description": "W",
  "characteristics": [
    "1A",
    "CH",
    "E",
    "IE",
    "L",
    "V",
    "W"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}
],
},
{
  "number": 13,
  "seats": [
    {
      "seat": "A",
      "number": 13,
      "description": "W",
      "characteristics": [
        "CH",
        "W"
      ]
    }
  ]
}

```

```

],
"occupation": "F",
"passengerRelatedPrice": [
  {
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "B",
  "number": 13,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [
    "A",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "A",

```



```

    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 13,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 13,
  "description": "W",
  "characteristics": [
    "CH",
    "W"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}
]
},
{

```

```

"number": 14,
"seats": [
  {
    "seat": "A",
    "number": 14,
    "description": "W",
    "characteristics": [
      "CH",
      "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "B",
    "number": 14,
    "description": "9",
    "characteristics": [
      "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "C",
    "description": "A",
    "characteristics": [
      "A",
      "CH"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ]
  }
]

```

```

    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "A",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
},
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 14,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ]
},
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 14,
  "description": "W",
  "characteristics": [
    "CH",
    "W"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,

```

```

        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 15,
    "seats": [
        {
            "seat": "A",
            "number": 15,
            "description": "W",
            "characteristics": [
                "CH",
                "W"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "name": "IVAN",
                    "type": "ADT",
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 299.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "B",
            "number": 15,
            "description": "9",
            "characteristics": [
                "I"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": false,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "C",
            "description": "A",
            "characteristics": [
                "A",
                "CH"
            ],
        },
    ],
}

```

```

"occupation": "F",
"passengerRelatedPrice": [
  {
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [
    "A",
    "CH"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 15,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "number": 15,
  "description": "W",
  "characteristics": [
    "CH",

```

```

        "W"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
],
},
{
    "number": 16,
    "seats": [
        {
            "seat": "A",
            "description": "W",
            "characteristics": [],
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "B",
            "number": 16,
            "description": "9",
            "characteristics": [
                "I"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": false,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "C",
            "description": "A",
            "characteristics": [],
            "passengerRelatedPrice": [

```

```

        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 16,
    "description": "9",
    "characteristics": [
        "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},

```

```

{
  "number": 17,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 17,
      "description": "9",
      "characteristics": [
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "C",
      "description": "A",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "D",
      "description": "A",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,

```



```

        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 17,
    "description": "9",
    "characteristics": [
        "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 18,
    "seats": [
        {
            "seat": "A",
            "description": "W",
            "characteristics": [],
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        }
    ],
    "classDesignator": "ECONOMY"
},

```

```

{
  "seat": "B",
  "number": 18,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 18,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 19,
    "seats": [
        {
            "seat": "A",
            "description": "W",
            "characteristics": [],
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "B",
            "number": 19,
            "description": "9",
            "characteristics": [
                "I"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": false,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        }
    ]
}

```

```

},
{
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 19,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}
]
},
{
  "number": 20,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 20,
      "description": "9",
      "characteristics": [
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "C",
      "description": "A",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "D",

```

```

    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "E",
    "number": 20,
    "description": "9",
    "characteristics": [
      "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
],
},
{
  "number": 21,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,

```

```

        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "B",
    "number": 21,
    "description": "9",
    "characteristics": [
        "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 21,
    "description": "9",
    "characteristics": [

```

```

        "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 22,
    "seats": [
        {
            "seat": "A",
            "description": "W",
            "characteristics": [],
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "B",
            "number": 22,
            "description": "9",
            "characteristics": [
                "I"
            ],
            "occupation": "F",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",

```



```

        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
},
],
"classDesignator": "ECONOMY"
},
{
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
},
],
"classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 22,
    "description": "9",
    "characteristics": [
        "I"
    ]
},
],
"occupation": "F",
"passengerRelatedPrice": [
    {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
]
},
],
"classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],

```

```

    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
]
},
{
  "number": 23,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 23,
      "description": "9",
      "characteristics": [
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "C",
      "description": "A",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ]
    }
  ]
}

```

```

    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 23,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}
]
},
{
  "number": 24,
  "seats": [
    {
      "seat": "A",

```

```

"description": "W",
"characteristics": [],
"passengerRelatedPrice": [
  {
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
},
{
  "seat": "B",
  "number": 24,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}

```

```

    },
    {
      "seat": "E",
      "number": 24,
      "description": "9",
      "characteristics": [
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "F",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    }
  ]
},
{
  "number": 25,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 25,
      "description": "9",
      "characteristics": [

```

```

    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 25,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ]
},
],

```

```

    "classDesignator": "ECONOMY"
  },
  {
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
]
},
{
  "number": 26,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 26,
      "description": "9",
      "characteristics": [
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "C",
      "description": "A",
      "characteristics": [],

```

```

    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "E",
    "number": 26,
    "description": "9",
    "characteristics": [
      "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  },
  {
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
      {
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }
]

```



```

},
{
  "number": 27,
  "seats": [
    {
      "seat": "A",
      "description": "W",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "B",
      "number": 27,
      "description": "9",
      "characteristics": [
        "I"
      ],
      "occupation": "F",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": false,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "C",
      "description": "A",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    },
    {
      "seat": "D",
      "description": "A",
      "characteristics": [],
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 27,
    "description": "9",
    "characteristics": [
        "I"
    ],
    "occupation": "F",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 28,
    "seats": [
        {
            "seat": "A",
            "description": "W",
            "characteristics": [],
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ]
        }
    ],
    "classDesignator": "ECONOMY"
}

```

```

},
{
  "seat": "B",
  "number": 28,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [
    {
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
},
{
  "seat": "E",
  "number": 28,
  "description": "9",
  "characteristics": [
    "I"
  ],
  "occupation": "F",
  "passengerRelatedPrice": [
    {

```

```

        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
},
{
    "number": 29,
    "seats": [
        {
            "seat": "A",
            "number": 29,
            "description": "W",
            "characteristics": [
                "W"
            ],
            "occupation": "O",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,
                    "isSeatUnavailable": true,
                    "hasInfant": true
                }
            ],
            "classDesignator": "ECONOMY"
        },
        {
            "seat": "B",
            "number": 29,
            "description": "9",
            "characteristics": [
                "I"
            ],
            "occupation": "O",
            "passengerRelatedPrice": [
                {
                    "uniqueCustomerIdentifier": "2",
                    "seatPrice": 0.0,

```

```

        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "C",
    "description": "A",
    "characteristics": [
        "A"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "D",
    "description": "A",
    "characteristics": [
        "A"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 29,
    "description": "9",
    "characteristics": [
        "I"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}

```

```

    },
    {
      "seat": "F",
      "number": 29,
      "description": "W",
      "characteristics": [
        "W"
      ],
      "occupation": "O",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    }
  ]
},
{
  "number": 30,
  "seats": [
    {
      "seat": "A",
      "number": 30,
      "description": "W",
      "characteristics": [
        "1D",
        "7A",
        "W"
      ],
      "occupation": "O",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",
          "seatPrice": 0.0,
          "isSeatUnavailable": true,
          "hasInfant": true
        }
      ],
      "classDesignator": "ECONOMY"
    }
  ],
  "seats": [
    {
      "seat": "B",
      "number": 30,
      "description": "9",
      "characteristics": [
        "1D",
        "7A",
        "I"
      ],
      "occupation": "O",
      "passengerRelatedPrice": [
        {
          "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "C",
    "description": "A",
    "characteristics": [
        "1D",
        "7A",
        "A"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "D",
    "description": "A",
    "characteristics": [
        "1D",
        "7A",
        "A"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
},
{
    "seat": "E",
    "number": 30,
    "description": "9",
    "characteristics": [
        "1D",
        "7A",
        "I"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {

```

```

        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
},
{
    "seat": "F",
    "number": 30,
    "description": "W",
    "characteristics": [
        "1D",
        "7A",
        "W"
    ],
    "occupation": "O",
    "passengerRelatedPrice": [
        {
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}
]
}
],
"isEmpty": false,
"passengersTotalData": [
    {
        "passengerReference": "2",
        "passengerPriceLevels": [
            {
                "price": 299.0,
                "priceLevel": 1
            },
            {
                "price": 999.0,
                "priceLevel": 2
            }
        ]
    }
]
}
],
"errors": [],
"success": true
}

```

An example of a request for a service catalog without a pre-made reservation for 1 adult with a child and an infant: POST api/CatalogStandalone

Request:


```

{
  "flights": {
    "outboundFlights": [
      {
        "origin": "LED",
        "destination": "DME",
        "departureDate": "2020-07-03T16:05:00",
        "arrivalDate": "2020-07-03T17:30:00",
        "flightNumber": "60",
        "classOfService": "P",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightIndex": 1
      }
    ]
  },
  "fares": {
    "outboundFareFamily": {
      "code": "ECOPREMURU",
      "category": "ECOPREM"
    },
    "fares": [
      {
        "passengerType": 0,
        "flightIndex": 1,
        "fareBasis": "PFLOW",
        "isBreakpoint": true
      },
      {
        "passengerType": 5,
        "flightIndex": 1,
        "fareBasis": "PFLOW",
        "isBreakpoint": true
      },
      {
        "passengerType": 4,
        "flightIndex": 1,
        "fareBasis": "PFLOW",
        "isBreakpoint": true
      }
    ]
  },
  "passengers": [
    {
      "id": 3,
      "firstName": "Sofia",
      "middleName": "Evgenievna",
      "surname": "Bodrova",
      "passengerType": 0,
      "birthDate": "1995-07-20",
      "sex": "Female",
      "hasInfant": true
    },
    {
      "id": 2,
      "firstName": "Rebenok",
      "surname": "Bodrova",

```

```

    "passengerType": 5,
    "birthDate": "2015-07-20",
    "sex": "Male"
  },
  {
    "id": 4,
    "firstName": "Mladenets",
    "surname": "Bodrova",
    "passengerType": 4,
    "birthDate": "2019-07-20",
    "sex": "Male"
  }
]
}

```

Response:

```

{
  "data": {
    "services": [
      {
        "code": "HBAG",
        "flightReferences": [
          {
            "referenceNumber": "1",
            "passengerPrices": [
              {
                "referenceNumber": "3",
                "amount": 2000.0
              },
              {
                "referenceNumber": "2",
                "amount": 2000.0
              },
              {
                "referenceNumber": "4",
                "amount": 2000.0
              }
            ]
          }
        ]
      }
    ],
    {
      "code": "BGML",
      "flightReferences": [
        {
          "referenceNumber": "1",
          "passengerPrices": [
            {
              "referenceNumber": "3",
              "amount": 500.0
            },
            {
              "referenceNumber": "2",
              "amount": 500.0
            }
          ]
        }
      ]
    }
  }
}

```

```

        {
            "referenceNumber": "4",
            "amount": 500.0
        }
    ]
}
],
{
    "code": "ABAG",
    "flightReferences": [
        {
            "referenceNumber": "1",
            "passengerPrices": [
                {
                    "referenceNumber": "3",
                    "amount": 2000.0
                },
                {
                    "referenceNumber": "2",
                    "amount": 2000.0
                },
                {
                    "referenceNumber": "4",
                    "amount": 2000.0
                }
            ]
        }
    ]
},
{
    "code": "SFML",
    "flightReferences": [
        {
            "referenceNumber": "1",
            "passengerPrices": [
                {
                    "referenceNumber": "3",
                    "amount": 600.0
                },
                {
                    "referenceNumber": "2",
                    "amount": 600.0
                },
                {
                    "referenceNumber": "4",
                    "amount": 600.0
                }
            ]
        }
    ]
},
{
    "code": "CHML",
    "flightReferences": [
        {
            "referenceNumber": "1",

```

```

    "passengerPrices": [
      {
        "referenceNumber": "3",
        "amount": 600.0
      },
      {
        "referenceNumber": "2",
        "amount": 600.0
      },
      {
        "referenceNumber": "4",
        "amount": 600.0
      }
    ]
  }
]
},
{
  "code": "HTML",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": [
        {
          "referenceNumber": "3",
          "amount": 600.0
        },
        {
          "referenceNumber": "2",
          "amount": 600.0
        },
        {
          "referenceNumber": "4",
          "amount": 600.0
        }
      ]
    }
  ]
}
],
{
  "code": "CAML",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": [
        {
          "referenceNumber": "3",
          "amount": 500.0
        },
        {
          "referenceNumber": "2",
          "amount": 500.0
        },
        {
          "referenceNumber": "4",
          "amount": 500.0
        }
      ]
    }
  ]
}
]
}

```

```

    ]
  }
]
},
{
  "code": "CBAG",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": [
        {
          "referenceNumber": "3",
          "amount": 2000.0
        },
        {
          "referenceNumber": "2",
          "amount": 2000.0
        },
        {
          "referenceNumber": "4",
          "amount": 2000.0
        }
      ]
    }
  ]
}
]
},
{
  "code": "PETC",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": []
    }
  ]
}
],
{
  "code": "SNML",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": [
        {
          "referenceNumber": "3",
          "amount": 350.0
        },
        {
          "referenceNumber": "2",
          "amount": 350.0
        },
        {
          "referenceNumber": "4",
          "amount": 350.0
        }
      ]
    }
  ]
}
]
},

```

```

{
  "code": "VGML",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": [
        {
          "referenceNumber": "3",
          "amount": 600.0
        },
        {
          "referenceNumber": "2",
          "amount": 600.0
        },
        {
          "referenceNumber": "4",
          "amount": 600.0
        }
      ]
    }
  ]
},
{
  "code": "BBAG",
  "flightReferences": [
    {
      "referenceNumber": "1",
      "passengerPrices": [
        {
          "referenceNumber": "3",
          "amount": 2000.0
        },
        {
          "referenceNumber": "2",
          "amount": 2000.0
        },
        {
          "referenceNumber": "4",
          "amount": 2000.0
        }
      ]
    }
  ]
},
"packs": [],
"seatServices": [
  {
    "seatCharacteristics": [],
    "code": "SEAT",
    "flightReferences": [
      {
        "referenceNumber": "1",
        "passengerPrices": [
          {
            "referenceNumber": "3",
            "amount": 0.0
          }
        ]
      }
    ]
  }
]

```

```

        },
        {
            "referenceNumber": "2",
            "amount": 0.0
        },
        {
            "referenceNumber": "4",
            "amount": 0.0
        }
    ]
}
]
},
],
"petsServiceConfiguration": [
    {
        "petsList": [
            "cat",
            "dog",
            "canary",
            "parrot"
        ],
        "flightReferences": [
            "1"
        ]
    }
],
"portions": [
    {
        "type": "BTU",
        "number": "1",
        "flightReferenceNumbers": [
            "1"
        ]
    }
],
"serviceCategories": [
    {
        "code": "URFL",
        "category": "insurance",
        "order": 4,
        "fullDirection": false
    },
    {
        "code": "URTR",
        "category": "insurance",
        "order": 4,
        "fullDirection": false
    },
    {
        "code": "SEAT",
        "category": "seat",
        "order": 1,
        "fullDirection": false
    },
    {
        "code": "ABAG",

```

```

    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "BBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "CBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "HBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "PETC",
    "category": "pets",
    "order": 4,
    "fullDirection": true
  },
  {
    "code": "CHML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "HTML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "MOML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "SFML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "SNML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }

```



```

},
{
  "code": "SPML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "VGML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "KSML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "CAML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "BGML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "AVML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "HNML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "VJML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "TCML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "GFML",

```

```

    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "BBML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "CPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "MPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "GRML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "CBBG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "RVML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "VLML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "VOML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "BIKE",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  }

```

```

},
{
  "code": "SKI",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "DBML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "BLML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
},
{
  "code": "EXST",
  "category": "seat",
  "order": 1,
  "fullDirection": false
},
{
  "code": "AVIH",
  "category": "pets",
  "order": 4,
  "fullDirection": true
},
{
  "code": "SVAN",
  "category": "pets",
  "order": 4,
  "fullDirection": true
},
{
  "code": "CHNM",
  "category": "data",
  "order": 6,
  "fullDirection": true
},
{
  "code": "NAMM",
  "category": "data",
  "order": 6,
  "fullDirection": true
},
{
  "code": "NAME",
  "category": "data",
  "order": 6,
  "fullDirection": true
},
{
  "code": "MEQT",

```

```

    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "GPST",
    "category": "seat",
    "order": 1,
    "fullDirection": false
  },
  {
    "code": "BULK",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "LCML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "WEAP",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "FPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "LFML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "NLML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "LSML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "NFML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }

```

```

},
{
  "code": "XBAG",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "HBGS",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "HBGT",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "GOLF",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "SPEQ",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "BUSS",
  "category": "other",
  "order": 7,
  "fullDirection": true
},
{
  "code": "DBGS",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "DBGT",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "XBLS",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
},
{
  "code": "TIME",

```

```

    "category": "other",
    "order": 7,
    "fullDirection": true
  },
  {
    "code": "FRAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "JPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  },
  {
    "code": "WBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "PFEE",
    "category": "data",
    "order": 6,
    "fullDirection": true
  },
  {
    "code": "SRXBLL",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "SRXBLS",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "XBLL",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  },
  {
    "code": "UMNR",
    "category": "other",
    "order": 7,
    "fullDirection": true
  },
  {
    "code": "PAKA",
    "category": "pack",
    "order": 0,
    "fullDirection": true
  }

```

```

    },
    {
      "code": "PAKB",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    },
    {
      "code": "PAKC",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    },
    {
      "code": "PAKD",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    },
    {
      "code": "PAKE",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    },
    {
      "code": "PAKF",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    },
    {
      "code": "PKDP",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    },
    {
      "code": "PKEP",
      "category": "pack",
      "order": 0,
      "fullDirection": true
    }
  ],
  "serviceOptions": [
    {
      "code": "HTML",
      "preselect": true,
      "flightReferences": [
        "1"
      ],
      "category": "meal",
      "priority": 1
    },
    {
      "code": "SNML",
      "preselect": true,

```

```
    "flightReferences": [
      "1"
    ],
    "category": "meal",
    "priority": 0
  },
  {
    "code": "URFL",
    "preselect": true,
    "flightReferences": [
      "1"
    ],
    "category": "insurance",
    "priority": 0
  }
],
"moneyWallCoefficient": 1.21
},
"errors": [],
"success": true
}
```