

EN_Internet Booking Engine (Sirena-Travel)

API documentation

Document version	v2.3.116 (Sirena) 2.3 - recommended API version 116 – sprint number where recent changes were introduced
Supported API versions	v2.2, v2.3

Table of contents

EN_Internet Booking Engine (Sirena-Travel)	1
API documentation	1
1_Change history	4
2_Glossary	24
3_General provision.....	25
4_General information about API methods	25
API Versions	25
HTTP(S)-headers v2.x	26
IBE API services query.....	26
IBE API services response codes.....	27
5_Booking flow description.....	27
General booking flow	27
Description of partners services upsale process after payment	28
6_API methods description.....	29
6_1_api/session: obtain session service	29
6_3_api/flights: obtain flights and flight conditions service.....	30
6_4_api/reservation: booking management service.....	68
6_6_api/services, api/catalogue: auxiliary services management service	129
6_7_api/CatalogueStandalone: service for obtaining a cataolog of serivces without PNR	150
6_8_api/seatmap: retrieve aircraft seatmap service	165
6_9_api/SeatmapStandalone: retrieve aircraft seatmap service without PNR.....	170
6_10_api/annul: seat annulling service	177
6_11_api/reservation/AddNewPassengers, AddAdditionalPassenger, RemoveAdditionalPassenger: service for changing passengers in already issued booking	181
6_12_api/partners: auxiliary partner's services management service	189
6_13_api/ticket/void: ticket void service	202
6_14_api/queue: services for working with queues (for agents)	204
6_16: api/exchange: service of ticket exchanging	214
6_18_api/InvoluntaryRefund/Moneyless: involuntary ticket refund service (for agents)..	246
6_19_api/refund: voluntary tickets and services refund service.....	253
6_24_api/reservation/AddContactInfo: contact info adding service (for agents).....	271
6_25_api/reservation/advanced: service for retrieving SSR and remarks information contained in reservation (for agents).....	273

6_26_api/reservation/AddSSR, api/reservation/AddRemark: services for adding SSR and remarks to reservation (for agents)	289
6_27_api/ticket/quota: service for retrieving information about ticket issuance quotas	293
6_32_api/SubsidiesConfirmationSession: service for getting the information about the status of creation of a subsidized reservation v2.3.....	295
6_33_api/passenger/AddFiles: service for submitting attachments when creating a subsidy application for Kaliningrad, the Far Eastern Federal District, and Pensioners/Youth.....	301
6_35_api/reservation/partners: obtain issued partner's services information service	303
6_36_api/reservation/ArchivedBookingOrders: service of information about archived booking	309
6_38_api/pricing: fares confirmation (pricing) service	312
7_Service API methods description.....	317
7_1_api/system/availability: service for monitoring the status of IBE.....	317
8_Attachments	319
8_1_Attachment: Dictionaries	319
8_2_Attachment: Main IBE error codes	340
8_3_Attachment: Partner's services.....	371
8_4_Attachment: service SSR	374

1_Change history

Date	Version	Changelog	Author
03.03.2026	2.3.116	<p>Updated:</p> <ul style="list-style-type: none"> ▪ dictionaries: additional services types, IBE API errors ▪ isVoidAvailable and voidTimeLimitUtc parameters have been added to the GET reservation and GET reservation/advanced method's responses ▪ GET api/checkin/seatmap method's description has been moved to api/checkin chapter ▪ api/payment/issueFree method's description has been removed (not actual) <p>Added:</p> <ul style="list-style-type: none"> ▪ ticketNumber parameter's description for the GET api/receipt request 	Trokhachev I.
30.12.2025	2.3.115	<p>Updated:</p> <ul style="list-style-type: none"> ▪ in request body of POST api/catalogueStandalone method relatedInfantWithSeats parameter has been changed to relatedInfantWithSeatId, added array sportEquipmentServiceConfiguration to response example ▪ outboundFlights and returnFlights arrays have been added to POST api/discounts request body ▪ POST refund/calculate, POST refund/partial: isExchangeSegments parameter has been added <p>Added:</p> <ul style="list-style-type: none"> ▪ POST api/exchange/confirmFree method description has been added ▪ POST api/payment/perform method description ▪ POST api/pricing method's description ▪ POST reservation/claim method's description ▪ POST api/user/external method's description ▪ GET api/weather method's description 	Trokhachev I.

17.11.2025	2.3.114	<p>Added:</p> <ul style="list-style-type: none"> • "isTrustedDevice" parameter has been added for POST api/user method • seat status "error during registration" in the chapter "Dictionaries" • reservation/AddRemark method's description has been added to reservation/AddSSR chapter • reservation/AddAdditionalPassenger and reservation/RemoveAdditionalPassenger method's description have been added to chapter 6_11_reservation/AddNewPassenger • reservation/PaymentStatus method's description has been added to chapter "Booking payment service" • chapter 6_35_api/reservation/partners - service of information about issued partner's services in reservation • chapter 6_36_api/reservation/ArchivedBookingOrders - service of information about archived booking • chapter 6_37_api/reservation/CancelSegments - service for removing the segment from reservation • request examples for POST-method descriptions with large request body • api/calendar/ScheduleByDay method description has been added to chapter api/calendar • new parameter "fareCode" in 6_9_api/SeatmapStandalone request - for clarify the commercial family <p>Changed:</p> <ul style="list-style-type: none"> • index numbers have been added to service names • endpoint names have been moved forward for easier navigation 	Trokhachev I.S.
------------	---------	---	-----------------

04.09.2025	2.3.113	<p>Added:</p> <ul style="list-style-type: none"> description of service 6_34_Promocode management service: api/promotions POST api/BoardingPass/EmailAddress method description <p>Changed:</p> <ul style="list-style-type: none"> IBE error codes - error descriptions have been extended description of business logic of voluntary and involuntary refunds has been updated api/seatmap and api/seatmap/checkin methods descriptions were merged into one section 	Trokhachev I.S.
29.07.2025	2.3.112	<p>Added:</p> <ul style="list-style-type: none"> description of new method GET api/BoardingPass/Info description of service method GET system/availability <p>Changed:</p> <ul style="list-style-type: none"> description of methods api/exchange (ticket exchange), api/Exchange/search (search for exchange recommendations) and api/exchange/cancel (exchange cancellation) are combined into section 6_16_Ticket exchange service number of service of getting information about quotas (ticket/quota) has been changed to 6_27 api/annul and api/annul/partial methods were merged in one partition - api/annul AddNewPassengers method's number has been changed from 6_35 to 6_11 description of method GetServiceCategoriesOrder has been moved into section of description of api/catalogue method 	Trokhachev I.S.

19.06.2025	2.3.111	<p>Added:</p> <ul style="list-style-type: none"> • description of request parameters of the GET api/partners method • description of required parameters in method's headers • list of IBE errors has been expanded <p>Updated:</p> <ul style="list-style-type: none"> • chapter "Attachments" has been updated • fixed errors and inaccuracies in the description of API methods 	Trokhachev I. S.
05.05.2025	2.3.110	<p>Added:</p> <ul style="list-style-type: none"> • currency parameter added in the response to the GET api/Reservation to display the actual currency used for payment. • New method Exchange/Cancel for cancelling the ticket exchange process • currency parameter added in the response to the GET api/flights and GET api/partners/[id] 	Lebedeva A. T.

<p>19.03.2025</p>	<p>2.3.109</p>	<p>Added:</p> <ul style="list-style-type: none"> • New GET api/Ticket/Quota method for retrieving up-to-date information on ticket issuance quotas. • New GET system/Statistics method for obtaining statistics on bookings, tickets, and services for a specified time period. • POST payment/IssueFree/Partial method for issuing EMD tickets for free services/tickets (e.g., for infants without seats on domestic flights). • New service type UPQE (upgrade service). The serviceCategories array is included in the response of the GET api/Catalogue v2.3 request. The services[] array is included in the response of the POST api/Services and GET api/Reservation requests. • passengerReferences parameter added to the POST /api/payment request array for paying for tickets/services for specific passengers. <p>Updated:</p> <ul style="list-style-type: none"> • The method for removing services has been replaced: POST /api/v2.3/Services/Remove is now POST /api/v2.3/Services/RemoveByReferences. • The PUT api/passenger method has been replaced with POST api/passenger. • The POST api/Passenger/AddFiles method is now used for submitting attachments to process subsidies for Pensioners/Youth. 	<p>Orlova A.S.</p>
-------------------	----------------	--	--------------------

<p>29.11.2024</p>	<p>2.3.108</p>	<p>Added:</p> <ul style="list-style-type: none"> • the subsidies_type parameter in the GET api/Flights request array to search for subsidized flight options. • the subsidiesType parameter in the response to the POST api/Reservation of the booking creation request to match the type of subsidy and the passenger category. • the isCheckinWithoutPseudo parameter in the GET api/Reservation response and in the POST api/Reservation response array for the possibility to work with the seatmap during open registration. • the isIssuedByRefundableOffice parameter in the response of the GET api/Reservation request the POST api/Reservation response array for processing services and tickets that cannot be refunded. • the isExchangePermitted parameter in the POST api/Reservation response for the possibility or prohibition of ticket exchange. • a new type of FST (FarEast) subsidies for the Far Eastern Federal District. • a new type of meals with SSR = CDML • error 401033 SirenaSoapPassengerAgentTransitBortChanging for processing and transmitting in the response of GET api/Checkin request to determine online registration arrangements. <p>Updated:</p> <ul style="list-style-type: none"> • the service for sending attachments together with an application to CRM to confirm the right to a subsidy of the Far Eastern Federal District (FarEast), POST api/Passenger/AddFiles. • using the GET api/SubsidiesConfirmationSession method to check the status of a subsidy application for the Far Eastern Federal District (FarEast) 	<p>Lebedeva A. T.</p>
-------------------	----------------	--	-----------------------

22.10.2024	2.3.107	<p>Added:</p> <ul style="list-style-type: none"> • a service to get flight recommendations from Yandex Webmaster by GET api/flight/meta/search request v2.3. <p>Changed:</p> <ul style="list-style-type: none"> • a process of realizing for Voluntary refund by POST api/Refund/Partial. In a case when the process of refunding a ticket and additional services gets an error, will be used automatically annulling seats by POST api/Annul from the booking before creating a request to CRM. • a process of realizing for Involuntary refund by POST api/InvoluntaryRefund. The process was added for automatically annulling seats by POST api/Annul from the booking before creating a request to CRM. 	Usolkin R.A.
04.09.2024	2.3.106	<p>Added:</p> <ul style="list-style-type: none"> • a new option type ISMS (SMS Notification Service). Appears in ServiceCategorie array in a response to GET api/Catalogue v2.3. Additionally ISMS is used in POST api/Services v2.3, GET api/Reservation in services [] array. • a relatedAdultReferenceNumber parameter to the array of the request POST api/Reservation, identifying the chosen adult (ADT) passenger to be linked to children (CHD/INS/INF) passengers in the reservation. <p>Changed:</p> <ul style="list-style-type: none"> • PUT api/Passenger method for updating passengers data with the possibility to change values in parameters: birthDate, sex, documentNumber, documentExpiryDate (for international flights). 	Lebedeva A. T.

31.07.2024	2.3.105	<p>Added:</p> <ul style="list-style-type: none"> • a certificateText parameter into an array in a request for creating a Certificate at POST api/Certificate, for sending a message, that will be added to an email notification with the Certificate. • a service of Sending attachments for a request to the CRM for getting an approval for using special Kaliningrad's subsidy, POST api/Passenger/AddFiles. • new Types of passengers for Kalilningrad's subsidy: STU (Student), CHR (Child). • new Types of passengers for subsidy by Disability: DCD, DIC, DAC, SBK. • a SubType parameter into services [] array for an option of Sport equipment transportation in responses by: GET api/Reservation; POST api/Services v2.3; POST api/Services/Add v2.3. • a SeatUnavailableReason array with parameters of a unavailability seat reason in a response by: GET api/Seatmap and POST api/SeatmapStandalone • a new option type as EQPT (Transportation of a Sport equipment). A SportEquipmentServiceConfiguration array in a response by GET api/Catalogue v2.3. Also it has been in responses by: POST api/Services v2.3, GET api/Reservation in services [] array. • a service for adding a new infant passenger (INF) to existed booking through POST api/Reservation/AddNewPassengers v2.3 • new type of options for weapon and ammunition transportation: PTWP (Weapon at the luggage, paid), FTWP (Weapon at the luggage, for free). At responses by requests: GET api/Catalogue; POST api/Services, GET api/Reservation in services [] arrays. <p>Changed:</p> <ul style="list-style-type: none"> • STR (adult) passenger type is re-used for getting flight recommendations and creating bookings via special Kaliningrad's subsidy. • PUT api/Passenger method for updating passengers data at: FirstName, Surname, MiddleName parameters. 	Usolkin R.A.
------------	---------	--	--------------

		<ul style="list-style-type: none"> • DisabilityDocumentNumber parameter is changed to SubsidyDocumentNumber at passengers[] array for requests GET, POST api/Reservation • using method GET api/SubsidiesConfirmationSession for checking a current state of request for Kaliningrad's subsidy 	
13.06.2024	2.3.104	<p>Added:</p> <ul style="list-style-type: none"> • the serviceOptionLabels array containing type and order of labels appearing at additional service cards received in GET /api/apiVersion/Catalogue method response; • the sortedInCategories array containing the sorting of services within the categories of additional services received in GET /api/apiVersion/Catalogue/GetServiceCategoriesOrder response; • the isEmailPayment parameter used as the tag for generating a payment link passed in the PUT /api/apiVersion/Passenger request; • a service of receiving subsidized tariff routes in GET /api/v2.3/Airports/SubsidiesDirections response. 	Lebedeva A.T.
	2.3.103	<p>Added:</p> <ul style="list-style-type: none"> • the searchIndirectFlights parameter indicates the search for flights to exchange with transfers in the POST api/exchange/search method request; • the relatedPnrNumbers parameter of the array of related bookings received as a result of a partial exchange in response to the api/reservation method; • in the Attachments: Partner's services new BUSINESSOW Aeroexpress fare "Business" and BUSINESSRT Aeroexpress fare "Business: Round-Trip". 	Gudumak S.E.

<p>20.03.2024</p>	<p>2.3.102</p>	<p>Added:</p> <ul style="list-style-type: none"> • the offsetUtc parameter of difference between local airport time and UTC time to the GET api/airports method response; • passengerAmounts array with information about the amount for the partner service in the context of the passenger to the response of GET api/partners/{Id} method; • the isBoardingPassAvailable parameter of the boarding pass availability for the passenger to the response of the GET api/Checkin method; • the boardingPassRestrictedBySegment object with information about the restrictions on obtaining a boarding pass by segment to the response of POST/GET api/Reservation v2.3 method; • the summToPayForExchange parameter is the amount to be paid for a voluntary exchange to the response of GET and POST api/Reservation v2.3 method; • in the Attachments: Partner's services new FAMILYRT Aeroexpress fare "Family: Round trip". 	<p>Gudumak S.E.</p>
-------------------	----------------	---	---------------------

24.01.2024	2.3.101	<p>Added:</p> <ul style="list-style-type: none"> • a new type of passengers mmi (Ministry of Defence) • a parameter as mmi at a request for GET flights • an array • the serviceCategoryLabels[] array at a response by GET Catalogue which is existed a list of Labels for appearing at additional option cards. • a parameter as searchPeriod at a request for POST Exchange/Search • a parameter as originalFlightReferenceNumber at the flights[] array at a response by POST Exchange/Search. • a service of adding a Loyalty program card number to issued booking by POST Reservation/AddLoyaltyCard v2.3 request. • a parameter as socialSecurityNumber to the passengers[] array for POST Reservation request. <p>Changed:</p> <ul style="list-style-type: none"> • a service of searching recommendations for voluntary exchanging flight segments at the booking. • at the Attachments: The dictionary of available meals; The dictionary of passenger types; The glossary of the partner services. • disabled an ability for creating a booking with subsidised fares via agents. POST Reservation. 	Usolkin R.A.
------------	---------	--	--------------

22.11.2023	2.3.100	<p>Added:</p> <ul style="list-style-type: none"> • accompanimentBookingInfo[] array in POST Reservation request • accompanimentBookingsInfo[] array in POST/GET Reservation(/Advanced) response • parameters outboundBonus, returnBonus in the POST/GET Reservation(/Advanced) response • outboundBonus, returnBonus parameters in GET flights response (v2.3) • parameters mil, mir, mnn, mns, mml in the GET flights request • fareCode parameter in the GET flights response in the BaggageAllowances array • fareCode parameter in POST/GET Reservation response (/Advanced) • flightsWithOpenDateCount parameter in POST/GET Reservation response (v2.3) • separatePrice, separateDiscount parameters in POST/GET Reservation (/Advanced) responses • separateFareFamilies, separateFlights, separateDuration, separateBaggageAllowances arrays in POST/GET Reservation (/Advanced) responses • new types of passengers MIR (RF Ministry of Internal Affairs), MNN (military child), MNS (military baby), MIL (RF Ministry of Internal Affairs, in development), MML (flying with a military man, in development) 	Savin A.A.
03.10.2023	03.10.2023	<p>Added:</p> <ul style="list-style-type: none"> • array of informationalServices[] in the api/Reservation response • parameters Code share and Code share Flight Number in the api/flights and api/Reservation response to determine the flight codeshare • validation of the number of characters for the full name has been changed, from 20 characters for each parameter to 43 characters for the sum of the full name 	Mikhaylov M.F.

03.08.2023	2.3.98	<p>Added:</p> <ul style="list-style-type: none"> • A FlightReference [] array into response post api/Partners/id into array PreselectedOption [] for indicating a flight segment with preselected additional option. • A RelatedChildrenReferenceNumbers [] array into responses: post and get api/Reservation; get api/Reservation/Advanced v2.3; post api/Reservation/AddSSR v2.3, to indicate child passengers which are travelling with an Adult. • A baseFarePricesBySegment [] array into responses: get and post api/Reservation v2.3; get api/Reservation/Advanced v2.3; post api/Reservation/AddSSR v2.3, with an information of base fare price for segments. • A refundOnlyWithTicket parameter into post api/Refund/Calculate; api/Refund/Partial; api/InvoluntaryRefund/Moneyless; api/InvoluntaryRefund which are indicating a responsibility to refund an additional option without a ticket. • A recommendations [] array was extended 'outboundDiscount' and 'returnDiscount' values for every valid combinations of flight segments for booking. 	Usolkin R.A.
------------	--------	---	--------------

30.06.2023	2.3.97	<p>Added:</p> <ul style="list-style-type: none"> code parameter to the recommendations[].outboundFareAndAmountDetails[] and recommendations[].returnFareAndAmountDetails[] arrays of GET api/flights service response, determines which fare brand the segment belongs to; check-in services; service of checking of participation of passengers in the loyalty program POST api/searchContact (unavailable for agents); optimized method of removing services from the reservation POST api/services/removeByReferences; preselectedSeats array of pre-added seats to the GET api/seatmap service response for the main IBE application; to the Attachment: Dictionaries description of statuses of seats on checkin. <p>Changed:</p> <ul style="list-style-type: none"> description of the documentNumber parameter in the GET api/reservation service response - masked according to the specified rules number of document of passenger; deleted taskId parameter (task Id in CRM) from the POST api/receipt service response. 	Gudumak S.E.
24.05.2023	2.3.96	<p>Added:</p> <ul style="list-style-type: none"> relatedInfantWithSeatId parameter for POST SeatmapStandalone, POST CatalogueStandalone services InfantsWithSeats parameter for POST Discounts, POST Reservation services ins parameter for GET Flight service an array for contacts not related to a passenger for POST Resrvation service flightNumber parameter type changed from string to integer 	Savin A.A.

12.04.2023	2.3.95	<p>Added:</p> <ul style="list-style-type: none"> • services for registration POST Checkin/reservation, POST Checkin/register, POST Checkin/cancel, GET Seatmap/checkin • updated method GET Checkin • boarding pass receipt service GET BoardingPass • services for working with certificates GET certificate/CheckCertificate/Advanced и POST Certificate/Refund • the country of South Ossetia has been added to the Attachment: Directories of countries 	Mikhaylov M.F
09.03.2023	2.3.94	<p>Added:</p> <ul style="list-style-type: none"> • the isGiftCertificate parameter in POST api/certificate/apply and POST api/certificate services responses, which indicates whether the certificate being applied is a gift certificate or not; • the appliedCertificates array in GET api/reservation response with information about applied certificates; • the quotas array in GET api/catalogue v2.3 and POST api/catalogueStandalone responses with information about available quotas for the service (currently this parameter is returned only for the service with the PETC code); • to the Attachment: Dictionaries of document types added other type of document with documentType = 3. 	Gudumak S.E.
13.02.2023	2.3.93	<p>Added:</p> <ul style="list-style-type: none"> • services for operating with certificates GET, POST, DELETE certificate, POST certificate/apply, GET certificate/CheckCertificate 	Savin A.A.

22.12.2022	2.3.92	<p>Added:</p> <ul style="list-style-type: none"> • the tourCode parameter in the POST reservation creation service; • service for the voluntary rebook of tickets on the ADP environment; • added expirationUtc parameter for the services GET reservation, GET reservation; • ability to create reservation with unaccompanied child. 	Savin A.A.
17.11.2022	2.3.91	<p>Changed:</p> <ul style="list-style-type: none"> • mandatory of query-parameters in GET api/seatmap request: only the referenceNumber of flight is required parameter; • format of loyaltyCard parameter in POST api/reservation request. 	Gudumak S.E.

03.10.2022	2.3.90	<p>Added:</p> <ul style="list-style-type: none"> • parameters checkedIn, checkInHasStarted are supported and parameters checkedInSegment, checkInHasStartedSegment are added in GET api/reservation and GET api/reservation/advanced; • parameters couponStatuses, couponStatusesBySegment in GET api/reservation and GET api/reservation/advanced are supported; • parameters flightReferences are added in GET reservation in arrays baseFarePrices и tickets; • MMI, INS passenger types are added; • DELETE api/reservation method to delete unpaid reservations was added; • the ability to pass PNR and lastName parameters in Cyrillic in GET api/reservation method; • description of SSR with SPAX type to Attachment: Service SSR; • ability to open reservation in BMM via ticketNumber; • booking payment service api/payment; • online checkin service api/checkin; • change passenger's data service api/passenger; • contact info add service api/Reservation/AddContactInfo; • service for retrieving SSR and remarks information api/version/Reservation/Advanced; • service for add SSR to the reservation /Reservaton/AddSSR. 	<p>Sheremet T.Y.</p> <p>Gudumak S.E.</p> <p>Savin A.A.</p> <p>Mikhaylov M.F.</p>
01.09.2022	2.3.89.1	<p>Changed the title and description for ADP method POST api/InvoluntaryRefund/moneyless.</p>	<p>Gudumak S.E.</p>

29.08.2022	2.3.89	<p>Added:</p> <ul style="list-style-type: none"> • services for working with queues on ADP environment; • STR, JNN passenger types are added to the EN_8_1_Attachment: Dictionaries; • service for obtaining recommendations for exchange; • accountCode parameter is supported in GET api/calendar, GET api/flights, POST api/reservation; • endorsements array is supported in POST api/reservation and GET api/reservation; • added the ability to receive discounts on flight via conventional and promocode discounts; • added the ability to receive TFC (Tariff Application Conditions) in POST/flights/conditions; • to the response of GET api/reservation method information about spent bonuses: spentBonuses and spentBonusesByPassengers; • the parameters of loyaltyCard and loyaltyCardValidTo to the POST api/reservation request; • POST api/InvoluntaryRefund method; • POST api/InvoluntaryRefund/moneyless method for ADP which let to refund tickets without returning money; • POST api/refund/Calculate and POST api/refund/partial methods - standard methods for calculating and performing voluntary refund; • GET api/refund method for getting information about already performed refunds; • the Attachment: Service SSR; • POST api/user method. 	<p>Gudumak S.E.</p> <p>Savin A.A.</p> <p>Sheremet T.Y.</p>
------------	--------	---	--

<p>26.07.2022</p>	<p>2.3.88</p>	<p>Added the ability to create reservations for subsidized passengers:</p> <ul style="list-style-type: none"> ▪ src, yth parameters are supported in GET api/calendar, GET api/flights; ▪ the isSubsidized parameter is supported in POST api/reservation; ▪ the attachment dictionaries of passenger types has been updated. <p>The possibility of payment in different currencies has been implemented:</p> <ul style="list-style-type: none"> ▪ the currency parameter is supported in GET api/flights; ▪ the currency parameter is not used when creating a reservation in the POST api/reservation, a note has been added. <p>The expectedPrice and searchPrice parameters are not used when creating a reservation.</p> <p>Added the ability to cancel seats</p> <p>Added the ability to partially cancel a booking</p> <p>Added the ability to add partner service to a booking</p> <p>Added the ability to void tickets and additional services</p>	<p>Gudumak S.E.</p> <p>Savin A.A.</p> <p>Sheremet T.Y.</p>
-------------------	---------------	--	--

<p>07.07.2022</p>	<p>2.3.87</p>	<p>Initial version of the documentation.</p> <p>Important changes after integration with Sirenaz-Travel:</p> <ul style="list-style-type: none"> ▪ changed GDS; ▪ changed booking process that now it is available only in one step without advanced reservation; ▪ to the responses of POST/GET api/reservation added new parameters relatedAdultReferenceNumber and relatedInfantReferenceNumber; ▪ in the response of POST/GET api/reservation changed the description of engineType parameter; ▪ added the second way of getting the itinerary receipt; ▪ services statuses Dictionary was added to Attachment 8_1. <p><i>Note: some services and parameters are under development, so they are marked accordingly.</i></p>	<p>Gudumak S.E.</p> <p>Savin A.A.</p>
-------------------	---------------	--	---------------------------------------

2_Glossary

U6	Ural Airlines company
IBE, booking engine	Internet Booking Engine, U6 new ticket booking system. IBE API is described in this document.
Sirena	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 Sirena 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 Sirena document with information regarding tariff options in PNR.
TSM	Transitional Stored Miscellaneous, part of Sirena document with information regarding auxiliary services, added in PNR.
GOI	Get Order Info, service for issuing tickets, airline's and partner's services.
BMM	Booking Management Module
RT	Round-Trip flight designation

3_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 Sirena/SIG/CRM/BPC systems, other partners services and end-user client applications.

4_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

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.

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)	authorization API-key	Service API methods
X-Session	sessionKey — value, retrieved from API Session query	Session management via HTTP header	POST session method GET flights method GET calendar method GET reservation method GET receipt without parameters Services methods except problemNotification.
Content-Type	application/json	General	No

IBE API services query

IBE API query string in common case (except service API methods) 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.

5_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 one way: single query passing all data.
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
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

6_API methods description

6_1_api/session: 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"  
}
```

6_3_api/flights: obtain flights and flight conditions 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	Sirena: pricing_route, pricing_route_and_mono_brand_tail SIG: SIG_AirShop CRM: CalculateDiscount

GET flights method

Description: obtain list of flights and flight's prices

Request format: /api/apiVersion/flights

Request header

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

Request samples

One way flight:

/api/apiVersion/flights?orig=LED&dest=MOW&dep=20220710&mode=Mono
&adt=1&chd=0&inf=0&src=0&yth=0&umnr=0&str=0&jnn=0&gcf=0&jcb=0

Two way flight:

/api/apiVersion/flights?orig=LED&dest=MOW&dep=20220710&ret=20220717&mode=Mono
&adt=1&chd=0&inf=0&src=0&yth=0&umnr=0&str=0&jnn=0&gcf=0&jcb=0

Request parameters

IMPORTANT: for correct response non-subsidized (ADT/CHD) or subsidized (SRC/YTH)passengers types should be used in request; to receive recommendations for round trip reservation need to use mode=Mono parameter

Parameter	Type	Description	Value samples	Notes
orig	string	Original airport IATA-code	LED	
dest	string	Destination airport IATA-code	DME	
dep	date	Departure date	Format yyyyMMdd Example: 20180801	
ret	date	Return date	Format: yyyyMMdd Example: 20180905	Can be passed only with the parameter dep
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		Can be passed only with the parameter adt, src, yth
ins	integer	Number of Infant with seats passengers		Can be passed only with the parameter adt, src, yth
src	integer	Number of Senior citizen passengers		Not suitable passenger type for issuing a booking via Agents

<i>yth</i>	integer	Number of Youth passengers		Not suitable passenger type for issuing a booking via Agents
<i>cnn</i>	integer	Number of passengers type children which are traveling with subsidized passenger type		
<i>umnr</i>	integer	Number of accompanied children		
<i>str</i>	integer	Number of passengers of subsidized adult for the Far East		Not suitable passenger type for issuing a booking via Agents
<i>jnn</i>	integer	Number of passengers of subsidized children for the Far East		Not suitable passenger type for issuing a booking via Agents
<i>gcf</i>	integer	Number of passengers of subsidized pensioner for the Far East		In developing
<i>jcb</i>	integer	Number of passengers of subsidized youth passengers for the Far East		In developing
<i>mil</i>	integer	Number of passengers with military travel document of "Ministry of Defence of the Russian Federation"		
<i>mmi</i>	integer	Number of passengers of the type "Ministry of Defence of the Russian Federation"		
<i>mir</i>	integer	Number of passengers of the type "Ministry of Internal Affairs of the Russian Federation"		
<i>mnn</i>	integer	Number of passengers of the "military children" type		
<i>mns</i>	integer	Number of "infant with seat" passengers		

mml	integer	Number of passengers flying with a military man		
dis	integer	Number of adult passengers with disabilities and I or II group disability		in developing
dcd	integer	Number of passengers of type "subsidized 2-11 years old child with disability"		in developing
dic	integer	Number of passengers of type "subsidized 12-17 years old child with disability"		in developing
dac	integer	Number of adult passengers of type "Accompanier for I group person with disability (SBK) or child with disability (DCD or DIC)"		in developing
sbk	integer	Number of adult passengers of type "Subsidized person with I group disability"		in developing
stu	integer	Number of passengers of type "student 14+ years old is learning at the full-time form or military college at the Kaliningrad city or Kaliningrad region. Kaliningrad's subsidy"		Not suitable passenger type for issuing a booking via Agents
chr	integer	Number of passengers of type "Child 2-11 years old is official living at the Kaliningrad city or Kaliningrad region. Kaliningrad's subsidy"		Not suitable passenger type for issuing a booking via Agents
mode	string	Search mode (optional)	mono - only mono-recommendations are requested, for example, PR there and PR back; full - all recommendations are requested.	For the Sirena search now it uses mode=Mono, full - in developing

<i>ofn</i>	string	Array of one-way flight numbers (optional)	60	In developing
<i>rfn</i>	string	Array of return flight numbers (optional)	371,392	In developing
accountCode	string	Personal account client code (optional)	ACCadd005	
<i>meta</i>	boolean	Sign of metasearch (optional)		In developing
currency	string	Currency of recommendations (optional)	EUR	
<i>cabin</i>	string	Class of service (optional)	Economy, Business	In developing
<i>SearchAllBrands</i>	bool	Search for all brands (optional)		In developing
subsidies_type	string	Type of subsidies (optional)	<p>Numeric value from 0 to 3: Disability = 0, Kaliningrad = 1, FarEast = 2, Age = 3.</p> <p>String value: Disability = people with disabilities, Kaliningrad = Kaliningrad, FarEast = Far Eastern District, Age = Age subsidies.</p>	

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": integer,
    "commercialFamily": string,
    "translation": {
        "ru": string,
        "en": string,
        "zh": string
    }
}],
"alternativeFlights": [{
    "departure": string,
    "arrival": string
}],
"discountConstraints": [{
    "fareFamilies": [string],
    "flights": [string],
    "excludedFlights": [{
        "flightNumber": integer,
        "departureAirport": string,
        "arrivalAirport": string
    }
    ]
}],
"recommendations": [{
    "outboundFlightGroupId": integer,
    "outboundFareFamilyId": integer,
    "returnFlightGroupId": integer,
    "returnFareFamilyId": integer,
    "outboundFlightGroupHash": string,
    "returnFlightGroupHash": string,
    "outboundFareFamilyCode": string,
    "returnFareFamilyCode": string,
    "outboundPrice": integer,
    "outboundTaxes": integer,
    "returnPrice": integer,
    "returnTaxes": integer,

```

```

"outboundSeatsLeft": integer,
"returnSeatsLeft": integer,
"totalPrice": integer,
    "outboundFareAndAmountDetails": [{
        "bookingCode": string,
        "fareBasis": string,
        "isBreakpoint": boolean,
        "paxType": integer,
        "paxCount": integer,
        "taxAmount": integer,
        "totalAmount": integer,
        "isSale": boolean,
        "code": string
    }
],
"returnFareAndAmountDetails": [{
    "bookingCode": string,
    "fareBasis": string,
    "isBreakpoint": boolean,
    "paxType": integer,
    "paxCount": integer,
    "taxAmount": integer,
    "totalAmount": integer,
    "code": string
}
],
"outboundBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer,
    "fareCode": string
}
],
"returnBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer,
    "fareCode": string
}
],
"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": integer,
        "departureAirport": string,
        "arrivalAirport": string
    }
]
    }
},
    "currency": 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": integer,
  "marketingCarrier": string,
  "operatingCarrier": string,
  "codeshare": string,
  "codeshareFlightNumber": 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	Notes
-----------	-------------	-------

success	Sign of successful query execution If errors array is empty, the value of field will be 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 subsidized tariff	
	<i>canRefund</i>	Sign of refundable tariff	For Sirena: in developing
	hasUpsale	Sign identifying 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	
	<i>outboundNextNearestFlightDate</i>	The nearest departure date for one-way flight, if there are no flights on the requested date	For Sirena: in developing
	<i>returnNextNearestFlightDate</i>	The closest date of departure back, if the flight is RT and there are no flights for the requested dates	For Sirena: in developing
	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	
	<i>totalDiscount</i>	Amount of discount	For Sirena: in developing
	totalBaseFare	Total amount of base fare	
	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	Information about discount available for present flight Discount may absent for certain flight they are in the list of exceptions. Exceptions stored on CRM and IBE side.	
	id	Discount campaign id in CRM	
	name	Discount campaign title	
	amount	Discount amount	

		discountType	Discount type, possible values 0 – Percent 1 – Roubles It is recommended to check discount type against its value. For ex for type 0 amount should be within [0;100] interval.	
		usageType	Discount application type, possible values: 0 – Fare 1 – Order	
		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	Promo code status Possible values: dictionary of promo codes statuses	
		promotionType	Promo code type 0 - promotional code for the flight 1 - promotional code for an additional service	
		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	
		description	Code description	
		messages	Array of objects, containing errors from Sirena/SIG response. Array can be empty, it is recommended to analyze errors by ibeCode.	
		code	Sirena/SIG error code	
		text	Sirena/SIG error text	

Object flightGroup

Contains information about flights

data.outboundFlights or data.returnFlights

flightGroupId	index number used in recommendations
flightGroupDuration	Total flight time: summed up of duration values of each transfer and technical stop.
hash	Service field
flights	Array of objects with information
flightId	Flight id
departureAirport	Airport of departure
departureTerminal	Terminal in airport of departure
arrivalAirport	Airport of arrival
arrivalTerminal	Terminal in airport of arrival
departureDateTime	Departure date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T17:25:00
excludedNationalities	Array with information about passengers with citizenships prohibited for transportation on present flight Example: ISR (for flights 3837 and 3838)
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
flightDurationUntilNextStop	Flight duration until <u>first</u> technical stop
stopDuration	Technical stop duration
flightNumber	flight number
marketingCarrier	Marketing carrier Example: U6, S7

operatingCarrier	Operating carrier (one actually performed flight) Example: U6, OK
codeshare	Full aviacompany title performing code-share flight Example: Czech Airlines for operatingCarrier = OK For operatingCarrier = U6, present field is empty
codeshareFlightNumber	The flight number of the airline that operates the codeshare flight.
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

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
code	The parameter that determines which fare brand the segment belongs to.

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
fareCode	Fare family for which baggage is returned

totalPrice field value examples

One way flight, no discounts

totalPrice is equal to value outboundPrice

```

▼ recommendations: [{outboundFlightGroupId: 0, outboundFareFamilyId: 1,...},...]
  ▼ 0: {outboundFlightGroupId: 0, outboundFareFamilyId: 1,...}
    ► outboundBaggageAllowances: [{quantity: 1, weight: 0, passengerType: 0}]
    ▼ outboundFareAndAmountDetails: [,...]
      ► 0: {bookingCode: "W", fareBasis: "WPROW", isBreakpoint: true, paxType: 0, paxCount: 1, taxAmount: 978,...}
        outboundFareFamilyCode: "PR"
        outboundFareFamilyId: 1
        outboundFlightGroupHash: "07/01/2022 16:25:00-60-LED-DME"
        outboundFlightGroupId: 0
        outboundPrice: 998
        outboundSeatsLeft: 9
        outboundTaxes: 978
        returnBaggageAllowances: []
        returnFareAndAmountDetails: []
        totalBaseFare: 20
        totalDiscount: 0
        totalPrice: 998
      ► 1: {outboundFlightGroupId: 0, outboundFareFamilyId: 2,...}
      ► 2: {outboundFlightGroupId: 0, outboundFareFamilyId: 3,...}
      ► 3: {outboundFlightGroupId: 0, outboundFareFamilyId: 0,...}
      returnFlights: []
    errors: []
    success: true

```

Two way flight:

totalPrice = outboundPrice + returnPrice

totalPrice = 998 + 140 = 1138

```

▼ recommendations: [,...]
  ▼ 0: {outboundFlightGroupId: 0, outboundFareFamilyId: 1, returnFlightGroupId: 0, returnFareFamilyId: 1,...}
    ▶ outboundBaggageAllowances: [{quantity: 1, weight: 0, passengerType: 0}]
    ▶ outboundFareAndAmountDetails: [,...]
      outboundFareFamilyCode: "PR"
      outboundFareFamilyId: 1
      outboundFlightGroupHash: "07/01/2022 16:25:00-60-LED-DME"
      outboundFlightGroupId: 0
      outboundPrice: 998
      outboundSeatsLeft: 9
      outboundTaxes: 978
    ▶ returnBaggageAllowances: [{quantity: 1, weight: 0, passengerType: 0}]
    ▶ returnFareAndAmountDetails: [,...]
      returnFareFamilyCode: "PR"
      returnFareFamilyId: 1
      returnFlightGroupHash: "07/08/2022 14:00:00-59-DME-LED"
      returnFlightGroupId: 0
      returnPrice: 140
      returnSeatsLeft: 9
      returnTaxes: 120
      totalBaseFare: 40
      totalDiscount: 0
      totalPrice: 1138
    ▶ 1: {outboundFlightGroupId: 0, outboundFareFamilyId: 2, returnFlightGroupId: 0, returnFareFamilyId: 2,...}
    ▶ 2: {outboundFlightGroupId: 0, outboundFareFamilyId: 3, returnFlightGroupId: 0, returnFareFamilyId: 3,...}
    ▶ 3: {outboundFlightGroupId: 0, outboundFareFamilyId: 0, returnFlightGroupId: 0, returnFareFamilyId: 0,...}
  ▶ returnFlights: [{flightGroupId: 0,...}]
errors: []
success: true

```

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 header

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

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":
      "eJxdjleKwjAQRK8yFxB2N5uk7SUUf/odpGBAFKr44a8XKHhh042VKpkEJpO8WRFchzGnU36kW76c
      uy6Nh2O+D2AFIZZNYKcsGiDYcLHlULAni0TQekiAR7/bb3s4+1JXAYERani54AhVaPPNmNibqfnc+
      alwoY0ssVhYi1X/Ele0eVSh2AbyzgjGWivR+BVTtPD/nlU5PKfXG/geN0w="
    },
    {
      "key":
      "eJxdjleKwjAQRK8yFxB2N5uk7SUUf/odpGBAFKr44a8XKHhh042VKpkEJpO8WRFchzGnU36kW76c
      uy6Nh2O+D2AFIZZNYKcsGiDYcLHlULAni0TQekiAR7/bb3s4+1JXAYERani54AhVaPPNmNibqfnc+
      alwoY0ssVhYi1X/Ele0eVSh2AbyzgjGWivR+BVTtPD/nlU5PKfXG/geN0w="
    }
  ]
}
```

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 Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode
code	Sirena error code
text	Sirena error text
success	Sign of successful request execution If the errors array is empty, the value of field 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 header

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

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	Flight type
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
updateDT	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 ibeCode 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": "XXXXXX",
        "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 header

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

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

Method GET Flights

Description: It lets to get flights which are available for booking and are matching with specific parameters of the request.

Discounts parameters by routes are included to the response.

Request format /api/apiVersion/flights

Request parameters: It has the same realisation as for get api/Flights v2.2

Request header

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

Changes in the api/apiVersion/flights version 2.3:

A new parameter `subsidies_type` is submitted in the search for subsidized flight options. Both string and numeric values can be given - `subsidies_type: 2 / subsidies_type: FarEast`

Request example:

One way flight:

```
/api/apiVersion/flights?orig=LED&dest=MOW&dep=20220710&mode=Mono
&adt=1&chd=0&inf=0&src=0&yth=0&umnr=0&str=0&jnn=0&gcf=0&jcb=0
```

Response body

```
{
  "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": integer,
    "commercialFamily": string,
    "translation": {
      "ru": string,
      "en": string,
      "zh": string
    }
  }
],
  "alternativeFlights": [{
    "departure": string,
```



```

    },
  ],
  "outboundBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer,
    "fareCode": string
  }
],
  "returnBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer,
    "fareCode": string
  }
],
  "outboundBonus": {
    "bonus": "string",
    "replacedBonus": "string"
  },
  "returnBonus": {
    "bonus": "string",
    "replacedBonus": "string"
  },
  "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": integer,
        "departureAirport": string,
        "arrivalAirport": string
    }
    ],
    }
    ],
    },
    ],
    "moneyWallCoefficient": integer,
    "containsInternationalFlights": bool
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ]
}
],
}
],
"success": boolean
}

```

Response parameters (at the short form)

A description of parameters are the same as for get api/Flights v2.2

The recommendations [] array are extended by parameters for any available combination of booking:

Parameter	Description	Notes
recommendations	Array of recommendations objects for flights	
outboundDiscount	An amount of discount for "There" route	
returnDiscount	An amount of discount for "From" route	
outboundBonus	The number of bonuses awarded for a flight "there"	
bonus	Current number of bonuses awarded for the flight	
replacedBonus	Outdated number of bonuses awarded per flight	
returnBonus	The number of bonuses awarded for a return flight	
bonus	Current number of bonuses awarded for the flight	
replacedBonus	Outdated number of bonuses awarded per flight	

Method GET Flights/meta/search

Description: realizing option to get flight recommendations by Yandex Webmaster with related parameters of searching.

Request format: /api/apiVersion/flights/meta/search/api/apiVersion/flights/meta/search

Request header: X-Api-Key - unique GUID of system using IBE API

Request parameters:

Parameter	Description
departure	Location of departing (IATA code)
arrival	Location of arriving (IATA code)
date_forward	Planned date of departure (format: yyyy-mm-dd)
date_backward	Planned date of arrival (optional)
class	Class of service (Available options: Economy; Business)
adults	An amount of Adults, 12 years old and more.
children	An amount of Childs with seats. 2 years old until 11 years old.
infants	An amount of Infants without seats. Under 2 years old.

Notice: An amount for passengers with seats should be can 9 in total or less; a request used the 'infants' passenger type has required rule - Adults amount \geq Infants amount; a request used the 'children' passenger type has required rule - adults amount \geq 1

Request example:

OW flight:

```
/api/apiVersion/search?departure=SVX&arrival=DME&date_forward=2025-12-02&date_backward=2025-12-10&adults=1&children=1&class=business
```

Response body:

```
{
  "variant": [
    {
      "url": string,
      "price": {
        "value": integer,
        "currency": string,

```

```

    "tariff_adult": integer,
    "tariff_child": integer,
    "tariff_infant": integer,
    "refundable": boolean,
    "seats_remaining": integer
  },
  "route_forward": [
    {
      "flight_number": string,
      "carrier_name": string,
      "marketing_carrier": string,
      "operating_flight_number": string,
      "operating_carrier": string,
      "validating_carrier": string,
      "departure_airport_code": string,
      "arrival_airport_code": string,
      "departure_datetime": datetime,
      "arrival_datetime": datetime,
      "fare_code": string,
      "fare_family": string,
      "class": string,
      "charter": boolean,
      "selfconnect": boolean,
      "technical_stop": boolean,
      "luggage": {
        "included": boolean,
        "pieces": integer,
        "weight": integer,
        "units": string,
        "dimensions": {
          "length": integer,
          "width": integer,
          "height": integer,
          "sum": integer
        }
      }
    },
    {
      "hand_luggage": {
        "included": boolean,
        "pieces": integer,
        "weight": integer,
        "units": string,
        "dimensions": {
          "length": integer,
          "width": integer,
          "height": integer,
          "sum": integer
        }
      }
    }
  ],
  "route_backward": [
    {
      "flight_number": string,
      "carrier_name": string,
      "marketing_carrier": string,
      "operating_flight_number": string,
      "operating_carrier": string,

```


	currency	Recommendation currency.	
	tariff_adult	A total amount price for whole existed Adult passengers.	
	tariff_child	A total amount price for whole existed passengers like 'Children'.	
	tariff_infant	A total amount price for whole existed infant passengers without seats.	
	refundable	Available of refunding the booking.	
	seats_remaining	A total amount of seats by the current price.	
	route_forward	An array of parameters data for 'Forward' flight.	
	flight_number	Airline code and flight number.	
	carrier_name	Full name of carrier Airline.	
	marketing_carrier	Selling ticket Airline code.	
	operating_flight_number	Flight number by the Airline.	
	operating_carrier	Carrier code.	
	validating_carrier	Carrier code Issuing ticket.	
	departure_airport_code	Departure airport code.	
	arrival_airport_code	Arrival airport code.	
	departure_datetime	Departing date time (local time)	
	arrival_datetime	Arriving date time (local time)	
	fare_code	Fare code	
	fare_family	Fare category	
	class	Class of service	Available values: BUSINESS; ECONOMY
	charter	Charter tariff parameter	
	selfconnect	A parameter of Self connecting flight segments for transfer flights and issuing unique tickets for every flight segment at the booking.	
	technical_stop	Existing tech stop until the flight parameter	
	luggage	An array of data for luggage parameters	

		included	A Parameter of existing a luggage at the tariff	
		pieces	An amount of luggage included at the tariff	
		weight	Available weight of one luggage piece included at the tariff.	
		units	A unit of measurement for luggage weight.	
		dimensions	An array of physical parameters data for a luggage	
		length	Available length of luggage	
		width	Available width of luggage	
		height	Available height of luggage	
		sum	Available sum of three luggage parameters in total	
		hand_luggage	An array of data for hand luggage parameters	
		included	A Parameter of existing a hand luggage at the tariff	
		pieces	An amount of hand luggage included at the tariff	
		weight	Available weight of one hand luggage piece included at the tariff.	
		units	A unit of measurement for hand luggage weight.	
		dimensions	An array of physical parameters data for a hand luggage	
		length	Available length of hand luggage	
		width	Available width of hand luggage	
		height	Available height of hand luggage	
		sum	Available sum of three hand luggage parameters in total	
		route_backward	An array of parameters data for 'Backward' flight.	
		flight_number	Airline code and flight number.	
		carrier_name	Full name of carrier Airline.	
		marketing_carrier	Selling ticket Airline code.	
		operating_flight_number	Flight number by the Airline.	
		operating_carrier	Carrier code.	
		validating_carrier	Carrier code Issuing ticket.	
		departure_airport_code	Departure airport code.	
		arrival_airport_code	Arrival airport code.	

	departure_datetime	Departing date time (local time)	
	arrival_datetime	Arriving date time (local time)	
	fare_code	Fare code	
	fare_family	Fare category	
	class	Class of service	Available values: BUSINESS; ECONOMY
	charter	Charter tariff parameter	
	selfconnect	A parameter of Self connecting flight segments for transfer flights and issuing unique tickets for every flight segment at the booking.	
	technical_stop	Existing tech stop until the flight parameter	
	luggage	An array of data for luggage parameters	
	included	A Parameter of existing a luggage at the tariff	
	pieces	An amount of luggage included at the tariff	
	weight	Available weight of one luggage piece included at the tariff.	
	units	A unit of measurement for luggage weight.	
	dimensions	An array of physical parameters data for a luggage	
	length	Available length of luggage	
	width	Available width of luggage	
	height	Available height of luggage	
	sum	Available sum of three luggage parameters in total	
	hand_luggage	An array of data for hand luggage parameters	
	included	A Parameter of existing a hand luggage at the tariff	
	pieces	An amount of hand luggage included at the tariff	
	weight	Available weight of one hand luggage piece included at the tariff.	
	units	A unit of measurement for hand luggage weight.	
	dimensions	An array of physical parameters data for a hand luggage	

			length	Available length of hand luggage	
			width	Available width of hand luggage	
			height	Available height of hand luggage	
			sum	Available sum of three hand luggage parameters in total	

6_4_api/reservation: booking management service

Booking management service v2.2

API service	reservation
Short description	<p>The service allows to manage booking:</p> <ul style="list-style-type: none"> - add information to the booking (data on flights, passengers and fares); - receive information about the reservation; - delete unpaid reservations.
Supported HTTP(S)-methods	GET, POST, DELETE
Used methods of related systems	<p>Sirena:</p> <p>Create booking - booking</p> <p>Getting information on booking - order, pnr_status</p> <p>Getting information about coupon statuses - view_flown_status</p> <p>Cancel booking - booking-cancel</p>

Method POST reservation

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

Important: There is only one option for creating a booking without advanced booking when all parameters for flights, fares and passengers are passed in one request.

Disabled an ability for creating a booking with subsidized fares via agents.

Request format /api/apiVersion/reservation

Request headers: X-Api-Key - API key

Request body

```
{
  "flights": {
    "outboundFlights": [reservationFlight],
    "returnFlights": [reservationFlight],
    "adults": integer,
    "children": integer,
    "infants": integer,
    "infantsWithSeats": 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": ingeger,
"searchPrice": ingeger,
"airShopSessionId": string,
"reservationContacts"[
  {
    "email": string,
    "phone": string
  }
],
"subsidiesType": "Disability"
}

```

reservationFlight object

```

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

```

The relatedAdultReferenceNumber parameter is added to the passengers[] array for CHD/INS/INF . The parameter is not available for other types of passengers.

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,
  "relatedAdultReferenceNumber": string,
  "disability": {
    "isDeaf": boolean,
    "isBlind": boolean,
    "hasGuideDog": boolean,
    "assistanceType": integer,
    "wheelchairType": integer,
    "commentary": string
  },
  "socialSecurityNumber": string,
  "endorsements": [{

```

```

    "endorsement": boolean,
    "isDisability": boolean,
    "endorsementDocNumber": string,
    "endorsementDocDate": datetime,
    "loyaltyPartnerCard": string
  }
],
"accompanimentBookingInfo": [{
  "endorsement": boolean,
  "isDisability": boolean
}]
],
"accompaniedPassengerId": number,
"SubsidyDocumentNumber": string
}

```

Request parameters

Parameter	Description	Notes
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, infantsWithSeats pensioners, youths, subsidizedChildren	Count of passengers of each type	
origin	Departure airport	
destination	Destination airport	
passengersCount	Total count of passengers with the seat	
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 fares	
accountCode	Account Code	
tourCode	Tour Code	

	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	Optional parameter
	returnFareFamily	Object with information about fare family 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 fare	Optional parameter
	fares	An array with information about fares for directions "outbound" and "return"	
	fareBasis	Base fare title Data is taken from the response api/flights: data.recommendations.outboundFareDetails.fareBasis and data.recommendations.returnFareDetails.fareBasis	

	isBreakpoint	Flag identifying whether fare 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	This parameter is not used to create a booking in Sirena, because for offices, the default currency is set in the settings.
	expectedPrice	The expected price of the booking, taken from the api/flights	This parameter is not used to create a booking.
	searchPrice	The price that was received in the search request, taken from the api/flights or api/flights/search	This parameter is not used to create a booking
	airShopSessionId	SIG search session ID, taken from the api/flights response	
	reservationContacts	Contacts that are not related to the passengers	
	email	Email	
	phone	Phone	
	subsidiesType	Type of subsidies	

reservationFlight object

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

passengerData object

Parameter	Description	Format	Notes
id	Passenger's serial number (numbered from 1)		
firstName	Passenger's name	[A-Z], from two characters	The sum of the full name is no more than 43 characters
middleName	Passenger's patronymic (may be absent)	[A-Z], from one character	The sum of the full name is no more than 43 characters
surname	Passenger's surname	[A-Z], from two characters	The sum of the full name is no more than 43 characters
passengerType	Passenger type		
birthDate	Date of birth		
sex	Passenger gender Possible values: see dictionary "gender of passengers"		
nationality	Three-letter code of country of citizenship Example: RUS Possible values: dictionary of countries		

documentNumber	Document number	<p>For different document types there is different format rules:</p> <ul style="list-style-type: none"> ▪ Passport of Russian Federation: 10 chars; ▪ Foreign passport of Russian Federation: 9 chars; ▪ Birth certificate of Russian Federation: $^{[IVX]}+([A-Яа-я]\{2\})[A-Za-z]\{2,4\})d\{6\}\$.$ <p>For documents of other countries there is no special format rules.</p>	
documentType	<p>Document type</p> <p>Possible values: <u>Attachments: dictionaries</u></p>		
documentExpiryDate	<p>Date until which the document is valid</p> <p>Mandatory only for a foreign passport of Russian Federation and national passports of other countries. For other documents, it is automatically set by Sirena.</p>		
loyaltyCard	Loyalty card number "Wings"	10 numbers	

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		In developing
email	Email		
phone	Phone		
relatedAdultReferenceNumber	Identifier of the adult attached to children		Only used for passenger types CHD/INS/INF
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: <u>dictionary of assistance types</u>		
wheelchairType	Wheelchair type Possible values: <u>dictionary of wheelchair type</u>		
commentary	Comment (optional)		
socialSecurityNumber	A specific number of SNILS (optional)	10 digits	

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. Example: disability certificate, ID number or other document confirming the status of a large family.		
endorsementDocDate	Subsidy document issue date.		
loyaltyPartnerCard	Loyalty partner card number.		
accompanimentBookingInfo	Information about the reservation containing an accompanying passenger		Currently used when creating reservations for MNN, MNS
number	PNR of a booking containing an accompanying passenger		
surname	Last name of the accompanying passenger		
accompaniedPassengerId	Id number of accompanied passenger (related with subsidized booking by Disability subsidies)		
SubsidyDocumentNumber	Document number for confirming a right to get a subsidy (related for subsidized booking by Disability subsidies or Kaliningrad's subsidy)		

Response format

```

{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "returnFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string,
      },
      "returnFareFamily": {
        "code": string,
      },
      "separateFareFamilies": {
        "code": string,
      },
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    },
    "number": string,
    "reservationDate": datetime,
    "passengers": [{
      "id": integer,
      "hasInfant": boolean,
      "referenceNumber": string,
      "relatedAdultReferenceNumber": string,
      "relatedInfantReferenceNumber": string,
      "relatedChildrenReferenceNumbers": [
        "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": {
        couponReference: boolean
      },
      "checkInHasStarted": {
        couponReference: boolean
      },
      "checkedInSegment": {
        flightReference: boolean
      },
      "checkInHasStartedSegment": {
        flightReference: boolean
      },
      "hasPassengersDataErrors": boolean,
    }
  ]
}

```

```

    "passengerDataErrors": [{
      "key": string,
      "value": string
    }],
    "endorsements": [{
      "endorsement": boolean,
      "isDisability": boolean,
      "loyaltyPartnerCard": string
    },
    {
      "endorsement": boolean,
      "isDisability": boolean,
      "endorsementDocNumber":
string,passengerDataErrors
      "endorsementDocDate": datetime
    }
  ],
  "outboundBonus": {
    "bonus": "string",
    "replacedBonus": "string"
  },
  "returnBonus": {
    "bonus": "string",
    "replacedBonus": "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": integer,
    "currency": string
  }],
  "ticketFaresWithoutZapOff": [{
    "baseAmount": integer,
    "passengerReferences": [{
      "referenceNumber": string,
      "referenceQualifier": string
    }],
    "passengerType": integer,
    "referenceNumber": string,
    "referenceQualifier": string,
    "totalAmount": integer,

```

```

        "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,
    "statuses": [
        string
    ],
    "codeshare": string,
    "codeshareFlightNumber": string,
    "isCancelledByAirline": boolean,
}

```

```

    "isCheckinWithoutPseudo": boolean
}

```

Response parameters

Parameter	Description	Notes
success	Sign of the successful request execution If the errors array is empty, the value of field 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	
outboundBonus	The number of bonuses awarded for a flight “there”	
bonus	Current number of bonuses awarded for the flight	
replacedBonus	Outdated number of bonuses awarded per flight	
returnBonus	The number of bonuses awarded for a return flight	
bonus	Current number of bonuses awarded for the flight	
replacedBonus	Outdated number of bonuses awarded per flight	
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	In developing
luggageSeats	An array of objects with information about luggage with separate air ticket and ssr CBBG.	In developing
referenceNumber	The reference to CBBG luggage.	

Parameter	Description	Notes
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	
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	
returnFareFamily	Service field containing fareFamily code	
separateFlights	An array of fullReservationFlight objects with detailed information about flights without reference to direction	
separateFareFamilies	Service field containing fareFamily code	
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	

Parameter	Description	Notes
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: IBE error codes	
description	Error description, see Attachment: 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	

fullReservationFlight

Parameter	Description	Notes
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	
statuses	Array with information about segment statuses	

Parameter	Description	Notes
codeshare	The full name of the airline that operates the codeshare flight. For example: Czech Airlines for operating Carrier = OK For operatingCarrier = U6, this field is not filled in	
codeshareFlightNumber	The flight number of the airline that operates the codeshare flight.	
isCancelledByAirline	Sign of cancellation of flights by the airline	
isCheckinWithoutPseudo	Sign of hiding a pseudo-booking at the checkin	

passengerData Response object

Parameter	Description	Notes
id	Passenger's serial number (numbered from 1)	
hasInfant	An sign that this passenger has a baby	
relatedAdultReferenceNumber	Reference to the adult passenger in the booking with whom the child or infant flies (it returns only for children and infants)	
relatedInfantReferenceNumber	Reference to the infant passenger in the booking with whom the adult flies (it returns only for adult)	
relatedChildrenReferenceNumbers	An array of links to child passengers which are travelling with an Adult (the array will be received only for an Adult with a children in the same booking)	
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	
isAdult	Whether the passenger is an adult true for ADT/SRC/YTH type passengers false for CHD/INF type passengers	
birthDate	Date of birth	

Parameter	Description	Notes
sex	<p>Passenger gender</p> <p>Possible values: see <u>dictionary "gender of passengers"</u></p>	
nationality	<p>Three-letter code of country of citizenship</p> <p>Example: RUS</p> <p>Possible values: <u>dictionary of countries</u></p>	
documentNumber	Document number	
documentType	<p>Document type</p> <p>Possible values: <u>dictionary of document types</u></p>	
documentExpiryDate	<p>Date until which the document is valid</p> <p>Mandatory only for a foreign passport. For other documents, +10 years from the current date is automatically set</p>	
email	Email	
phone	Phone	
checkedIn	Passenger's check-in information	
checkedInSegment	Passenger's check-in on flight segment information	
checkInHasStarted	Check-in beginning's information	
checkInHasStartedSegment	Check-in beginnings on flight segment information	
hasPassengersDataErrors	The sign of errors in passengers data.	
passengerDataErrors	Array with detailed information of errors in passengers data.	
key	The key.	
value	The value.	

Parameter	Description	Notes
accompanimentBookingsInfo	An array containing information about bookings with accompanying passengers	For MIR, contains information about linked MNN and MNS bookings For MNN, MNS contains information about the booking from the MIR to which the current booking is linked
number	PNR booking with accompanying passenger	
surname	Last name of the accompanying passenger	

Reservation examples

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

ВЫБРАТЬ ДРУГОЙ РЕЙС

С. Петербург LED
 * 00:45
 U6-192

1 Ч 35 МИН

Москва DME
 02:20

от 998 ₺

ПРОМО 998 ₺ ВАШИ ВОЗМОЖНОСТИ	ЭКОНОМ 2 198 ₺	ПРЕМИУМ-ЭКОНОМ 4 698 ₺	КОМФОРТ 32 968 ₺	БИЗНЕС 13 428 ₺
---	------------------------------	--------------------------------------	--------------------------------	-------------------------------

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

```
{
  "outboundFlightGroupId": 0,
  "outboundFareFamilyId": 2,
  "outboundFlightGroupHash": "07/16/2022 00:45:00-192-LED-DME",
  "outboundFareFamilyCode": "PR",
  "outboundPrice": 998.00,
  "outboundTaxes": 978.00,
  "outboundSeatsLeft": 9,
  "outboundFareAndAmountDetails": [{
    "bookingCode": "W",
    "fareBasis": "WPROW",
    "isBreakpoint": true,
    "paxType": 0,
    "paxCount": 1,
    "taxAmount": 978.00,
    "totalAmount": 998.00,
    "flightIndex": 0,
    "isSale": false,
    "ruleKey": "U6.BRAND.Y.1.PR"
  }
],
  "returnFareAndAmountDetails": [],
  "outboundBaggageAllowances": [{
    "quantity": 1,
    "weight": 0,
    "passengerType": 0
  }
],
  "returnBaggageAllowances": [],
  "totalPrice": 998.00,
  "totalDiscount": 0.0,
  "totalBaseFare": 20.00
}
```

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:

```
{
  "flightGroupId": 0,
  "flights": [{
    "flightId": 0,
    "departureAirport": "LED",
    "departureTerminal": "1",
    "arrivalAirport": "DME",
    "departureDateTime": "2022-07-16T00:45:00",
    "arrivalDateTime": "2022-07-16T02:20:00",
    "aircraft": "Airbus A321",
    "aircraftIataCode": "321",
    "flightDuration": "1:35",
    "flightNumber": "192",
    "marketingCarrier": "U6",
    "operatingCarrier": "U6",
    "hash": "07/16/2022 00:45:00-192-LED-DME"
  }
]
```

```

    }
  ],
  "flightGroupDuration": "1:35",
  "hash": "07/16/2022 00:45:00-192-LED-DME"
}

```

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

```

"flights": {
  "outboundFlights": [{
    "origin": "LED",
    "destination": "DME",
    "departureDate": "2022-07-16T00:45:00.000Z",
    "flightNumber": "192",
    "classOfService": "W"
  }
],
  "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 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:

```

{
  "fareFamilyId": 2,
  "code": "PR",
  "category": "ECOLIGHT",
  "isSubsidized": false,
  "options": [...],
  "order": 150,
  "commercialFamily": "U6ECONOMY",
  "canRefund": true,
  "hasUpsale": true,
  "isLowCost": false
}

```

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

```

"fares": {
  "outboundFareFamily": {
    "category": "ECOLIGHT",
    "code": "PR",
    "isSubsidized": false
  },
  "returnFareFamily": null,
  "fares": [{
    "fareBasis": "WPROW",
    "isBreakpoint": true,
    "passengerType": 0,
    "flightDirection": 0,
    "flightIndex": 0
  }
]
}

```

```

    },
    "sessionMarker": null
}

```

Also need to add valid passenger's data in request.

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

Direction outbound

The screenshot shows a flight selection interface with the following details:

- Header:** ВЫБРАТЬ ДРУГОЙ РЕЙС
- Route:** С. Петербург LED to Сочи AER
- Flight Details:** U6-123, 06:30 departure, 11:30 arrival, 5 ч 00 мин duration.
- Price:** от 705 ₺
- Class Selection:**
 - ПРОМО:** 705 ₺, ВАШИ ВОЗМОЖНОСТИ
 - ЭКОНОМ:** 2 705 ₺, недоступно
 - ПРЕМИУМ-ЭКОНОМ:** 5 205 ₺, недоступно
 - БИЗНЕС:** 24 200 ₺, недоступно

Direction return

The screenshot shows a flight selection interface with the following details:

- Header:** ВЫБРАТЬ ДРУГОЙ РЕЙС
- Route:** Сочи AER to С. Петербург LED
- Flight Details:** U6-524, 08:20 departure, 16:20 arrival, 8 ч 00 мин duration, 1 пересадка, U6-256.
- Price:** от 1 331 ₺
- Class Selection:**
 - ПРОМО:** 1 331 ₺, ВАШИ ВОЗМОЖНОСТИ

Recommendation:

```
{
  "outboundFlightGroupId": 0,
  "outboundFareFamilyId": 1,
  "returnFlightGroupId": 3,
  "returnFareFamilyId": 1,
  "outboundFlightGroupHash": "2022-07-13 06:30:00-123-LED-AER",
  "returnFlightGroupHash": "2022-07-20 08:20:00-524-AER-SVX,2022-07-20
15:30:00-256-SVX-LED",
  "outboundFareFamilyCode": "PR",
  "returnFareFamilyCode": "PR",
  "outboundPrice": 705.00,
  "outboundTaxes": 700.00,
  "returnPrice": 1331.00,
  "returnTaxes": 66.00,
  "outboundSeatsLeft": 9,
  "returnSeatsLeft": 9,
  "outboundFareAndAmountDetails": [{
    "bookingCode": "O",
    "fareBasis": "OPROW",
    "isBreakpoint": true,
    "paxType": 0,
    "paxCount": 1,
    "taxAmount": 700.00,
    "totalAmount": 705.00,
    "flightIndex": 0,
    "isSale": false,
    "ruleKey": "U6.BRAND.Y.1.PR"
  }
],
  "returnFareAndAmountDetails": [{
    "bookingCode": "W",
    "fareBasis": "WPROW",
    "isBreakpoint": true,
    "paxType": 0,
    "paxCount": 1,
    "taxAmount": 0.0,
    "totalAmount": 1265.00,
    "flightIndex": 0,
    "isSale": false,
    "ruleKey": "U6.BRAND.Y.1.PR"
  }, {
    "bookingCode": "N",
    "fareBasis": "N",
    "isBreakpoint": true,
    "paxType": 0,
    "paxCount": 1,
    "taxAmount": 66.00,
    "totalAmount": 66.00,
    "flightIndex": 1,
    "isSale": false
  }
],
  "outboundBaggageAllowances": [{
    "quantity": 1,
```

```

        "weight": 0,
        "passengerType": 0
    }
],
"returnBaggageAllowances": [{
    "quantity": 1,
    "weight": 23,
    "passengerType": 0
}
],
"totalPrice": 2036.00,
"totalDiscount": 0.0,
"totalBaseFare": 1270.00
}

```

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.

Flight outbound:

```

{
  "flightGroupId": 0,
  "flights": [{
    "flightId": 0,
    "departureAirport": "LED",
    "departureTerminal": "1",
    "arrivalAirport": "AER",
    "departureDateTime": "2022-07-13T06:30:00",
    "arrivalDateTime": "2022-07-13T11:30:00",
    "aircraft": "Airbus A321",
    "aircraftIataCode": "321",
    "flightDuration": "5:00",
    "flightNumber": "123",
    "marketingCarrier": "U6",
    "operatingCarrier": "U6",
    "hash": "2022-07-13 06:30:00-123-LED-AER"
  }
],
  "flightGroupDuration": "5:00",
  "hash": "2022-07-13 06:30:00-123-LED-AER"
}

```

Flights return:

```

{
  "flightGroupId": 3,
  "flights": [{
    "flightId": 0,
    "departureAirport": "AER",
    "arrivalAirport": "SVX",
    "departureDateTime": "2022-07-20T08:20:00",
    "arrivalDateTime": "2022-07-20T13:50:00",
    "aircraft": "Airbus A320",
  }
]
}

```

```

    "aircraftIataCode": "320",
    "flightDuration": "3:30",
    "stopDuration": "1:40",
    "flightNumber": "524",
    "marketingCarrier": "U6",
    "operatingCarrier": "U6",
    "hash": "2022-07-20 08:20:00-524-AER-SVX"
  }, {
    "flightId": 1,
    "departureAirport": "SVX",
    "arrivalAirport": "LED",
    "arrivalTerminal": "1",
    "departureDateTime": "2022-07-20T15:30:00",
    "arrivalDateTime": "2022-07-20T16:20:00",
    "aircraft": "Airbus A320",
    "aircraftIataCode": "320",
    "flightDuration": "2:50",
    "flightNumber": "256",
    "marketingCarrier": "U6",
    "operatingCarrier": "U6",
    "hash": "2022-07-20 15:30:00-256-SVX-LED"
  }
],
"flightGroupDuration": "8:00",
"hash": "2022-07-20 08:20:00-524-AER-SVX,2022-07-20 15:30:00-256-SVX-LED"
}

```

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

```

{
  "flights": {
    "outboundFlights": [{
      "origin": "LED",
      "destination": "AER",
      "departureDate": "2022-07-13T06:30:00.000Z",
      "flightNumber": "123",
      "classOfService": "O"
    }
  ],
  "returnFlights": [{
    "origin": "AER",
    "destination": "SVX",
    "departureDate": "2022-07-20T08:20:00.000Z",
    "flightNumber": "524",
    "classOfService": "W"
  }, {
    "origin": "SVX",
    "destination": "LED",
    "departureDate": "2022-07-20T15:30:00.000Z",
    "flightNumber": "256",
    "classOfService": "N"
  }
],
  "adults": 1,
  "children": 0,

```

```

    "infants": 0,

    "infantsWithSeats": 0,
    "pensioners": 0,
    "youths": 0,
    "unaccompaniedMinors": 0,
    "strAdults": 0,
    "jnnChildren": 0,
    "origin": "LED",
    "jcbYouths": 0,
    "gcfAdults": 0,
    "destination": "AER",
    "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": "PR",
    "isSubsidized": false
  },
  "returnFareFamily": {
    "category": "ECOLIGHT",
    "code": "PR",
    "isSubsidized": false
  },
  "fares": [{
    "fareBasis": "OPROW",
    "isBreakpoint": true,
    "passengerType": 0,
    "flightDirection": 0,
    "flightIndex": 0
  }, {
    "fareBasis": "WPROW",
    "isBreakpoint": true,
    "passengerType": 0,
    "flightDirection": 1,
    "flightIndex": 0
  }, {
    "fareBasis": "N",
    "isBreakpoint": true,
    "passengerType": 0,
    "flightDirection": 1,

```

```

        "flightIndex": 1
    }
],
"sessionMarker": null
}

```

The fareBasis values are taken from the recommendation.

Fare 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
}

```

GET reservation method

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

Request format /api/apiVersion/reservation

Request headers: X-Api-Key - API key

Examples of requests

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

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

Request parameters

Parameter	Description	Notes
pnrNumber	PNR number for which information is needed	It can be transmitted both in Latin and in Cyrillic.
ticketNumber	Issued ticket number. The ticket number must be 13 digits Example: 2622438214047 or 2622438214046.	
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.	The surname is transmitted in Latin or Cyrillic, depending on how it was indicated when creating the booking.

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": [{
      "flightReferences": [
        "string"
      ],
      "number": string,
      "passengerReference": string,
      "isInfantTicket": boolean,
      "issueDate": datetime,
      "isVoidAvailable": boolean,
      "voidTimeLimitUtc": "datetime"
    }
  ],
  "outboundFareFamily": fareFamilyObject,
  "returnFareFamily": fareFamilyObject,
  "separateFareFamily": fareFamilyObject,
  "tsmData": [{
    "passengerReference": string,
```

```

    "price": number,
    "referenceType": string,
    "referenceNumber": string
  }
],
  "outboundBonus": {
    "bonus": "string",
    "replacedBonus": "string"
  },
  "returnBonus": {
    "bonus": "string",
    "replacedBonus": "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,
    "flightReferences": [
      "string"
    ],
    "passengerReference": string,
    "isInfantFare": boolean,
    "totalPrice": number,
    "basePrice": number,
    "separatePrice": number,
    "outboundDiscount": integer,
    "returnDiscount": integer,
    "separateDiscount": integer,
    "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": {
            couponReference: boolean
        },
        "couponStatusesBySegment": {
            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,
        "separateFareFamilies": {
            "code": string
        },
        "separateFlights": [fullReservationFlight],
        "separateDuration": string,
        "origin": string,
        "destination": string,
        "departureDate": datetime,
        "returnDate": datetime,
        "outboundBaggageAllowances": [{
            "quantity": integer,
            "weight": integer,
            "passengerType": integer,

```

```

        "fareCode": "string"
    }
],
"returnBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer,
    "fareCode": "string"
}],
"separateBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer,
    "fareCode": "string"
}],
},
"number": string,
"reservationDate": datetime,
"passengers": [passengerObject],
"services": [{
    "code": string,
    "status": string,
    "text": string,
    "petType": string,
    "subType": string,
    "category": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string,
    "packReferenceNumber": string,
    "price": number,
    "currency": string,
    "issueDate": datetime,
    "issuedByThirdPartyOffice": boolean,
    "isIssuedByRefundableOffice": boolean
}],
"seats": [{
    "number": integer,
    "title": string,
    "code": string,
    "status": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string,
    "packReferenceNumber": string,
    "issueDate": datetime,
    "issuedByThirdPartyOffice": boolean,
    "isIssuedByRefundableOffice": boolean
}],
"insurance": {
    "number": string,
    "code": string,
    "status": string,

```

```

    "referenceNumber": string,
    "packReferenceNumber": string,
    "issueDate": datetime,
    "issuedByThirdPartyOffice": boolean,
    "isIssuedByRefundableOffice": boolean
  },
  "partnerProducts": [{
    "partnerId": integer,
    "partnerType": integer,
    "code": string,
    "amount": number,
    "currency": string,
    "passengerReference": string,
    "isInfantProduct": boolean,
    "isIssuedByRefundableOffice": 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,
  "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,
  "statuses": [
    string
  ],
  "codeshare": string,
  "codeshareFlightNumber": string
  "isCancelledByAirline": boolean,
  "isCheckinWithoutPseudo": boolean
}
```

passenger object

```
{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
  "relatedAdultReferenceNumber": string,
  "relatedInfantReferenceNumber": string,
  "relatedChildrenReferenceNumbers": [
    "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,
"accompanimentBookingsInfo":[
    {
        "number": string,
        "surname": string
    }
]
}

```

bonusObject object

```

{
    "bonus": "string",
    "replacedBonus": "string"
}

```

Response parameters

Parameter	Description	Notes
success	Sign of the successful request execution If the errors array is empty, the value of field will be true	
data	The data returned by the request	
tickets	Information about issued tickets	
flightReferences	Flight segment	
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.	
isVoidAvailable	Sign of ticket void availability	
voidTimeLimitUtc	Time in UTC until void is possible	

Parameter	Description	Notes
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	
separateFareFamilies	Service field with information about the selected fare for a flight without reference to a segment	
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	
outboundBonus	Bonuses for flying there	
bonus	Current number of bonuses awarded for the flight	
replacedBonus	Outdated number of bonuses awarded per flight	
returnBonus	Bonuses for return flights	
bonus	Current number of bonuses awarded for the flight	
replacedBonus	Outdated number of bonuses awarded per flight	
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	Notes
	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	
	flightReferences	Flight segment	
	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)	
	separatePrice	Price per segment without direction	
	outboundDiscount	An amount of discount for "There" route	
	returnDiscount	An amount of discount for "From" route	
	separateDiscount	The amount of discount applied to segments without direction	

Parameter		Description	Notes
	priceHidden	The sign if the rate amount should be hidden for the client. Will be "true" for IT-tariffs.	
	taxDetailsInformation	Services information about tariffs and fees.	
	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	
	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.	

Parameter	Description	Notes
couponStatuses	Statuses of flight coupons.	
couponStatusesBySegment	Statuses of flight coupons for flight segments.	
engineType	The system in which the reservation was created. Can take values: 0 - Unknown, 2 - Tais, 4 - Sirena	
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	
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	

Parameter		Description	Notes
	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	
	separateFareFamilies	Service field with information about the selected fare for a flight without reference to a segment	
	separateFlights	An array of fullReservationFlight objects with detailed information about flights without reference to a segment	
	separateDuration	Total flight duration without reference to segment	
	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	
	returnBaggageAllowances	An array of objects containing detailed information about baggage allowances for the “return” route	
	separateBaggageAllowances	An array of objects containing detailed information about baggage allowances for segments not tied to a destination	
	containsInternationalFlights	A sign of one airports of booking isn't in Russia	
	number	Number of PNR	
	reservationDate	Date and time of booking creation	
	passengers	An array of passengerData objects with passenger information	

Parameter	Description	Notes
services	Array of objects with information about additional services added to the booking	
code	Additional service code Possible values: <u>dictionary of additional codes services</u>	
status	Service adding status. Example: HK - service added successfully	
text	Text in ssr format for sending pet information. Example: DOG 1PC 8KG 45X35X25	
petType	Pet type Possible values: DOG, CAT, PARROT, CANARY	
subType	Sport Equipment type Possible values: BIKE, SKIS, SURFING, FISHING, GOLF, HOCKEY, SCOOTER	
category	Additional service category Possible values: <u>dictionary of additional codes services</u>	
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	
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	
currency	Additional service currency	
issueDate	Date and time of issuing the service.	

Parameter		Description	Notes
	issuedByThirdPartyOffice	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued.	
	isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office.	
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	
	finalPrice	Final price	
	price	Seat's price	
	currency	Seat's pricing currency	
	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.	
	isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office.	
insurance		Object with information about the insurance added to the booking	In developing
	number	Insurance document number	

Parameter		Description	Notes
	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.	
	isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office.	
partnerProducts		Array of objects with information about the partner services added to the booking	
	partnerId	Unique ID of the partner in the IBE engine.	
	partnerType	Unique code of partner's type	
	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	
	isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office.	

Parameter	Description	Notes
hasSpecialConditions	The sign if one of the following services has been added to the booking : PETC, CBBG, WEAP, UMNR, 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	
description	Error description	
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	

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.

Parameter	Description
code	<p>Tariff option code</p> <p>For example:</p> <p>upTo5bonuses</p> <p>refundExchangeProhibited</p> <p>mealsOnBoard</p> <p>and etc.</p> <p>Used for the website of the booking engine.</p>
category	<p>Category code for option</p> <p>For example:</p> <p>bonus, ffServices and etc.</p> <p>Used for the website of the booking engine.</p>
order	Order for option output
showInPopup	Whether to show an option in the "your options" pop-up window.
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	Notes
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	

Parameter	Description	Notes
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	
statuses	Array with information about segment statuses	
codeshare	The full name of the airline that operates the codeshare flight. For example: Czech Airlines for operating Carrier = OK For operatingCarrier = U6, this field is not filled in	

Parameter	Description	Notes
codeshareFlightNumber	The flight number of the airline that operates the codeshare flight.	
isCancelledByAirline	Sign of cancellation of flights by the airline	
isCheckinWithoutPseudo	Sign of hiding a pseudo-booking at the checkin	

outboundBaggageAllowances, returnBaggageAllowances, separateBaggageAllowances

Parameter	Description
quantity	count of baggage
weight	weight of baggage
passengerType	passenger type
fareCode	fare family for which baggage is returned

passengerData

Parameter	Description	Note
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	
relatedAdultReferenceNumber	Reference to the adult passenger in the booking with whom the child or infant flies (it returns only for children and infants)	
relatedInfantReferenceNumber	Reference to the infant passenger in the booking with whom the adult flies (it returns only for adult)	
relatedChildrenReferenceNumbers	An array of links to child passengers which are travelling with an Adult (the array will be received only for an Adult with a children in the same booking)	
firstName	Passenger's name	
middleName	Passenger's patronymic (may be absent)	
surname	Passenger's surname	

Parameter	Description	Note
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	Three-letter code of country of citizenship Possible values: dictionary of countries	
documentNumber	Masked number of document of passenger. Format: <ul style="list-style-type: none"> • passport: 53X67, • birth certificate: VIX43. 	
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"	
email	Email	
phone	Phone	

Parameter	Description	Note
accompanimentBookingsInfo	An array containing information about bookings with accompanying passengers	For MIR, contains information about linked MNN and MNS bookings For MNN, MNS contains information about the booking from the MIR to which the current booking is linked
number	PNR booking with accompanying passenger	
surname	Last name of the accompanying passenger	

DELETE reservation method

Description: the method allows to delete created unpaid reservation, after successful deletion of booking, all flight segments in the response of GET api/reservation in journey array have status XX.

Request format /api/apiVersion/reservation

Request headers

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

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 successful request execution If the errors array is empty, the value of field 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 Sirena answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
code	Sirena error code
text	Sirena error text

Examples of unsuccessful requests and responses

An attempt 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 v2.3:

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

In the GET reservation service response, the isCheckinWithoutPseudo parameter is added to the outboundFlights[], returnFlights[], separateFlights[] arrays.

The isIssuedByRefundableOffice parameter is added to the GET reservation service response and can be displayed in the tickets[]; ticketsWithoutFlightRefs:[]; informationalServices:[]; seats[]; insurances[]; services:[]; partnerProducts[] arrays.

The currency parameter is added to the GET reservation response and can be displayed in ticketFares[]; informationalServices arrays[]; seats[]; insurances[]; services[]; partnerProducts[];

Parameters isVoidAvailable and voidTimeLimitUtc have been added to the GET reservation response in case of request by ticket number - with information about void availability and its term

Response format

```
{
  "data": {
    "isGroupReservation": boolean,
    "isCompanyGroupReservation": boolean,
    "summToPayForExchange": double,
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string
      },
      "outboundDuration": string,
      "returnFlights": [fullReservationFlight],
      "returnDuration": string,
      "separateFareFamilies": {
        "code": string
      },
      "separateFlights": [fullReservationFlight],
      "separateDuration": string,
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    },
    "baseFarePricesBySegment": [
      {
        "flightReference": string,
        "passengerReference": string,
        "totalPrice": string,
        "basePrice": string,
        "discountAmount": string,

```

```

    "passengerType": integer
  }
],
  "fareBasisCodesFlightRefs" : {[fareBasisCodesFlightRefs object]},
"number": string,
"reservationDate": datetime,
  "bookingTimeLimit": datetime,
"passengers": [{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
    "relatedAdultReferenceNumber": string,
    "relatedInfantReferenceNumber": string,
    "relatedChildrenReferenceNumbers": [
      "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,
    "boardingPassRestrictedBySegment": {
      flightReference: boolean
    }
  }
],
"services": array,
"seats": array,
"insurances": [{
  "number": string,
  "passengerType": integer,
  "code": string,
  "status": string,
    "passengerReference": string,
    "price": double,
    "currency": string,
    "isIssuedByRefundableOffice": boolean
}]
],
"relatedPnrNumbers": [
  string
],
},
"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,
  "statuses": [
    string
  ],
  "codeshare": string,
  "codeshareFlightNumber": string,
  "isCancelledByAirline": boolean,
  "isCheckinWithoutPseudo": boolean
}

```

fareBasisCodesFlightRefs object

```

{
  "fareBasis": string,

```

```

"flightReferenceNumber": string,
"passengerReferenceNumber": string,
"expirationUtc": datetime
}

```

Response parameters (short format)

Parameter (returns only if rebooking in process)	Description	Notes
success	Sign of successful request execution If errors array is empty, the value of field will be 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.	
<i>insurances</i>	Array of objects with information about selected insurances.	In developing
partnerProducts	Array of objects with information about partner's services.	
summToPayForExchange	The amount to be paid for ticket exchange (parameter is returned only when rebook process is in progress)	
journey	Information about flights for present booking.	
isAgentReservation	Agent booking sign to disable the booking management module	

flightsWithOpenDateCount	Number of flights with open dates	
relatedPnrNumbers	An array of related PNRs (obtained as a result of splitting a reservation as part of a partial exchange).	
isPayForbiddenForAgency	Sign that payment is not available for an unpaid booking	In developing
baseFarePricesBySegment	An array of object with information about base fare prices by segments of flights	
informationalServices	An array of objects with information about informational services	
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
currency	Currency used for payment
isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office.

passengerData structure description

Parameter	Description
boardingPassRestrictedBySegment	Restrictions on obtaining a boarding pass by segment

POST api/Reservation method

Description: booking creating, adding an information of flights, passengers, fares.

Request format: /api/apiVersion/Reservation

Request body

It has totally the same realisation as for post api/Reservation v2.2

Response format

It has the same realisation for post api/Reservation v2.2. A **data** object is extended by the baseFarePricesBySegment [] array and InformationalServices[] array.

Changes in version 2.3:

In the GET reservation service response, the isCheckinWithoutPseudo parameter is added to the outboundFlights[], returnFlights[], separateFlights[] arrays.

The isIssuedByRefundableOffice parameter is added to the GET reservation service response and can be displayed in the tickets[]; ticketsWithoutFlightRefs:[]; informationalServices:[]; seats[]; insurances[]; services:[]; partnerProducts[] arrays.

In the POST reservation service response, the isExchangePermitted parameter is added to the data{} object.

In the request body and the response of the POST reservation service, the subsidiesType parameter is added to the data{} object.

baseFarePricesBySegment

```
{
  "data": {
    "isGroupReservation": boolean,
    "isCompanyGroupReservation": boolean,
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string
      },
      "outboundDuration": string,
      "returnFlights": [fullReservationFlight],
      "returnDuration": string,
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    },
    "baseFarePricesBySegment": [
      {
        "flightReference": string,
        "passengerReference": string,
        "totalPrice": string,

```

```

        "basePrice": string,
        "discountAmount": string,
        "passengerType": integer
    }
],
    "fareBasisCodesFlightRefs" : {[fareBasisCodesFlightRefs object]},
"number": string,
"reservationDate": datetime,
    "bookingTimeLimit": datetime,
"passengers": [{
    "id": integer,
    "hasInfant": boolean,
    "referenceNumber": string,
        "relatedAdultReferenceNumber": string,
        "relatedInfantReferenceNumber": string,
        "relatedChildrenReferenceNumbers": [
            "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,
        "currency": string,
        "isIssuedByRefundableOffice": boolean
    }
],
"isAgentReservation": boolean,
    "isPayForbiddenForAgency": boolean,
    "isExchangePermitted": boolean,
"partnerProducts": array,
    "flightsWithOpenDateCount": integer
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,

```

```

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

```

Response parameters (at the short form, only extension)

Parameter	Description	Notes
data	A data which were received by the request	
baseFarePricesBySegment	An array of objects with information about base fare prices by segments of flights	

InformationalServices

```

{
  "code": string,
  "status": string,
  "text": string,
  "petType": string,
  "category": string,
  "passengerReference": string,
  "flightReference": string,
  "referenceNumber": string,
  "price": number,
  "currency": string,
  "issueDate": datetime,
  "issuedByThirdPartyOffice": boolean,
  "isIssuedByRefundableOffice": boolean
}

```

Response parameters

Parameter	Description	Notes
code	Additional service code Possible values: reference codes of the add. services	
status	The status of adding the service. Example: HD service successfully added	
text	Text in ssr format for sending information. Example: DOG 1PC 8KG 45X35X25	

Parameter	Description	Notes
petType	Type of pet. Possible values: DOG, CAT, PARROT, CANARY	
category	Category of additional service Possible values: reference codes of the add. services	
passengerReference	Link to the passenger for whom an additional service has been added	
flightReference	Link to the segment on which the service operates	
referenceNumber	SSR number. There is an unique SSR number for each service in PNR	
price	The cost of an additional service	
currency	Currency used for payment	
issueDate	Date and time of service release.	
issuedByThirdPartyOffice	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued	
isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office.	

6_6_api/services, api/catalogue: auxiliary services management service

Auxiliary services management service v2.2

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	Sirena: svc_prices, svc_add CRM: getContactMarkup, EmdPromoCode

GET catalogue method

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

Request format /api/apiVersion/catalogue

Request headers

Parameter	Description
X-Session	Session key
X-Api-Key	API key

Request parameters

Parameter	Description
currency	Current that is used for pricing services

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]
}
],
"partnerProducts": [{
  "partnerId": integer,
  "partnerType": integer,
  "code": string,
  "amount": number,
  "currency": string,
  "bookingStatus": integer,
  "passengerReference": string,
  "isInfantProduct": boolean,
  "bookingId": string,
  "internalError": integer,
  "internalErrorMessage": string
}
],
"portions": [{
  "type": string,
  "number": string,
  "flightReferenceNumbers": [string]
}
],
"serviceCategories": [{
  "code": string,

```

```

        "category": string,
        "subCategory": string,
        "order": integer,
        "fullDirection": boolean,
        "onlyForRussianCitizens": true
    }
],
    "serviceOptions": [{
        "code": string,
        "preselect": boolean,
        "category": string,
        "priority": integer,
        "flightReferences": [string],
        "passengerReferences": [string]
    }
],
    "crmEmdPromoCodes": [{
        "serviceType": string,
        "promo": string,
        "promoDescription": string,
        "discountLevel": integer,
        "expireDate": datetime
    }
]
    "moneyWallCoefficient": string
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string,
        "additionalInfo": {
            "ssrType": string,
            "info": string
        }
    }
]
}
],
"success": boolean
}

```

Response parameters

Parameter	Description	Note
success	Sign of successful request execution If errors array is empty, the value of field will be true	
data	Returned data	

	services	Array of objects with information about airline's available services	
	code	Service code Possible values: EN 8 1 Attachments: Dictionary	
	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	
	packs	An array with available service packages	In developing
	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	Array of objects with information about which services should be automatically pre-selected in IBE frontend.	
	category	Service category	

	code	Service code Possible values: Attachment: dictionaries	
	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.	
	passengerReferences	Array of objects with information about references to passengers for which the service is preselected.	
	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.	In developing
	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	

	serviceCategories	Array of objects with information about services and order of its display in website	
	code	Service code	
	category	Service category	
	subCategory	Service subcategory At the present time it returns only for meals, possible values: vegetarian, special, snacks, hotmeal, childmeal, breakfast, dessert	
	order	Display index in auxiliary services page. 1 - page top (A deprecated option that is no longer used. The order of arrangement of additional services is determined on the side of the booking engine)	
	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. Alfainsurance – 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.	
	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	In developing
		type	Service field	
		number	Service field	
		flightReferenceNumbers	Service field	
	errors		Information about errors appeared during query execution.	
		ibeCode	Error code	
		description	Code description	
	messages		Array of objects, containing errors from Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode	
		code	Sirena error code	
		text	Sirena error text	
	additionalInfo		An array with error's additional information	
		ssrType	SSR type	
		info	Additional information	

POST services method

Description: allows you to add services to the booking.

Request format: /api/apiVersion/services

Request headers

Parameter	Description
X-Session	Session key
X-Api-Key	API key

Request body:

```
{
  "seats": [{
    "number": integer,
    "title": string,
    "code": string,
    "text": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": "string"
  }
],
  "services": [{
    "code": string,
    "text": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string
  }
],
  "currency": string
}
```

Request parameters:

Parameters	Description
services	Array of objects with information about added services
code	Additional service code Possible values: attachments: dictionaries (categories different from insurance, seat)

text	Animal type Possible values: attachments: dictionaries
passengerReference	Passenger reference
flightReference	Flight reference
referenceNumber	Paid services reference
seats	Array of objects with information about added seats
number	Row number of selected seat
title	Seat
code	Seat code (Attachments: dictionaries)
text	Text
referenceNumber	Paid seat reference
passengerReference	Passenger reference
flightReference	Flight reference
currency	Currency

Examples of requests with the services of the Airline

Add one extra baggage:

```
{
  "seats": [],
  "services": [
    {
      "code": "ABAG",
      "passengerReference": "12",
      "flightReference": "12"
    }
  ],
  "currency": "RUB"
}
```

Add seat:

```
{
  "seats": [
    {
      "number": 7,
      "title": "A",
      "passengerReference": "12",
      "flightReference": "12"
    }
  ]
}
```

```

    ],
    "services": [],
    "currency": "RUB"
}

```

Add pet:

```

{
  "seats": [],
  "services": [
    {
      "code": "PETC",
      "text": "DOG",
      "passengerReference": "12",
      "flightReference": "12"
    }
  ],
  "currency": "RUB"
}

```

Logic of adding/removing services to a booking

IBE always updates the booking for additional services based on the parameters in the request.

Example:

Add baggage:

```

{
  "seats": [],
  "services": [{
    "code": "ABAG",
    "passengerReference": "12",
    "flightReference": "12"
  }
  ],
  "currency": "RUB"
}

```

To remove baggage from a booking, send the following request:

```

{
  "seats": [],
  "services": [],
  "currency": "RUB"
}

```

This logic is implemented for all types of services

Response format

The format of the response in terms of adding services is the same as the response of the GET api/reservation method.

Method GET api/catalogue/GetServiceCategoriesOrder

Description: gets information about types and subtypes of additional services and the order of it's displaying

Request format /api/apiVersion/catalogue/GetServiceCategoriesOrder

Request headers: X-Api-Key - API key

Response body:

The sortedInCategories [] array is added with information about the category of additional services and subspecies related to an additional service, as well as the order in which the subspecies are displayed inside the block

```
{
  "data": {
    "sortedServiceGroups": [
      {
        "condition": 0,
        "orderType": 1,
        "sortedServiceCategories": []
      },
    ],
    "sortedInCategories": [
      {
        "category": "string",
        "serviceOrder": [
          {
            "code": "string",
            "order": integer
          },
        ]
      }
    ]
  },
  "errors": [],
  "success": true
}
```

Response parameters

Parameter		Description
success		Sign of successful request execution. If the errors array is empty, the value of field will be true
data		Data returned by request
	sortedServiceGroups	Array of objects with information about additional services
	condition	Service field
	orderType	Order type
	sortedServiceCategories	Array of with categories of additional services Example: "seat", "luggage", "meal". See: Attachment: dictionaries
	sortedInCategories	An array of objects with information about additional services, subspecies and their display
	category	Additional services category Example: "seat", "luggage", "meal". See: Attachment: dictionaries
	serviceOrder	An array of objects with information about the service code and the order of display on the page
	code	Additional service subspecies code
	order	The additional service subspecies order of display related to other subspecies located in the same serviceOrder array
errors		Information about errors appeared during request

Service for obtaining information about additional services v2.3

Changes in GET method response api/catalogue version 2.3:

Version 2.2	Version 2.3
In the services.flightReferences.referenceNumber array of type string changed to services.flightReferences.referenceNumbers array of type string.	

<pre>"services": [{ "code": string, "flightReferences": [{ "referenceNumber": string, "passengerPrices": [{ "referenceNumber": string, "amount": number }] }]]</pre>	<pre>"services": [{ "code": string, "flightReferences": [{ "referenceNumbers": [string], "passengerPrices": [{ "referenceNumber": string, "amount": number }] }]]</pre>
--	---

In the packs.flightReferences.referenceNumber array of type string changed to packs.flightReferences.referenceNumbers array of type string.

<pre>"packs": [{ "includedSeats": [string], "excludedSeats": [string], "includedServiceCode": string, "code": string, "flightReferences": [{ "referenceNumber": string, "passengerPrices": [{ "referenceNumber": string, "amount": number }] }]]</pre>	<pre>"packs": [{ "includedSeats": [string], "excludedSeats": [string], "includedServiceCode": string, "code": string, "flightReferences": [{ "referenceNumbers": [string], "passengerPrices": [{ "referenceNumber": string, "amount": number }] }]]</pre>
--	---

Changed the seatServices.flightReferences.referenceNumber array of type string to seatServices.flightReferences.referenceNumbers array of type string.

<pre> "seatServices": [{ "seatCharacteristics": [string], "code": string, "flightReferences": [{ "referenceNumber": string, "passengerPrices": [{ "referenceNumber": string, "amount": number }] }] </pre>	<pre> "seatServices": [{ "seatCharacteristics": [string], "code": string, "flightReferences": [{ "referenceNumbers": [string], "passengerPrices": [{ "referenceNumber": string, "amount": number }] }] </pre>
--	---

In version 2.3, the partnerProducts array has been removed.

In the array services[].flightReferences[] added information about available quotas for returned service (this array is returned only for service with the code = "PETC")

<pre> "services": [{ "code": string, "flightReferences": [{ "referenceNumber": string, "passengerPrices": [{ "referenceNumber": string, "amount": number }] }] </pre>	<pre> "services": [{ "code": string, "flightReferences": [{ "referenceNumbers": [string], "passengerPrices": [{ "referenceNumber": string, "amount": number }], "quotas": [{ "referenceNumber": string, "quantity": number] }] </pre>
---	--

At the v2.3 version in an array as serviceCategoryLabels[] receiving a data of additional option with Labels for appearing at the Front also the array has a data of label types.

	<pre>"services": [{ "serviceCategoryLabels": [{ "categoryName": string, "labelTitles": [string] }] }</pre>
<p>In version 2.3, the serviceOptionLabels[] array displays the data of additional services, information of Labels to display, as well as the order of the Labels</p>	
	<pre>"services": [{ "serviceOptionLabels": [{ "code": string, "labels": [{ "labelTitle": string, "priority": integer }] }]</pre>
<p>At the v2.3 version in a sportEquipmentServiceConfiguration[] array receiving a data of able options for Sport equipment transportation option. it is receiving for "Promo" fare.</p>	
	<pre>"data": { "services": [{ "sportEquipmentServiceConfiguration": [{ "subTypes": [string], "flightReferences": [string] }] } }</pre>

Parameters of the serviceOptionLabels array[]

Parameter	Description
services	An array of objects with information about additional services, Label titles and their display on the page
serviceOptionLabels	An array of objects with additional service codes and Label information
code	Additional service code
labels	An array of objects with information about the Label's title and their location
labelTitle	The specific name/title of Label

priority	The priority of displaying Labels next to the service they belong to, where 0 is the top position above other labels
----------	--

Parameters of sportEquipmentServiceConfiguration[] array

Parameter	Description
subTypes	The array of strings with a list of able Sport equipment transportation options
flightReferences	The array of strings with a list of flight segments that are able to add the option

Examples of failed requests and responses

Request	Response
<i>Error while adding airline service (invalid service code)</i>	
<pre>{ "seats": [], "insurance": null, "partnerProducts": [], "services": [{ "code": "ABBB", "passengerReference": "12", "flightReference": "12" }] }</pre>	<pre>"errors": { "services": [{ "code": "ABBB", "passengerReference": "12", "flightReference": "12", "internalError": 1001, "internalErrorMessage": "Invalid passenger or flight reference." }] }</pre>
<i>Error when adding airline service (invalid reference to passenger)</i>	
<pre>{ "seats": [], "insurance": null, "partnerProducts": [], "services": [{ "code": "ABAG", "passengerReference": "5", "flightReference": "12" }] }</pre>	<pre>"errors": { "services": [{ "code": "ABAG", "passengerReference": "5", "flightReference": "12", "internalError": 1001, "internalErrorMessage": "Invalid passenger or flight reference." }] }</pre>

Method for adding services to a booking: POST api/services/add

Description: a method for adding new services that allows not passing in the parameters already existing services in the order

Request format: api/apiVersion/services/add

Request headers: X-Api-Key - API key, X-Session - session key

Request body

```
{
  "services": [{
    "code": "string",
    "text": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string"
  }
],
  "seats": [{
    "number": "integer",
    "title": "string",
    "isCheckinSeat": boolean,
    "code": "string",
    "text": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string"
  }
]
}
```

Request parameters:

Parameter	Description
currency	Pricing currency. For example, RUB
services	Array of objects with information about added services
code	Additional service code Possible values: Attachments: dictionaries (categories different from seat)
text	Animal type
passengerReference	Link to the passenger for whom the service is being added
flightReference	Link to the segment for which the service is being added
referenceNumber	Number of reference
seats	Array with objects about seats

Parameter	Description
number	Row number for selected seat
title	Seat
code	Seat code
text	Text
passengerReference	Passenger reference for adding seat
flightReference	Flight reference for adding seat
referenceNumber	Number of reference

Valid request example:

Adding of service "Hot breakfast" (TCML), first additional bag (ABAG) and seat select (SEAT):

```
{
  "currency": "RUB",
  "services": [
    {
      "code": "TCML",
      "flightReference": "12",
      "passengerReference": "12"
    },
    {
      "code": "ABAG",
      "flightReference": "12",
      "passengerReference": "12"
    }
  ],
  "seats": [
    {
      "code": "SEAT",
      "flightReference": "12",
      "passengerReference": "12",
      "number": 5,
      "title": "A",
      "isCheckinSeat": false
    }
  ]
}
```

Optimized method for removing services from booking api/services/removeByReferences

Description: a method for removing services from an order (more optimized analog than POST api/services/remove), provided that no ticket has been issued for the service and the flight segment for which the service was added has not yet departed.

Note: in the body request need to pass only references to those services that need to be deleted.

Request format: api/apiVersion/services/removeByReferences

Request headers: X-Api-Key - API key

Request body:

```
{
  "services": [{
    "referenceNumber": string
  }],
  "seats": [{
    "referenceNumber": string
  }]
}
```

Request body parameters:

Parameter	Description
services	The array with the links to services which need to delete from reservation
referenceNumber	Link to service which need to delete. Value is possible to get from the response of GET api/reservation, parameter data.services[].referenceNumber
seats	The array with the links to seats which need to delete from reservation
referenceNumber	Link to seat which need to delete. Value is possible to get from the response of GET api/reservation, parameter data.seats[].referenceNumber

Response format:

Response format regarding the removal of services and seats is the same with the response of GET api/reservation:

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

```

    {
      "aircraft": "string",
      "aircraftIataCode": "string",
      "referenceNumber": "string",
      "origin": "string",
      "destination": "string",
      "departureDate": "datetime",
      "arrivalDateUtc": "datetime",
      "departureDateUtc": "datetime",
      "arrivalDate": "datetime",
      "flightNumber": "string",
      "classOfService": "string",
      "commercialFamily": "string",
      "operatingCarrier": "string",
      "marketingCarrier": "string",
      "flightDuration": "string",
      "techStops": [],
      "hash": "string",
      "statuses": [
        "string"
      ],
      "isCancelledByAirline": boolean,
      "isCheckinWithoutPseudo": boolean
    },
    "separateFlights": [],
    "outboundFareFamily": {},
    "separateFareFamilies": [],
    "outboundDuration": "string",
    "separateDuration": "string",
    "origin": "string",
    "destination": "string",
    "departureDate": "datetime",
    "outboundBaggageAllowances": [],
    "returnBaggageAllowances": [],
    "separateBaggageAllowances": [],
    "containsInternationalFlights": boolean
  "number": "string",
  "passengers": [],
  "services": [],
  "seats": [],
  "partnerProducts": [],
  "hasSpecialConditions": boolean
"errors": [],
"success": true
}

```

Method POST `api/services/favorite`

Description: allows you to get your favorite services for passengers

Request format /api/apiVersion/services/favorite

Request headers: X-Api-Key - API key

Request body example

```
{
  "docNumbers": [
    string
  ],
  "emdCodes": [
    string
  ],
  "months": integer
}
```

Request body parameters

Parameter	Description
docNumbers	Passengers' document numbers
emdCodes	Service codes
months	Number of months to search

Response body

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

Response parameter

Parameter	Description
data	The data returned by the request
docNum	Document Number
emdCode	Service code
value	Value

Parameter	Description
errors	Information about errors that occurred during the operation of the service.
ibeCode	Error code
description	Error description
messages	An array of objects containing the errors from Sirena's response. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Sirena error code
text	Sirena error text
success	Sign of the successful request execution If the errors array is empty, the value of field will be true

Request body example

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

6_7_api/CatalogueStandalone: service for obtaining a catalog of services without PNR

API service	CatalogueStandalone
Short description	Service allows to receive information about additional services without PNR
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: pricing_variant

Method POST CatalogueStandalone

Description: gets information about catalog of additional services without PNR

Request format: api/apiVersion/CatalogueStandalone

Request headers: X-Api-Key - API key

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
    },
    "returnFareFamily": {
      "code": string,
      "category": string
    },
    "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,
        "relatedInfantWithSeatId": integer
    ],
    "currency": string
}

```

Request parameters

Parameter	Description	Note
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	Optional
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	
<i>isSubsidized</i>	Sign of a subsidized tariff.	In developin g

Parameter		Description	Note
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	
	<i>isSubsidized</i>	Sign of a subsidized tariff.	In developin g
fares		Array with information about tariffs for directions "outbound" and "return"	
	passengerType	Passenger type.	
	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	
passengers		Object with passenger information	
	id	Passenger's serial number (numbered from 1)	
	firstName	Passenger's name	
	middleName	Passenger's patronymic	

Parameter	Description	Note
surname	Passenger's surname	
passengerType	Passenger type	
birthDate	Date of birth	
sex	Passenger gender	
loyaltyCard	Loyalty card number "Wings"	In developing
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	
wheelchairType	Wheelchair type	
commentary	Comment (optional)	
hasInfant	An indication that this passenger has a baby	
nationality	Pax nationality	
relatedInfantWithSeatId	Id of the ins related to the adult	
currency	Search currency	

StandaloneInputFlight

Parameter	Description
origin	Departure airport
destination	Arrival airport

Parameter	Description
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)

Successful request example

```
{
  "flights": {
    "outboundFlights": [
      {
        "origin": "SVX",
        "destination": "DME",
        "departureDate": "2025-08-17T06:50:00",
        "arrivalDate": "2025-08-17T07:30:00",
        "flightNumber": "264",
        "classOfService": "W",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightIndex": 12
      }
    ],
    "returnFlights": null
  },
  "fares": {
    "outboundFareFamily": {
      "category": "ECOFLEX",
      "code": "EC",
      "isSubsidized": false
    },
    "returnFareFamily": null,
    "fares": [
      {
        "passengerType": 0,
        "flightIndex": 1,

```

```

        "fareBasis": "WECOW",
        "isBreakpoint": true
    }
  ],
  },
  "passengers": [
    {
      "id": 12,
      "passengerType": 0,
      "birthDate": "1996-09-10",
      "hasInfant": false,
      "nationality": "RUS"
    }
  ]
}

```

Response format

```

{
  "data": {
    "services": [{
      "code": string,
      "flightReferences": [{
        "referenceNumbers": string,
        "passengerPrices": [{
          "referenceNumber": string,
          "amount": number
        }
      ],
      "quotas": [{
        "referenceNumber": string,
        "quantity": 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"]
        }
    ],
    "sportEquipmentServiceConfiguration": [
        {
            "subTypes": ["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": true
    }
],
    "serviceOptions": [{

```

```

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

```

Response parameters

Parameter	Description	Note
success	Sign of the successful request execution If the errors array is empty, the value of field will be true	
data	The data returned by the request	
services	An array of objects with information about available airline services	
code	Service code	
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	

Parameter				Description	Note
			amount	Service cost for a specific passenger	
		quotas		An array with information about available quotas for service (at the current time it's returned only for the service with code = "PETC").	
			referenceNumber	Reference to a specific passenger for whom quota is returned.	
			quantity	Count of available quotas for service.	
	packs			An array with available service packages	In developing
		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.	
		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.	
		flightReferences		An array with information about segments for which services are available	

Parameter		Description	Note
	preselect	<p>Service field.</p> <p>If true, then the service should be automatically preselected in the UI.</p> <p><i>Note: The logic was originally added for use in the Web client.</i></p>	
	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>	In developing
	seatCharacteristics	Array of site characteristics.	
	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	

Parameter	Description	Note
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	
sportEquipmentServiceConfiguration	An array of objects that contain information about the availability of the sport equipment transport on different flight segments (available on "Promo" fare)	
subTypes	An array of strings, contains a list of sport equipment types available for transport	
flightReferences	An array of strings, contains a list of segments on which the service is available	
serviceCategories	An array of objects, contains information about services and the order in which they are displayed on the site.	
code	Service code	
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	
portions	Service field	In developing
type	Service field	
number	Service field	
flightReferenceNumbers	Service field	
moneyWallCoefficient	Coefficient for calculating the monthly installment payment	

Parameter	Description	Note
errors	Information about errors that occurred during the operation of the service.	
ibeCode	Error code	
description	Error description	
messages	An array of objects containing errors from Sirena' answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.	
code	Sirena error code	
text	Sirena error text	
additionalInfo	An array with additional error information	
ssrType	Ssr type	
info	Additional information	

Example of valid request:

```
{
  "flights": {
    "outboundFlights": [
      {
        "origin": "SVX",
        "destination": "KGD",
        "departureDate": "2022-11-28T13:15:00",
        "arrivalDate": "2022-11-28T14:20:00",
        "flightNumber": "403",
        "classOfService": "U",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightIndex": 12
      }
    ],
    "returnFlights": [
      {
        "origin": "KGD",
        "destination": "SVX",
        "departureDate": "2022-11-30T10:45:00",
        "arrivalDate": "2022-11-30T17:40:00",
        "flightNumber": "404",
        "classOfService": "U",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightIndex": 13
      }
    ]
  },
}
```

```

"fares": {
  "outboundFareFamily": {
    "category": "SUBSIDIZED",
    "code": "SOC",
    "isSubsidized": true
  },
  "returnFareFamily": {
    "category": "SUBSIDIZED",
    "code": "SOC",
    "isSubsidized": true
  },
  "fares": [
    {
      "passengerType": 1,
      "flightIndex": 1,
      "fareBasis": "UBDRTCD",
      "isBreakpoint": true
    },
    {
      "passengerType": 2,
      "flightIndex": 1,
      "fareBasis": "UBDRTZZ",
      "isBreakpoint": true
    },
    {
      "passengerType": 5,
      "flightIndex": 1,
      "fareBasis": "UBDRTCH",
      "isBreakpoint": true
    },
    {
      "passengerType": 1,
      "flightIndex": 2,
      "fareBasis": "UBDRTCD",
      "isBreakpoint": true
    },
    {
      "passengerType": 2,
      "flightIndex": 2,
      "fareBasis": "UBDRTZZ",
      "isBreakpoint": true
    },
    {
      "passengerType": 5,
      "flightIndex": 2,
      "fareBasis": "UBDRTCH",
      "isBreakpoint": true
    }
  ]
},
"passengers": [
  {
    "id": 12,
    "passengerType": 1,
    "birthDate": "1962-05-24",
    "hasInfant": false,
    "nationality": "RUS"
  }
]

```

```

    },
    {
      "id": 13,
      "passengerType": 2,
      "birthDate": "2008-10-26",
      "hasInfant": false,
      "nationality": "RUS"
    },
    {
      "id": 14,
      "passengerType": 5,
      "birthDate": "2012-07-26",
      "hasInfant": false,
      "nationality": "RUS"
    }
  ]
}

```

6_8_api/seatmap: 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	Sirena: seat_map, svc_order_view

GET seatmap method

Description: requests the seatmap in context of the created reservation

Request format: api/apiVersion/seatmap

Request headers

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request example

api/apiVersion/seatmap?referenceNumber=12

Request parameters

Parameter	Description
referenceNumber	Reference to requested flight segment

Response format

```
{
  "data": {
    "compartmentDetails": [
      {
        "number": 0,
        "seats": [
          {
            "seat": "string",
            "number": 0,
            "description": "string",
            "characteristics": [
              "string"
            ],
            "occupation": "string",
            "passengerRelatedPrice": [
              {
                "name": "string",
                "type": "string",
                "uniqueCustomerIdentifier": "string",
                "seatPrice": 0,
                "seatUnavailableReason": {
                  "isByCharacteristics": boolean,
                  "isByCheckin": boolean,
                  "isByAgent": boolean,
                  "isByRegistrationUnavailability": boolean,
                  "isByPreselects": boolean,
                  "isByRelatedPassengerUnavailability": boolean
                },
                "isSeatUnavailable": boolean,
                "hasInfant": boolean,
                "autoSeating": boolean
              }
            ],
            "classDesignator": "string"
          }
        ]
      }
    ],
    "isEmpty": boolean,
    "passengersTotalData": [
      {
        "passengerReference": "string",
        "passengerPriceLevels": [
          {
            "price": 0,
            "priceLevel": 0
          }
        ]
      }
    ]
  }
}
```

```

    ]
  }
],
"passengerSeats": [
  {
    "seatCheckinStatus": "ReservationCanceled",
    "flightReference": "string",
    "passengerReference": "string",
    "title": "string",
    "number": 0
  }
],
"preselectedSeats": [
  {
    "flightReference": "string",
    "passengerReference": "string",
    "title": "string",
    "number": 0
  }
]
},
"errors": [
  {
    "ibeCode": "string",
    "description": "string",
    "messages": [
      {
        "code": "string",
        "text": "string",
        "additionalInfo": {
          "ssrType": "string",
          "info": "string"
        }
      }
    ]
  }
]
},
],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be true
data	Returned data

	isEmpty	Flag identifying whether data is returned from sirena 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 row number
	description	Seat description 9 – central seat A – aisle seat W – window seat
	characteristics	Seat characteristics
	occupation	Sign of seat's occupation
	passengerRelatedPrice	Array of objects with information about seat prices for. Only available for seats with CH characteristics.
	name	Passenger name
	type	Passenger type
	uniqueCustomerIdentifier	Unique passenger ID
	seatPrice	Seat price
	seatUnavailableReason	Array of objects with parameters that are reason of unavailability seats
	isByCharacteristics	Seat is not available for the passenger by characteristics
	isByCheckin	The passenger has been already checked-in
	isByAgent	Another seat was chosen by Agent
	isByRegistrationUnavailability	Check-in is not available
	isByPreselects	Another seat was pre-selected earlier

			isByRelatedPassengerUnavailability	Seat is not available for the passenger because the seat is not suitable for accompanied passengers
			isSeatUnavailable	Seat availability flag
			hasInfant	Sign identifying that passenger travels with infant
			autoSeating	Sign of free seat
			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
	passengerSeats			Array with passenger seats information
			seatCheckinStatus	Seat checkin status
			flightReference	Flight segment reference
			passengerReference	Passenger reference
			title	Seat title
			number	Seat number
	preselectedSeats			Array with information about pre-added seats at the seat selection step in the booking engine
			flightReference	Flight segment reference
			passengerReference	Passenger reference
			title	Seat title
			number	Seat number
	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	Sirena error code
	text	Sirena error text

6_9_api/SeatmapStandalone: retrieve aircraft seatmap service without PNR

API service	SeatmapStandalone
Short description	Allows to receive information about seatmap scheme in aircraft without PNR
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: pricing_variant, svc_pricing CRM: getContactMarkup

POST SeatmapStandalone method

Description: retrieve information about seatmap scheme in aircraft without PNR.

Request format: api/apiVersion/SeatmapStandalone

Request headers: X-Api-Key - API key

Request body

```
{
  "passengers": [
    {
      "id": integer,
      "firstName": string,
      "middleName": string,
      "surname": string,
      "passengerType": string,
      "birthDate": datetime,
      "sex": integer,
      "loyaltyCard": string,
      "disability": {
```

```

        "isDeaf": boolean,
        "isBlind": boolean,
            "isDpna": boolean,
        "hasGuideDog": boolean,
        "assistanceType": integer,
        "wheelchairType": integer,
        "commentary": string
    },
    "hasInfant": boolean
    "nationality": string,
        "relatedInfantWithSeatId": integer
    }
],
"origin": string,
"destination": string,
"departureDate": datetime,
"flightNumber": string,
"classOfService": string,
"commercialFamily": string,
"fareCode": string,
"currency": string
}

```

Request parameters

Parameter	Description	Note
passengers	Array with data about passengers	
id	Passenger's serial number (numbered from 1)	
firstName	Passenger's name	
middleName	Passenger's patronymic (may be absent)	
surname	Passenger's surname	
passengerType	Passenger type	
birthDate	Date of birth. Affects the availability of emergency row seats	
sex	Passenger gender Possible values: see dictionary "gender of passengers"	
loyaltyCard	Number of "Wings" loyalty card (if exists)	In developing
disability	Information about people with disabilities	
isDeaf	Hearing impairment	
isBlind	Visual impairment	

Parameter	Description	Note
isDpna	Intellectual disability	
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)	
hasInfant	A sign that this passenger has a baby	
nationality	Citizenship (country) Possible values: dictionary of countries	
relatedInfantWithSeatId	Id of the infant related to the adult with seat	
origin	Departure airport	
destination	Destination airport	
departureDate	Date and time of departure Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T01:00:00	
flightNumber	Flight number.	
classOfService	Information about the selected tariff	
commercialFamily	Fare family type (different meals may be available on the same flight for economy and business).	
fareCode	Commercial family Example: FL	
currency	Search currency	

Response format

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

```

{
  "number": 0,
  "seats": [
    {
      "seat": "string",
      "number": 0,
      "description": "string",
      "characteristics": [
        "string"
      ],
      "occupation": "string",
      "passengerRelatedPrice": [
        {
          "name": "string",
          "type": "string",
          "uniqueCustomerIdentifier": "string",
          "seatPrice": 0,
          "seatUnavailableReason": {
            "isByCharacteristics": boolean,
            "isByCheckin": boolean,
            "isByAgent": boolean,
            "isByRegistrationUnavailability": boolean,
            "isByPreselects": boolean,
            "isByRelatedPassengerUnavailability": boolean
          },
          "isSeatUnavailable": boolean,
          "hasInfant": boolean,
          "autoSeating": boolean
        }
      ],
      "classDesignator": "string"
    }
  ],
  "isEmpty": boolean,
  "passengersTotalData": [
    {
      "passengerReference": "string",
      "passengerPriceLevels": [
        {
          "price": 0,
          "priceLevel": 0
        }
      ]
    }
  ],
  "passengerSeats": [
    {
      "seatCheckinStatus": "ReservationCanceled",
      "flightReference": "string",
      "passengerReference": "string",
      "title": "string",
      "number": 0
    }
  ],
  "preselectedSeats": [

```

```

    {
      "flightReference": "string",
      "passengerReference": "string",
      "title": "string",
      "number": 0
    }
  ]
},
"errors": [
  {
    "ibeCode": "string",
    "description": "string",
    "messages": [
      {
        "code": "string",
        "text": "string",
        "additionalInfo": {
          "ssrType": "string",
          "info": "string"
        }
      }
    ]
  }
]
},
],
"success": boolean
}

```

Response parameters

Parameter		Description
success		Sign of successful request execution If errors array is empty, the value of field will be true
data		Returned data
	isEmpty	Flag identifying whether data is returned from sirena 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 row number
		description	Seat description 9 – central seat A – aisle seat W – window seat
		characteristics	Seat characteristics
		occupation	Sign of seat's occupation
		passengerRelatedPrice	Array of objects with information about seat prices for. Only available for seats with CH characteristics.
		name	Passenger name
		type	Passenger type
		uniqueCustomerIdentifier	Unique passenger ID
		seatPrice	Seat price
		seatUnavailableReason	Array of objects with parameters that are reason of unavailability seats
		isByCharacteristics	Seat is not available for the passenger by characteristics
		isByCheckin	The passenger has been already checked-in
		isByAgent	Another seat was chosen by Agent
		isByRegistrationUnavailability	Check-in is not available
		isByPreselects	Another seat was pre-selected earlier
		isByRelatedPassengerUnavailability	Seat is not available for the passenger because the seat is not suitable for accompanied passengers
		isSeatUnavailable	Seat availability flag
		hasInfant	Sign identifying that passenger travels with infant
		autoSeating	Sign of free seat
		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
	passengerSeats	Array with passenger seats information
	seatCheckinStatus	Seat checkin status
	flightReference	Flight segment reference
	passengerReference	Passenger reference
	title	Seat title
	number	Seat number
	preselectedSeats	Array with information about pre-added seats at the seat selection step in the booking engine
	flightReference	Flight segment reference
	passengerReference	Passenger reference
	title	Seat title
	number	Seat number
	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	Sirena error code
	text	Sirena error text

Valid request example:

```

{
  "referenceNumber": 12,
  "origin": "LED",
  "destination": "DME",
  "departureDate": "2025-10-20T01:00:00",
  "flightNumber": "60",
  "classOfService": "A",
  "fareCode": "FL",
  "commercialFamily": "U6ECONOMY",
  "passengers": [
    {
      "id": 1,
      "firstName": "Artem",
      "surname": "Ivanov",
      "passengerType": 0,
      "birthDate": "1995-07-20",
      "sex": 0,
      "disability": null,
      "hasInfant": false
    },
    {
      "id": 2,
      "firstName": "Ivan",
      "surname": "Ivanov",
      "passengerType": 0,
      "birthDate": "1950-10-20",
      "sex": 0,
      "disability": null,
      "hasInfant": false
    }
  ]
}

```

6_10_api/annul: seat annulling service

API service	annul, annul/partial
Short description	Allows to cancel tickets, cancel tickets for certain passengers in reservation and receive information on canceled bookings
Supported HTTP-(S) methods	GET, POST
Conterminous systems methods used	<p>Sirena:</p> <p>Cancel (part or all) PNR seats (without tickets refund) - release_seats</p> <p>Getting information on booking - order</p>

POST api/annul method

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

Request format: /api/apiVersion/Annul

Request headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Response format:

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

Response parameters:

Parameter	Description
success	Sign of the successful request execution If the errors array is empty, the value of field 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 Sirena answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
code	Error code
text	Error text

GET api/annul method

Description: allows to get information on booking cancellation

Request parameters: X-Session is passed in the header

Parameter	Description
X-Api-Key	API key
pnrNumber	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 successful request execution If the errors array is empty, the value of field 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

Parameter		Description
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from Sirena answer. The array may be empty, it is recommended to analyze the error by the ibeCode field.
	code	Error code
	text	Error text

POST api/annul/partial method

Description: allow to cancel tickets for certain passengers in reservation

Request format: /api/apiVersion/Annul/Partial

Request parameters

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body

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

Response format

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

```

    }],
    "success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of the successful request execution If the errors array is empty, the value of field will be 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 Sirena response. Array can be empty, it is recommended to analyze errors by ibeCode.
code	Error code
text	Error text

6_11_api/reservation/AddNewPassengers, AddAdditionalPassenger, RemoveAdditionalPassenger: service for changing passengers in already issued booking

API service	api/reservation/AddNewPassengers
Short description	Adding an infant without seat (INF) to booking, also making a connection between INF and adult passenger 18+ years old without related infants, adding or remove new passengers in already issued booking
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: modify_pnr

Method POST Reservation/AddNewPassengers

Description: Creating an option to add an infant without seat (INF) to booking, also making a connection between INF and adult passenger 18+ years old without related infants. At the next step making available to issue additional ticket for added passenger via usual issuing process.

A data of added passenger is able to check at a response by GET Reservation request.

Request format: /api/apiVersion/Reservation/AddNewPassengers

Request header

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request parameters

Parameter	Description	Format	Notes
relatedAdultReferenceNumber	ID of an adult passenger at the issued booking that has no accompanied another infants at the booking		
id	Passenger's serial number (numbered from 1)		
firstName	Passenger's name	[A-Z], from two characters	The sum of the full name is no more than 43 characters
middleName	Passenger's patronymic (may be absent)	[A-Z], from one character	The sum of the full name is no more than 43 characters
surname	Passenger's surname	[A-Z], from two characters	The sum of the full name is no more than 43 characters
passengerType	Passenger type		
birthDate	Date of birth		
sex	Passenger's gender Possible values: see dictionary "gender of passengers"		

Parameter	Description	Format	Notes
nationality	Citizenship (country) Possible values: dictionary of countries		
documentNumber	Document number	For different document types there are different format rules: <ul style="list-style-type: none"> • Passport of Russian Federation: 10 chars; • Foreign passport of Russian Federation: 9 chars; • Birth certificate of Russian Federation: $^{[IVX]}+([A-Яa-я]\{2\})[A-Za-z]\{2,4\})\d\{6\}\\$.$ For documents of other countries there is no special format rules.	
documentType	Document type		
documentExpiryDate	Date until which the document is valid Mandatory only for a foreign passport of Russian Federation and national passports of other countries. For other documents, it is automatically set by Sirena.		
loyaltyCard	Loyalty card number "Wings"	10 numbers	

Parameter	Description	Format	Notes
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		In developing
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 assistancetype Possible values: dictionary of assistance types		
wheelchairType	Wheelchair type Possible values: dictionary of wheelchair type		
commentary	Comment (optional)		
socialSecurityNumber	A specific numer of SNILS (optional)	10 digits	
endorsements	Array with information about documents to buy tickets on the subsidized fares or to apply discounts on partner loyalty cards.		

Parameter	Description	Format	Notes
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.		
accompanimentBookingInfo	Information about the reservation containing an accompanying passenger		Currently used when creating reservations for MNN, MNS
number	PNR of a booking containing an accompanying passenger		
surname	Last name of the accompanying passenger		
accompaniedPassengerId	Id number of accompanied passenger (related with subsidized booking by Disability subsidies)		
SubsidyDocumentNumber	Document number for confirming a right to get a subsidy (related for subsidized booking by Disability subsidies or Kaliningrad's subsidy)		

Request body

```
{
  "passengers": [
    {
```

```

    "relatedAdultReferenceNumber": string,
    "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
    },
    "socialSecurityNumber": string,
    "endorsements": [{
        "endorsement": boolean,
        "isDisability": boolean,
        "endorsementDocNumber": string,
        "endorsementDocDate": datetime,
        "loyaltyPartnerCard": string
    }
],
"accompanimentBookingInfo": [{
    "endorsement": boolean,
    "isDisability": boolean
}
],
"accompaniedPassengerId": number,
"SubsidyDocumentNumber": string
}
]
}

```

A format and parameters of a response are the same as for usual response by POST api/Reservation request.

When the passenger will be added to the booking, the next step is issuing the ticket by usual process in depended by a price to pay for INF ticket.

A price for INF at International flights has a cost. Issuing the booking by usual process of paying for booking or paid additional options.

Examples of successful requests and responses

Correct request for adding an infant to reservation:

```
{
  "passengers": [
    {
      "relatedAdultReferenceNumber": "12",
      "id": 14,
      "firstName": "Ss",
      "middleName": null,
      "surname": "Ss",
      "passengerType": 4,
      "birthDate": "2024-01-01",
      "sex": 1,
      "nationality": "RU",
      "documentNumber": "8856561111",
      "documentType": 3,
      "documentExpiryDate": null,
      "loyaltyCard": null,
      "loyaltyCardValidTo": null,
      "phone": null,
      "email": null,
      "disability": {
        "isDeaf": false,
        "isBlind": false,
        "hasGuideDog": false,
        "assistanceType": 0,
        "wheelchairType": null
      },
      "socialSecurityNumber": "31662027345",
      "subsidyDocumentNumber": null,
      "accompaniedPassengerId": null,
      "files": []
    }
  ]
}
```

Method POST Reservation/AddAdditionalPassenger

Description: allows to add an additional passenger with a previously issued ticket to an existing booking within a session

Request format: /api/apiVersion/Reservation/AddAdditionalPassenger

Request header:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body:

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

Request parameters:

Parameter	Description
additionalTicketNumbers	Ticket number of adding passenger

Method POST Reservation/RemoveAdditionalPassenger

Description: allows to remove an additional passenger from an existing booking

Request format: /api/apiVersion/Reservation/RemoveAdditionalPassenger

Request header:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body:

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

}

Request parameters:

Parameter	Description
additionalTicketNumbers	Ticket number of removing passenger

6_12_api/partners: 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.

Request Format /api/apiVersion/partners/types

Request parameters

Parameter	Description
X-Session	Session key
X-API-Key	Unique GUID for systems using IBE API

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 successful request execution If errors array is empty, the value of field will be true
data	Returned data
partnerId	Unique partner's code
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 Sirena response. Array can be empty, it is recommended to analyze errors by ibeCode.
code	Sirena error code
text	Sirena error text

Response example

```

{
  "data": [
    {
      "partnerId": 4,
      "name": "Aeroexpress"
    },
    {
      "partnerId": 1,
      "name": "AlfaInsurance"
    }
  ],
  "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 Alfainsurance services.

Request parameters

Parameter	Description
X-Session	Session key
X-API-Key	Unique GUID for systems using IBE API

Response format

```
{
  "data": [{
    "partnerId": integer,
    "partnerType": integer,
    "bookingStatus": integer,
    "code": string,
    "amount": number,
    "currency": string,
    "preselectedOption": {
      "preselect": boolean,
      "passengers": [{
        "referenceNumber": string,
        "passengerType": string,
        "flightReferences": [
          string
        ]
      }
    ]
  }
},
  "compatibleProductCodes": [
    string
  ],
  "internalError": integer,
  "internalErrorMessage": string,
  "passengerAmounts": [
    {
      "referenceNumber": string,
      "amount": number,
      "currency": 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 successful request execution If errors array is empty, the value of field will be true
data	Returned data
partnerId	Unique partner's code
partnerType	Unique partner's type
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 of partner's pricing
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
	flightReferences	id of segments in the booking with preselected option
	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
	passengerAmounts	An array with information about the amounts for partner services in the context of passengers (returned for Alfainsurance).
	referenceNumber	Passenger reference.
	amount	Amount for the partner product for the passenger.
	currency	Currency.
	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 Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Sirena error code
	text	Sirena 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, 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,
        "flightReferences": [
          "12"
        ]
      }
    ]
  }
], {
  "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 headers

Parameter	Description
X-Session	Session key
X-Api-Key	Unique GUID for systems using IBE API

Request body

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

Request parameters

Parameter	Description
partnerId	Partner's unique code
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)
flightReference	SIgn of service belonging to flight segment id, if not applicable to all segments

Response example

```
{
  "data": {
    "journey": {
      "outboundFlights": [{
        "aircraft": "string",
        "referenceNumber": "integer",
        "origin": "string",
        "destination": "string",
        "departureDate": "datetime",
        "arrivalDate": "datetime",
        "flightNumber": "integer",
        "classOfService": "string",
        "operatingCarrier": "string",
```

```

        "marketingCarrier": "string"
    }
],
"outboundFareFamily": {
    "code": "string"
},
"origin": "string",
"destination": "string",
"departureDate": "datetime"
},
"number": "string",
"passengers": [{
    "id": integer,
    "hasInfant": boolean,
    "referenceNumber": "integer",
    "firstName": "string",
    "middleName": "string",
    "surname": "string",
    "passengerType": integer,
    "birthDate": "datetime0",
    "sex": integer,
    "nationality": "string",
    "documentNumber": "string",
    "documentExpiryDate": "datetime",
    "email": "string",
    "phone": "integer",
    "documentType": integer,
    "isAdult": boolean,
    "isChanged": boolean
}
],
"services": [],
"seats": [],
"partnerProducts": [{
    "partnerId": integer,
    "partnerType": integer,
    "bookingStatus": integer,
    "code": "string",
    "amount": integer,
    "currency": "string",
    "passengerReference": "integer",
    "isInfantProduct": boolean,
    "bookingId": "string",
    "isIssuedByRefundableOffice": boolean
}, {
    "partnerId": integer,
    "partnerType": integer,
    "bookingStatus": integer,
    "code": "string",
    "amount": integer,
    "currency": "string",
    "passengerReference": "integer",
    "isInfantProduct": boolean,
    "bookingId": "string",
    "orderId": string,
    "isIssuedByRefundableOffice": boolean
}
}

```

```

    ],
    "hasSpecialConditions": boolean
  },
  "errors": [],
  "success": boolean
}

```

Examples of invalid requests and responses

Correct request for adding services STANDARTOW (Aeroexpress-standart), ALFADOC (therapist's consultation, DWRP (double luggage packing) and TRAVEL (Alfainsurance-Travel):

```

{
  "products": [
    {
      "partnerId": 4,
      "code": "STANDARTOW"
    },
    {
      "partnerId": 21,
      "code": "ALFADOC",
      "passengerReference": "2"
    },
    {
      "partnerId": 8,
      "code": "DWRP",
      "passengerReference": "2",
      "flightReference": "1"
    },
    {
      "partnerId": 1,
      "code": "TRAVEL"
    }
  ]
}

```

Response:

```

{
  "data": {
    "journey": {
      "outboundFlights": [
        {
          "aircraftIataCode": "320",
          "referenceNumber": "12",
          "origin": "DME",
          "destination": "SVX",
          "departureDate": "2025-09-20T13:35:00",
          "arrivalDateUtc": "2025-09-20T13:05:00Z",
          "departureDateUtc": "2025-09-20T10:35:00Z",
          "arrivalDate": "2025-09-20T18:05:00",
          "flightNumber": "269",
          "classOfService": "W",
          "commercialFamily": "U6ECONOMY",
          "operatingCarrier": "U6",

```

```

        "marketingCarrier": "U6",
        "flightDuration": "2:30",
        "techStops": [],
        "hash": "09/20/2025 13:35:00-269-DME-SVX",
        "statuses": [
            "HK"
        ],
        "isCancelledByAirline": false,
        "isCheckinWithoutPseudo": false
    }
],
"separateFlights": [],
"outboundFareFamily": {
    "code": "FL",
    "category": "ECOPREM",
    "isSubsidized": false,
    "isLowCost": false,
    "isUnaccompaniedMinor": false
},
"separateFareFamilies": [],
"outboundDuration": "2:30",
"separateDuration": "0:00",
"origin": "DME",
"destination": "SVX",
"departureDate": "2025-09-20T13:35:00",
"outboundBaggageAllowances": [
    {
        "quantity": 2,
        "weight": 23,
        "passengerType": 0,
        "fareCode": "FL"
    }
],
"returnBaggageAllowances": [],
"separateBaggageAllowances": [],
"containsInternationalFlights": false
},
"number": "28NVFV",
"passengers": [
    {
        "id": 12,
        "hasInfant": false,
        "referenceNumber": "12",
        "relatedChildrenReferenceNumbers": [],
        "firstName": "PETER",
        "middleName": "",
        "surname": "PARKER",
        "passengerType": 0,
        "birthDate": "2000-01-01T00:00:00",
        "sex": 1,
        "nationality": "RUS",
        "documentNumber": "0111111111",
        "documentExpiryDate": "2049-12-31T00:00:00",
        "loyaltyCard": "0000000002",
        "email": "IURII.TROKHACHEV@ARCADIA.SPB.RU",
        "phone": "79215780959",
        "documentType": 0,
    }
]

```

```

        "isAdult": true,
        "isChanged": false,
        "checkedIn": {},
        "checkInHasStarted": {},
        "checkedInSegment": {},
        "checkInHasStartedSegment": {},
        "hasPassengersDataErrors": false,
        "passengerDataErrors": [],
        "endorsements": []
    }
],
"services": [],
"seats": [],
"partnerProducts": [
    {
        "partnerId": 4,
        "partnerType": 2,
        "bookingStatus": 1,
        "code": "STANDARTOW",
        "amount": 650.00,
        "currency": "RUB",
        "passengerReference": "12",
        "isInfantProduct": false,
        "bookingId": "798efb06-6cef-4dc6-945a-2b03fab8f9da",
        "orderId": "964940",
        "isIssuedByRefundableOffice": false
    }
],
"hasSpecialConditions": false,
"errors": {
    "seats": [],
    "services": [],
    "partnerProducts": [
        {
            "partnerId": 21,
            "partnerType": 3,
            "bookingStatus": 0,
            "code": "ALFADOC",
            "amount": 0.0,
            "currency": "RUB",
            "passengerReference": "2",
            "isInfantProduct": false,
            "isIssuedByRefundableOffice": false,
            "internalError": 5001,
            "internalErrorMessage": "A partner type is not available
for current App Type."
        }
    ]
},
{
    "partnerId": 21,
    "partnerType": 3,
    "bookingStatus": 0,
    "code": "ALFADOC",
    "amount": 0.0,
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "isIssuedByRefundableOffice": false,

```

```

        "internalError": 5000,
        "internalErrorMessage": "Invalid product code"
    },
    {
        "partnerId": 8,
        "partnerType": 4,
        "bookingStatus": 0,
        "code": "DWRP",
        "amount": 0.0,
        "currency": "RUB",
        "passengerReference": "2",
        "flightReference": "1",
        "isInfantProduct": false,
        "isIssuedByRefundableOffice": false,
        "internalError": 5000,
        "internalErrorMessage": "Invalid product code"
    },
    {
        "partnerId": 1,
        "partnerType": 1,
        "bookingStatus": 0,
        "code": "TRAVEL",
        "amount": 0.0,
        "currency": "RUB",
        "isIssuedByRefundableOffice": false,
        "internalError": 5000,
        "internalErrorMessage": "Invalid product code"
    },
    {
        "partnerId": 8,
        "partnerType": 4,
        "bookingStatus": 0,
        "code": "DWRP",
        "amount": 0.0,
        "currency": "RUB",
        "passengerReference": "2",
        "flightReference": "1",
        "isInfantProduct": false,
        "isIssuedByRefundableOffice": false,
        "internalError": 1001,
        "internalErrorMessage": "Invalid passenger or flight
reference."
    },
    {
        "partnerId": 8,
        "partnerType": 4,
        "bookingStatus": 0,
        "code": "DWRP",
        "amount": 0.0,
        "currency": "RUB",
        "passengerReference": "2",
        "flightReference": "1",
        "isInfantProduct": false,
        "isIssuedByRefundableOffice": false,
        "internalError": 1001,
        "internalErrorMessage": "Invalid passenger or flight
reference."
    }

```

```

        },
        {
            "partnerId": 21,
            "partnerType": 3,
            "bookingStatus": 0,
            "code": "ALFADOC",
            "amount": 0.0,
            "currency": "RUB",
            "passengerReference": "2",
            "isInfantProduct": false,
            "isIssuedByRefundableOffice": false,
            "internalError": 5017,
            "internalErrorMessage": "A partner type is not
available."
        }
    ]
},
"errors": [],
"success": true
}

```

Invalid partner's code for TRAVEL service:

```

{
    "products": [
        {
            "partnerId": 4,
            "code": "TRAVEL"
        },
        {
            "partnerId": 4,
            "code": "STANDARTOW"
        }
    ]
}

```

Response:

```

"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"
    }
  ]
}

```

6_13_api/ticket/void: ticket void service

API service	ticket/void
Short description	Allows to void air tickets and additional services
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: void_tickets

POST ticket/void method

Description: void of air tickets and additional services.

Note: if not all tickets from a reservation are entered, the split is performed and all tickets that are entered are moved to a separate reservation, the PNR of which is returned in the POST /ticket/void response.. Tickets that were not voided remain in the original booking. The error is returned when trying to split an infant (or leave it in the original booking) without an adult.

Request format /api/apiVersion/ticket/void

Request header

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

Request body

```

{
  "TicketNumbers": [string],
  "NeedReturnSeats": boolean
}

```

Request parameters

Parameter	Description
TicketNumbers	Array of air tickets and additional services to void Example: 2622400006710
NeedReturnSeats	Seat void flag

Response body

```
{
  "data": {
    "voidedTicketNumbers": [string],
    "notVoidedTicketNumbers": [string],
    "notVoidReason": string,
    "separatedPnr": string
  },
  "errors": [{
    "ibeCode": string,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }
],
  "success": boolean
}
```

Response parameters

Parameter	Description
data	Returned data
voidedTicketNumbers	Array of voided tickets
notVoidedTicketNumbers	Array of tickets that failed to be voided
notVoidReason	Reason why void failed
separatedPnr	Pnr of splitted reservation
errors	Information about errors appeared during query execution.
ibeCode	IBE error code, see attachment: IBE error codes

	description	IBE error description, see attachment: IBE error codes
	messages	Array of objects, containing conterminous system's errors Array might be empty. It is recommended to analyze ibeCode field.
	code	Sirena error code
	text	Sirena error text
	success	Sign of successful request execution If errors array is empty, the value of field will be true

6_14_api/queue: services for working with queues (for agents)

API service	queue/placePNR, queue/list, queue/counttotal, queue/removeItem
Short description	The service allows placing PNR in queue, retrieve information from queue, removes reservations from queue
Supported HTTP-(S) methods	POST
Used methods of related systems	Sirena: queue_pnr_place, queue_show, queue_count, queue_pnr_remove

POST queue/placePNR method

Description: placing the PNR in queues

Request format /api/apiVersion/queue/placePNR

Request header

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body:

```

{
  "queueNumber": integer,
  "pnr": "string",
  "queue_PlacePNR": "string"
}

```

Request parameters

Parameter	Description
queueNumber	Queue number in which need to place the booking
pnr	Booking number which need to be queued
queue_PlacePNR	(optional) parameter in developing

Response format

```

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

```

Response parameters

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be true
data	Empty array of objects
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	Sirena error code
	text	Sirena error text
	additionalInfo	Additional information about error
	ssrType	Ssr type
	info	Information

Example of successful request and response:

Request:

Request

```
{
  "queueNumber": 1,
  "pnr": "1Z9RR1"
}
```

Response:

Response

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

Examples of failed requests and responses:

An example of an attempt to add the booking to the queue, which is not available for the key of the office under which the addition is being made:

Response

```
{
  "errors": [{
    "ibeCode": 36001,
    "description": "Pnr access error",
    "messages": []
  }],
  "success": false
}
```

An example of an attempt to add the booking to inaccessible queue for the office key under which the addition is being made:

Response

```
{
  "errors": [{
    "ibeCode": 36002,
    "description": "Queue access error",
    "messages": []
  }
],
  "success": false
}
```

POST queue/list method

Description retrieving the list of bookings from the queue

Request format /api/apiVersion/queue/list

Request header

Parameter	Description
X-API-Key	API key
X-Session	Session ID

Request body

```
{
  "queueNumber": integer,
  "queue_List": "string"
}
```

Request parameters

Parameter	Description
queueNumber	Number of the queue for which the list of bookings is requested to be returned
queue_List	(optional) list's name

Response format

```
{
  "data": [{
    "queuedPnrs": [{
      "number": string,
      "passengers": [{
        "name": string,
        "surname": 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 successful request execution If the errors array is empty, the value of field will be true
data	Array with information about booking lists
queuedPnrs	Array of objects with basic information about queues
number	Queue number
passengers	Array with information about passengers in the booking
name	Name and patronymic (if exists) of the passenger
surname	Last name of the passenger
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	Sirena error code
text	Sirena error text

		additionalInfo	Additional information about error
		ssrType	Ssr type
		info	Information

Example of successful request and response:

Request:

Request

```
{
  "queueNumber": 1
}
```

Response:

Response

```
{
  "data": {
    "queuedPnrs": [{
      "number": "1Z29X1",
      "passengers": [{
        "name": "IVAN IVANOVICH",
        "surname": "IVANOV"
      }]
    }]
  },
  "errors": [],
  "success": true
}
```

Examples of failed requests and responses

Request for a non-existent queue

Response

```
{
  "errors": [{
    "ibeCode": 36000,
    "messages": [{
      "code": "33065",
      "text": "No such queue"
    }]
  }],
  "success": false
}
```

POST queue/counttotal method

Description retrieving the count of records

Request format /api/apiVersion/queue/counttotal

Request header

Parameter	Description
X-API-Key	API key
X-Session	Session ID

Request parameters are empty

Response format

```
{
  "data": [{
    "queuesInfo": [{
      "queueNumber": integer,
      "queueName": string,
      "count": integer
    }
  ]
},
  "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 successful request execution If the errors array is empty, the value of field will be true

data	An array with information about the number of bookings in queues	
	queuesInfo	Array of objects with basic information about the number of PNRs in the queues
	queueNumber	Queue number
	queueName	Queue title
	count	Number of bookings in the queue
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	Sirena error code
	text	Sirena error text
	additionalInfo	Additional information about error
	ssrType	Ssr type
	info	Information

Exempl of successful response:

Response

```
{
  "data": {
    "queuesInfo": [{
      "queueNumber": 1,
      "queueName": "CONFOS",
      "count": 7
    }, {
      "queueNumber": 2,
      "queueName": "NOT SOLD PTA",
      "count": 4
    }, {
      "queueNumber": 7,
      "queueName": "SCH CHD",
      "count": 36
    }
  ]
},
"errors": [],
"success": true
}
```

POST queue/removeItem method

Description removing the PNR from the queues (actual for ADP only)

Request format /api/apiVersion/queue/removeItem

Request header

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request parameters

Parameter	Description
queueNumber	Number of the queue from which the reservation is requested to be deleted
pnr	Booking number to be removed from the queue

Response format

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

Response parameters

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

success	Sign of successful request execution If errors array is empty, the value of field will be true
data	Empty array of objects
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	Sirena error code
text	Sirena error text
additionalInfo	Additional information about error
ssrType	Ssr type
info	Information

Example of successful request and response:

Request:

Request

```
{
  "queueNumber": 111,
  "pnr": "1Z9S18"
}
```

Response:

Response

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

Examples of failed requests and responses are the same as examples of failed requests and responses for POST queue/placePNR method

6_16: api/exchange: service of ticket exchanging

API service	api/exchange, api/exchange/search, api/exchange/confirmFree, api/exchange/cancel
Short description	Service of exchanging tickets by API IBE
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: exchange_pricing, divide_order, exchange_confirm, exchange_cancel

POST Exchange/Search method

Description: gets information about flights available for exchange with regards to specified search parameters

Request format: api/apiVersion/Exchange/Search

Request headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body:

```
{
  "pnrNumber": string,
  "lastName": string,
  "originDestinations": [
    {
      "origin": string,
      "destination": string,
      "flightNumber": integer,
      "classOfService": string,
      "departureDate": datetime,
      "segmentReference": string,
      "segmentReferenceLast": string
    }
  ],
  "passengerReferences": [
    string
  ],
  "searchPeriod": string,
  "searchIndirectFlights": boolean
}
```

Request parameters

Parameter	Description
pnrNumber	Number of reservation (or it exists in session context)
lastName	Passenger's surname
originDestination	An array of data about the part of the flight to be exchanged
origin	IATA-code of origin airport
destination	IATA-code of destination airport
flightNumber	Flight number
classOfService	Class of service
departureDate	Departure date
segmentReference	Reference to the segment to be exchanged
segmentReferenceLast	Last segment reference for exchange. Should be provided if require to change segment range
passengerReference	Array of references to the passengers whose tickets to be exchanged
searchPeriod	A period of days for getting recommendations to exchange (available values: 1, 3, 5; 'null' - accepting as '1' value)
searchIndirectFlights	An indication of the search for transfer flights in Sirena. If the value is true, Sirena searches for flights with transfers. By default, the value is false

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,
                "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
            }
        ]
    }
],
"currency": string,

```

```

    "foodOnBoards": [{
      "departureAirport": string,
      "arrivalAirport": string,
      "flightNumber": string,
      "commercialFamily": string,
      "translation": {
        "ru": string,
        "en": string,
        "zh": string
      }
    }
  ],
  "airShopSessionId": string,
  "moneyWallCoefficient": float,
  "containsInternationalFlights": boolean
},
"errors": [{
  "ibeCode": integer,
  "description": string,
  "messages": [{
    "code": string,
    "text": string,
    "additionalInfo": {
      "ssrType": string,
      "info": string
    }
  }
]
},
],
"success": true
}

```

Response parameters

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be true
data	Returned data
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

Parameter		Description
	code	Fare family code, for example: ECOLIGHTRU/ECOFLEXRU/ECOPREMURU etc.
	category	Fare family category, for example: ECOLIGHT/ECOFLEX/ECOPREM Possible values: see Attachment 2, Main fare family categories
	order	Fare families display order
	commercialFamily	Fare family type for example: U6ECONOMY or U6BUSINESS
	isSubsidized	Sign of subsidized fare
	canRefund	Sign of refundable fare
	hasUpsale	Indication that present fare can be upgraded to next one in price category. Used for management of Upsale block display at frontend.
	options	Array of fareFamiliesOptions objects with information about fare 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
	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	Option's translation to russian
	en	Option's translation to english
	zh	Option's translation to chinese
recommendations		Array of recommendations for flights

Parameter		Description
	totalPrice	Total ticket price including all applied discounts
	totalAddCollectPrice	Price to be paid in case of exchange
	breakdowns	Array of tickets that are available for exchange
	ticketNumber	Number of issued ticket
	passengerReference	Reference to a passenger
	paxType	Type of a passenger
	baseFareAmount	Base price
	taxAmount	Amount of fees
	totalAmount	Full ticket price including fees
	AddCollectAmount	Amount to be paid in case of exchange
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 Sirena response. Array can be empty, it is recommended to analyze errors by ibeCode.
	code	Sirena error code
	text	Sirena error text

FlightGroup object

Contains information about flights

```
{
  "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,
    "originalFlightReferenceNumber": string
}
],
"flightGroupDuration": string,
"hash": string
}

```

Parameters of flightGroup object

data.flights	
flightGroupId	Index number used in recommendations
flights	Array of objects with information about flight segments
flightId	Segment number
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 Example: Airbus A320

data.flights	
aircraftIataCode	IATA aircraft code
flightDuration	Duration of flight for current segment
flightNumber	Flight number
marketingCarrier	Marketing carrier Example: U6, S7
operatingCarrier	Operating carrier (one actually performed flight) Example: U6, OK
originalFlightReferenceNumber	A number of flight segment for exchanging
flightGroupDuration	Total flight time: summed up of duration values of each transfer and technical stops
hash	Service field

Example of valid request:

Search for recommendations for ticket exchange for SVX-DME flight in 3 days range:

```
{
  "pnrNumber": "28NVFV",
  "lastName": "Ivanov",
  "originDestinations": [
    {
      "origin": "SVX",
      "destination": "DME",
      "flightNumber": "273",
      "classOfService": "W",
      "departureDate": "2025-09-20T13:35:00",
      "segmentReference": "12"
    }
  ],
  "passengerReferences": [
    "12"
  ],
  "searchPeriod": "3",
  "searchIndirectFlights": true
}
```

POST Exchange method

Note: method is also available on ADP

Description: allows to make a rebook of ticket

Request format: api/apiVersion/Exchange

Request headers

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body

```
{
  "pnrNumber": string,
  "lastName": string,
  "travelParts": [
    {
      "segmentReference": string,
      "segmentReferenceLast": string,
      "fareFamily": {
        "code": string,
        "category": string,
        "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 lastname
travelParts	Array with information about the flight segment
segmentReference	Reference to the flight segment
segmentReferenceLast	Reference to the last flight segment
fareFamily	Object with information about the fare family
code	Tariff code
category	Tariff category
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	Flight number
classOfService	Information about selected fare
fares	Fare's information object
fareBasis	Base fare name
isBreakpoint	Flag identifying whether fare ends on present segment
paxType	Passenger type
passengerReferences	An array of references to passengers for whom the exchange is being made

Response format

```
{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
```

```

"returnFlights": [fullReservationFlight],
"outboundFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": "boolean",
  "isLowCost": "boolean",
  "isUnaccompaniedMinor": "boolean"
},
"returnFareFamily": {
  "code": "string",
  "category": "string",
  "isSubsidized": "boolean",
  "isLowCost": "boolean",
  "isUnaccompaniedMinor": "boolean"
},
"outboundDuration": "string",
"returnDuration": "string",
"origin": "string",
"destination": "string",
"departureDate": "datetime",
"returnDate": "datetime",
"outboundBaggageAllowances": [
  {
    "quantity": "integer",
    "weight": "integer",
    "passengerType": "string"
  }
],
"returnBaggageAllowances": [
  {
    "quantity": "integer",
    "weight": "integer",
    "passengerType": "string"
  }
],
"containsInternationalFlights": "boolean"
},
"number": "string",
"reservationDate": "datetime",
"bookingTimeLimit": "datetime",
"passengers": [passengerData],
"services": [
  {
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": "integer",
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": "integer",
    "price": "integer",

```

```

        "discountAmount": "integer",
        "amadeusErrorCode": "string",
        "amadeusErrorMessage": "string",
        "internalError": "string",
        "internalErrorMessage": "string",
        "issueDate": "datetime",
        "issuedByThirdPartyOffice": true
    },
    ],
    "seats": [
        {
            "number": "integer",
            "title": "string",
            "code": "string",
            "status": "string",
            "text": "string",
            "petType": "string",
            "category": "string",
            "fareAmount": "integer",
            "passengerQualifier": "string",
            "passengerReference": "string",
            "flightReference": "string",
            "referenceNumber": "string",
            "packReferenceNumber": "string",
            "ticketNumber": "string",
            "finalPrice": "integer",
            "price": "integer",
            "discountAmount": "integer",
            "amadeusErrorCode": "string",
            "amadeusErrorMessage": "string",
            "internalError": "string",
            "internalErrorMessage": "string",
            "issueDate": "datetime",
            "issuedByThirdPartyOffice": "boolean"
        }
    ],
    "luggageSeats": [
        {
            "referenceNumber": "string",
            "passengerReference": "string"
        }
    ],
    "insurances": [
        {
            "number": "string",
            "passengerType": "string",
            "code": "string",
            "status": "string",
            "text": "string",
            "petType": "string",
            "category": "string",
            "fareAmount": "integer",
            "passengerQualifier": "string",
            "passengerReference": "string",
            "flightReference": "string",
            "referenceNumber": "string",
            "packReferenceNumber": "string",

```

```

        "ticketNumber": "string",
        "finalPrice": "integer",
        "price": "integer",
        "discountAmount": "integer",
        "amadeusErrorCode": "string",
        "amadeusErrorMessage": "string",
        "internalError": "string",
        "internalErrorMessage": "string",
        "issueDate": "datetime",
        "issuedByThirdPartyOffice": "boolean"
    }
],
"partnerProducts": [
    {
        "partnerId": "string",
        "partnerType": "string",
        "bookingStatus": "string",
        "code": "string",
        "amount": "integer",
        "currency": "string",
        "passengerReference": "string",
        "flightReference": "string",
        "isInfantProduct": "boolean",
        "bookingId": "string",
        "orderId": "string",
        "preselectedOption": {
            "preselect": "boolean",
            "passengers": [
                {
                    "referenceNumber": "string",
                    "passengerType": "string"
                }
            ]
        },
        "compatibleProductCodes": [
            "string"
        ],
        "internalError": "string",
        "internalErrorMessage": "string"
    }
],
"ticketFares": [
    {
        "referenceNumber": "string",
        "referenceQualifier": "string",
        "baseAmount": "integer",
        "totalAmount": "integer",
        "currency": "string",
        "passengerReferences": [
            {
                "referenceNumber": "string",
                "referenceQualifier": "string"
            }
        ],
        "passengerType": "string"
    }
],

```

```

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

```

Response parameters

Parameter	Description
success	Sign of successful request execution. If the errors array is empty, the value of the field will be true.
data	The data returned by the request.
number	New booking number. It can be either equal to the previous one, or differ in the case of a partial exchange.
reservationDate	Date and time of booking creation Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T15:44:00
bookingTimeLimit	Date and time until which the unpaid reservation will exist (rules are set by settings in IBE)
passengers	An array of passengerData objects with information about the passengers that were requested to be exchanged
services	Array of objects with information about additional 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's services.
journey	Information about the new segments for which the exchange is made
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
returnFareFlights	Service field, contains fareFamily code
outboundDuration	The total duration of the flight "there"

Parameter		Description
	returnDuration	The total duration of the return flight
	origin	Initial airport of departure
	destination	Initial airport of destination
	departureDate	Departure date
	returnDate	Return date
	outboundBaggageAllowances	An array of objects containing detailed information about the baggage allowance for the outward direction
	returnBaggageAllowances	An array of objects containing detailed information about the baggage allowance for the "back" direction
	containsInternationalFlights	A sign that one of the airports is located outside the territory of the Russian Federation
ticketFares		An array of objects with information about fares 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. Passenger types .
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "TST"
	totalAmount	The total fare's amount
	currency	Reservation currency
	fareBasisCodesFlightRefs	Array of objects with information about base fares
	ticketFaresWithoutZapOff	An array of objects with information about fares without applying a discount
	baseAmount	Base fare amount
	passengerReferences	An array of objects with links to passengers
	referenceNumber	Reference number

Parameter		Description
	referenceQualifier	Link type, for example "PA"
	passengerType	Passenger type Possible values: see Attachment. Passenger types
	referenceNumber	Reference number
	referenceQualifier	Link type, for example "TST"
	totalAmount	The total fare's amount
	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	Sign of availability of booking cancellation
	accountCode	Account code
	tourCode	Tour code
errors		Information about errors that occurred during the operation of the service
	ibeCode	Error code, see Attachment. IBE error codes
	description	Error description, see Attachment. IBE error codes
	messages	An array of objects containing errors from Sirena's answer. The array may be empty, it is recommended to analyze the error by the ibeCode field
	code	Sirena error code
	text	Sirena error text

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
}

```

Parameters of 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 fare
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

Parameter	Description
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

```
{
  "id": "integer",
  "hasInfant": "boolean",
  "referenceNumber": "string",
  "relatedAdultReferenceNumber": "string",
  "relatedInfantReferenceNumber": "string",
  "disability": {
    "isDeaf": "boolean",
    "isBlind": "boolean",
    "hasGuideDog": "boolean",
    "assistanceType": "string",
    "wheelchairType": "string",
    "commentary": "string"
  },
  "firstName": "string",
  "middleName": "string",
  "surname": "string",
  "passengerType": "string",
  "birthDate": "datetime",
  "sex": "string",
  "nationality": "string",
  "documentNumber": "string",
  "documentExpiryDate": "datetime",
  "loyaltyCard": "string",
  "email": "string",
  "phone": "string",
  "documentType": "string",
  "isAdult": "boolean",
  "isChanged": "boolean",
  "checkedIn": {
    "additionalProp1": "boolean",
    "additionalProp2": "boolean",

```

```

        "additionalProp3": "boolean"
    },
    "checkInHasStarted": {
        "additionalProp1": "boolean",
        "additionalProp2": "boolean",
        "additionalProp3": "boolean"
    },
    "checkedInSegment": {
        "additionalProp1": "boolean",
        "additionalProp2": "boolean",
        "additionalProp3": "boolean"
    },
    "hasPassengersDataErrors": "boolean",
    "passengerDataErrors": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "endorsements": [
        {
            "endorsement": "boolean",
            "isDisability": "boolean",
            "endorsementDocNumber": "string",
            "endorsementDocDate": "datetime",
            "loyaltyPartnerCard": "string"
        }
    ]
}

```

Parameters of passengerData object

Parameter	Description
id	Passenger's serial number (numbered from 1)
hasInfant	A sign that this passenger has a baby
referenceNumber	Reference to this passenger in the booking
relatedAdultReferenceNumber	Reference to the adult passenger in the booking with whom the child or infant flies. Returns only for children and infants
relatedInfantReferenceNumber	Reference to the infant passenger in the booking with whom the adult flies. Returns only for adult
disability	Array with information about persons with disabilities
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
birthDate	Date of birth
sex	Passenger's gender Possible values: see dictionary "gender of passengers"
nationality	Nationality (country) Possible values: dictionary of countries
documentNumber	Document number
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	E-mail
phone	Phone number
documentType	Document type Possible values: dictionary of document types
isAdult	A sign that the passenger is an adult true for ADT/SRC/YTH type passengers false for CHD/INF type passengers
isChanged	Indication of passenger's data changing
checkedIn	Array with information about passenger's check-in on flights
checkInHasStarted	Array with information about starting check-in on flights
checkedInSegment	Array with information about passenger's check-in on the flight segment
hasPassengersDataErrors	The sign of errors in passengers data.

Parameter	Description
passengerDataErrors	Array with detailed information of errors in passengers data.
key	The key.
value	The value.
endorsements	Array with information about documents to buy tickets on the subsidized fares or to apply discounts on partner loyalty cards

Examples of valid and invalid requests and responses:

Valid request for ticket exchange for one adult passenger for AER-DME flight:

```
{
  "pnrNumber": "28NVFV",
  "lastName": "Ivanov",
  "travelParts": [
    {
      "segmentReference": "1",
      "segmentReferenceLast": "0",
      "fareFamily": {
        "code": "BS",
        "isSubsidized": false,
        "category": "BUSFLEX",
        "isLowCost": true,
        "isUnaccompaniedMinor": false
      },
      "addFlights": [
        {
          "origin": "AER",
          "destination": "DME",
          "departureDate": "2025-10-12T11:15:00",
          "arrivalDate": "2025-10-19T13:40:00",
          "flightNumber": "280",
          "classOfService": "Z",
          "fares": [
            {
              "fareBasis": "ZBSOW",
              "isBreakpoint": true,
              "paxType": "Adult"
            }
          ]
        }
      ]
    }
  ],
  "passengerReferences": [
    "2"
  ]
}
```

Response for the invalid request (incorrect segmentReference parameter has been passed):

```
{
  "errors": [
    {
      "ibeCode": 11004,
      "description": "Unknown segment reference.",
      "messages": []
    }
  ],
  "success": false
}
```

POST Exchange/ConfirmFree method

Description: confirms exchange if no additional payment required

Request format: api/apiVersion/Exchange/ConfirmFree

Request headers:

Parameter	Description
X-API-Key	API key
X-Session	Session ID

Request body:

```
{  
  "pnrNumber": string,  
  "lastName": string  
}
```

Response format is the same as POST api/Exchange response

POST Exchange/Cancel method

Description: allows to cancel the running ticket exchange process

Request format: api/apiVersion/Exchange/Cancel

Request headers:

Parameter	Description	Mandatory
X-API-Key	API key	Yes
X-Session	Session ID	Yes

Request body:

```
{  
  "pnrNumber": string,  
  "lastName": string  
}
```

Request parameters:

Parameter	Description
pnrNumber	Reservation (PNR) number
lastName	Passenger's surname

Response format:

```
{
  "data": {
    "tickets": [
      {
        "number": "string",
        "passengerReference": "string",
        "flightReferences": [
          "16"
        ],
        "isInfantTicket": boolean,
        "isIssuedByRefundableOffice": boolean,
        "issueDate": datetime
      }
    ],
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "returnFlights": [fullReservationFlight],
      "separateFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": "string",
        "category": "string",
        "isSubsidized": true,
        "isLowCost": true,
        "isUnaccompaniedMinor": true
      },
      "returnFareFamily": {
        "code": "string",
        "category": "string",
        "isSubsidized": true,
        "isLowCost": true,
        "isUnaccompaniedMinor": true
      },
      "separateFareFamilies": [
        {
          "flightReference": "string",
          "code": "string",
          "category": "string",
          "isSubsidized": true,
          "isLowCost": true,
          "isUnaccompaniedMinor": true
        }
      ],
      "outboundDuration": "string",
      "returnDuration": "string",
      "separateDuration": "string",
    }
  }
}
```

```

"origin": "string",
"destination": "string",
"departureDate": "2025-05-07T16:32:04.939Z",
"returnDate": "2025-05-07T16:32:04.939Z",
"outboundBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult",
    "fareCode": "string"
  }
],
"returnBaggageAllowances": [
  {
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult",
    "fareCode": "string"
  }
],
"separateBaggageAllowances": [
  {
    "flightReference": "string",
    "quantity": 0,
    "weight": 0,
    "passengerType": "Adult",
    "fareCode": "string"
  }
],
"containsInternationalFlights": true
},
"number": "string",
"reservationDate": "2025-05-07T16:32:04.939Z",
"bookingTimeLimit": "2025-05-07T16:32:04.939Z",
"passengers": [passengerData],
"services": [
  {
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "subType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "currency": "string",
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",

```

```

    "internalErrorMessage": "string",
    "issueDate": "2025-05-07T16:32:04.939Z",
    "issuedByThirdPartyOffice": true,
    "isIssuedByRefundableOffice": true,
    "serviceDuration": 0,
    "expireDateUtc": "2025-05-07T16:32:04.939Z"
  }
],
"seats": [
  {
    "number": 0,
    "title": "string",
    "isCheckinSeat": true,
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "subType": "string",
    "category": "string",
    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "currency": "string",
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2025-05-07T16:32:04.939Z",
    "issuedByThirdPartyOffice": true,
    "isIssuedByRefundableOffice": true,
    "serviceDuration": 0,
    "expireDateUtc": "2025-05-07T16:32:04.939Z"
  }
],
"luggageSeats": [
  {
    "referenceNumber": "string",
    "passengerReference": "string"
  }
],
"insurances": [
  {
    "number": "string",
    "passengerType": "Adult",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "subType": "string",
    "category": "string",

```

```

    "fareAmount": 0,
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": 0,
    "price": 0,
    "currency": "string",
    "discountAmount": 0,
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "UnknownError",
    "internalErrorMessage": "string",
    "issueDate": "2025-05-07T16:32:04.939Z",
    "issuedByThirdPartyOffice": true,
    "isIssuedByRefundableOffice": true,
    "serviceDuration": 0,
    "expireDateUtc": "2025-05-07T16:32:04.939Z"
  }
],
"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",
    "isIssuedByRefundableOffice": true,
    "preselectedOption": {
      "preselect": true,
      "passengers": [
        {
          "referenceNumber": "string",
          "passengerType": "Adult",
          "flightReferences": [
            "string"
          ]
        }
      ]
    }
  ]
},
"compatibleProductCodes": [
  "string"
],
"internalError": "UnknownError",
"internalErrorMessage": "string",
"passengerAmounts": [
  {
    "referenceNumber": "string",
    "amount": 0,

```

```

        "currency": "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",
    "expirationUtc": "2025-05-07T16:32:04.939Z"
  }
],
"hasSpecialConditions": true,
"isRefundAvailable": true,
"annulUnavailabilityReason": "FromKyrgyzstan",
"isAnnulAvailable": true,
"isAgentReservation": true,
"accountCode": "string",
"tourCode": "string",
"flightsWithOpenDateCount": 0,
"relatedPnrNumbers": [
  "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 successful request execution If errors array is empty, the value of field will be true
data	Returned data
tickets	Information about issued tickets
flightReferences	Flight segments
number	Ticket number. Example: 262 2439494143
passengerReference	A reference to the passenger this ticket belongs to
isInfantTicket	A sign of ticket for baby
issueDate	Date and time of issuing the tickets
journey	Information about flights in current reservation
outboundFlights	An array of fullReservationFlight objects with information about flights for "there" direction
returnFlights	An array of fullReservationFlight objects with information about flights for "return" direction
separateFlights	An array of fullReservationFlight objects with information about non-directional flight segments

Parameter		Description
	outboundFareFamily	Object with information about tariffs for "there" direction
	returnFareFamily	Object with information about tariffs for return direction
	separateFareFamilies	Object with information about tariffs for non-directional flight segments
	outboundDuration	Outbound flight's duration
	returnDuration	Return flight's duration
	separateDuration	Overall duration for non-directional flight segments
	origin	Origin airport
	destination	Destination airport
	departureDate	Outbound flight's date
	returnDate	Return flight's date
	outboundBaggageAllowances	An array of objects containing information about the baggage allowances for the outbound flight
	returnBaggageAllowances	An array of objects containing information about the baggage allowances for the return flight
	separateBaggageAllowances	An array of objects containing detailed information about the baggage allowances for non-directional flight segments
	containsInternationalFlights	A sign that one of the airports is located outside the territory of the Russian Federation
	number	Number of PNR
	reservationDate	Date and time of booking creation Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T15:44:00
	bookingTimeLimit	Date and time until the unpaid booking will exist (rules are set in the settings in IBE)
	passengers	Array of passengerData object with information about passengers
	services	Array of objects with information about auxiliary services added to the reservation
	seats	Array of objects with information about seats added to the reservation

Parameter	Description
luggageSeats	An array of objects with information about luggage with separate air ticket and ssr (CBBG)
insurances	Object with information about the selected insurance
partnerProducts	An array of objects with information about partner services
ticketFares	An array of objects with information about fares after applying the discount
ticketFaresWithoutZapOff	An array of objects with information about fares without applying a discount
fareBasisCodesFlightRefs	Array of objects with basis fare information
hasSpecialConditions	Indication of the additional conditions
isRefundAvailable	Indication of the availability of refund realization
annulAnavailabilityReason	Reason for unavailability of booking cancellation
isAnnulAvailable	Indication of availability of booking cancellation
isAgentReservation	Indication of agent's booking
accountCode	Account code
tourCode	Tour code
flightsWithOpenDateCount	Number of flights with open date
relatedPnrNumbers	An array of related PNRs obtained as a result of splitting a reservation as part of a partial exchange
errors	Information about errors appeared during query execution
ibeCode	Error code, see attachment: IBE error codes
description	Error's description, see attachment: 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's code
text	Error's text
additionalInfo	Additional info

6_18_api/InvoluntaryRefund/Moneyless: involuntary ticket refund service (for agents)

API service	InvoluntaryRefund/moneyless, InvoluntaryRefund/Calculate
Short description	Let to process involuntary refund of tickets and services for agents, calculates sum for refund
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: Receiving booking information: order, pnr_status Refund initiation: svc_emd_refund_query, svc_emd_refund_confirm, payment-ext-auth Making changes to a booking: add_ssr, modify_pnr, divide_order

POST InvoluntaryRefund/Calculate method

Description: calculates a sum for involuntary refund

Request format: /api/apiVersion/involuntaryRefund/Calculate

Request headers:

Parameter	Description
X-API-Key	API key
X-Session	Session ID

Request parameters: are absent

Request body:

```
{
  "passengers": [
    {
      "passengerReferenceNumber": "string",
      "relatedAdultReferenceNumber": "string",
      "passengerType": "Adult"
    }
  ]
}
```

Request body parameters:

Parameter	Description
passengers	Array of passenger's info
passengerReferenceNumber	Reference to passenger for whom refund is performing
relatedAdultReferenceNumber	Reference to related adult passenger (for child)
passengerType	Passenger type Example: Adult

POST involuntaryRefund/moneyless method

Description: method for agents that let to process involuntary refund of tickets and services in PSS system without interaction with airline's payment system.

Request format: /api/apiVersion/involuntaryRefund/Moneyless

Request headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request parameters:

Parameter	Description
pnrNumber	Unique booking number (PNR).
passengers	Array with information about passengers whom needs to process the involuntary moneyless refund
isTicketRefund	The sign that a ticket with services or only services is being refunded.
partnerProducts	An array of partner services to be returned (the array parameters are similar to the apiVersion/partners/id response parameters)
lastName	Passenger's last name

passengerReferenceNumber	<p>Passenger reference.</p> <p>Associated with field data -> passengers -> referenceNumber</p>
relatedAdultReferenceNumber	<p>Reference to the adult passenger with whom the child or infant is flying.</p> <p>The parameter is filled for the child and infant</p>
relatedInfantReferenceNumber	<p>Reference to the infant with whom the adult is flying.</p> <p>The parameter is filled for the adult who flies with infant.</p>
passengerType	<p>Passenger type.</p> <p>Possible types of passengers are presented in EN 8 1 Attachment: Dictionaries of passenger types.</p>
services	An array with information about purchased and refundable additional services of the airline.
code	Returned additional service code.
ticketNumber	Ticket number of the returned additional service.

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": [],
```

```

    "serviceRefundDetails": [{
      "code": string,
      "text": string,
      "commissionAmount": integer,
      "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,
  "flightReference": string,
  "isInfantProduct": boolean,
  "bookingId": string,
  "orderId": string,
  "compatibleProductCodes": [string]
}
],
  "crmRefundRequestId": [string]
  "refundWithTicketOnlyServices": [string]
},
"errors": [{
  "ibeCode": number,
  "description": string,
  "messages": [{
    "code": string,
    "text": string
  }
]
},
"success": boolean
}

```

Response parameters

Parameter	Description	Comments
success	Sign of successful request execution If errors array is empty, the value of field will be 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: see EN 8.1 Attachment: Dictionaries of 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 for refund	
refundBonuses	Number of refunded "Wings" bonuses	
refundCurrency	Amount of roubles for refund	
refundTotal	Total refund amount, sum of refundBonuses and refundCurrency	
ticketNumber	ATC number Ticket number should contain 13 digits	
taxDetailsInformation	Services information about tariffs and fees. Important: Information from present segment can be ignored. RefundBonuses and refundCurrency values should be displayed to user.	

				taxCategory	Tax category For refundable taxes "R" is returned, for non-refundable taxes - "NRT".	
				taxDetails	Array of objects with full information about fares and fees details.	
				rate	Fare/fee sum	
				currencyCode	Currency code	
				type	Fare/fee type	
				luggageSeatsRefundDetails	Array with information about CBBG luggage with separate air ticket to refund. All array elements are the same with passengerRefundDetails array elements, the field "surname" has value "CBBG".	In developing
				serviceRefundDetails	Array of objects with service's refund information.	
				code	Service code Possible values: see EN 8 1 Attachment: Dictionaries of services types	
				text	Type of refundable pet when code="PETC".	
				commissionAmount	Commission amount for refund of service.	
				refundBonuses	Number of refunded "Wings" bonuses	
				refundCurrency	Number of refunded roubles	
				refundTotal	Total refund amount, sum of refundBonuses and refundCurrency	
				ticketNumber	EMD number Ticket number should contain 13 digits	
				taxDetailsInformation	Services information about tariffs and fees.	
				taxCategory	Tax category. For refundable taxes "R" is returned, for non-refundable taxes - "NRT".	
				taxDetails	Array of objects with information about tariffs and fees.	

				rate	Fare/fee sum	
				currencyCode	Currency code	
				type	Fare/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 EN 8 3 Attachment: Partner's services.	
				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 the amount for refund of partner product is unknown.	
				partnerId	Partner's id.	
				partnerType	Partner's type.	
				passengerReference	Reference to passenger for whom partner's service is being refunded	
				refundAmount	Refundable sum.	
				flightReference	Reference to the flight segment, if the partner product wasn't added for the entire flight.	
				orderId	Partner order id	
				compatibleProductCodes	Codes of services of the current partner, which are compatible with this service when purchased it	
				crmRefundRequestId	Refund request number in CRM. Example: 1-46MJI9D.	
				refundWithTicketOnlyServices	Array with service codes that can't be refunded without refunding the ticket.	In developing

errors	Information about errors appeared during query execution.	
ibeCode	Error code, see EN 8 2 Attachment: IBE error codes	
description	Code description, see. EN 8 2 Attachment: IBE error codes	
messages	Array of objects, containing errors from Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode.	
code	Sirena error code	
text	Sirena error text	

6_19_api/refund: voluntary tickets and services refund service

API Service	refund, refund/calculate, refund/partial
Short description	Allows to calculate refund sum and perform tickets and services refund and also get information about already performed refund.
Supported HTTP(S) methods	POST, GET

Conterminous systems methods used	<p>Sirena:</p> <p>Receive information about booking: order, pnr_status</p> <p>Initiate refund: svc_emd_refund_query, svc_emd_refund_confirm, payment-ext-auth</p> <p>Making changes to a booking: add_ssr, modify_pnr, divide_order</p> <p>CRM:</p> <p>Create refund request: ServiceReqInsert_Input</p> <p>BPC:</p> <p>Registering an order for autorefund: bmmRefund</p>
--	---

POST refund/calculate method

Description: allows split sum for refund for each passenger in booking.

Request format: /api/apiVersion/refund/calculate

Request headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body:

```
{
  "pnrNumber": "string",
  "lastName": "string",
  "passengers": [
    {
      "lastName": "string",
      "passengerReferenceNumber": "string",
      "relatedAdultReferenceNumber": "string",
      "luggageSeats": [
        {
          "referenceNumber": "string"
        }
      ],
      "passengerType": "string"
    }
  ],
}
```

```

    "isInvoluntaryMoneyless": true,
    "isExchangeSegments": boolean
}

```

Request parameters:

Parameter	Description	Comments
pnrNumber	PNR number, unique 6-digit booking identifier	
lastName	(optional) passenger's last name	
passengers	Array of passengers for which refund sums needs to be calculated	
lastName	Passenger's last name	
passengerReferenceNumber	Reference to passenger Linked with data → passengers → reference Number	
relatedAdultReferenceNumber	(optional) reference to the adult passenger with whom the child or infant is flying. The parameter is filled for the child and infant	
luggageSeats	(optional) array with information about luggage with separate air ticket and ssr CBBG.	In developing
referenceNumber	Reference to the service for which a separate air ticket was issued. Related to the field in the response GET Reservation data → luggageSeats → referenceNumber.	In developing
passengerType	Passenger type. Possible values: see EN 8 1 Attachment: Dictionaries of passenger types	
isInvoluntaryMoneyless	Indicates whether the calculation of refundable amounts is for performing POST api/InvoluntaryRefund/Moneyless refund on ADP.	
isExchangeSegments	(optional) sign of segment exchange	

Valid request examples:

Refund for adult passenger:

```

{
  "pnrNumber": "1ZGDLS",

```

```

    "isInvoluntaryMoneyless": false,
    "passengers": [{
      "lastName": "CHEREPANOV",
      "passengerType": 0,
      "passengerReferenceNumber": "12",
      "luggageSeats": []
    }
  ]
}

```

Refund for adult and child:

```

{
  "pnrNumber": "1ZGDWS",
  "passengers": [{
    "lastName": "BYKOV",
    "passengerType": 0,
    "passengerReferenceNumber": "12",
    "luggageSeats": []
  }, {
    "lastName": "BARABANOV",
    "passengerType": 5,
    "passengerReferenceNumber": "14",
    "luggageSeats": [],
    "relatedAdultReferenceNumber": "12"
  }
]
}

```

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": [],
    "serviceRefundDetails": [{
        "code": string,
        "text": string,
        "commissionAmount": integer,
        "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,
    "flightReference": string,
    "isInfantProduct": boolean,
    "bookingId": string,
    "orderId": string,
    "compatibleProductCodes": [string]
}
]
],
"refundWithTicketOnlyServices": [string]
},
],
"errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ]
}
],
"success": boolean
}

```

Response parameters

Parameter	Description	Comments
success	Sign of successful request execution If errors array is empty, the value of field will be 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: see EN 8.1 Attachment: Dictionaries of 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 for refund	
refundBonuses	Number of refunded "Wings" bonuses	
refundCurrency	Amount of roubles for refund	
refundTotal	Total refund amount, sum of refundBonuses and refundCurrency	
ticketNumber	ATC number Ticket number should contain 13 digits	
taxDetailsInformation	Services information about tariffs and fees. Important: Information from present segment can be ignored. RefundBonuses and refundCurrency values should be displayed to user.	

				taxCategory	Tax category For refundable taxes "R" is returned, for non-refundable taxes - "NRT".	
				taxDetails	Array of objects with full information about tariffs and fees details.	
				rate	Tariff/fee sum	
				currencyCode	Currency code	
				type	Tariff/fee type	
				luggageSeatsRefundDetails	Array with information about CBBG luggage with separate air ticket to refund. All array elements are the same with passengerRefundDetails array elements, the field "surname" has value "CBBG".	In developing
				serviceRefundDetails	Array of objects with service's refund information.	
				code	Service code Possible values: see EN 8 1 Attachment: Dictionaries of services types	
				text	Type of refundable pet when code="PETC".	
				commissionAmount	Commission amount for refund of service.	
				refundBonuses	Number of refunded "Wings" bonuses	
				refundCurrency	Number of refunded roubles	
				refundTotal	Total refund amount, sum of refundBonuses and refundCurrency	
				ticketNumber	EMD number Ticket number should contain 13 digits	
				taxDetailsInformation	Services information about tariffs and fees.	
				taxCategory	Tax category. For refundable taxes "R" is returned, for non-refundable taxes - "NRT".	

			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 EN 8 3 Attachment: Partner's services.	
			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 the amount for refund of partner product is unknown.	
			partnerId	Partner's id.	
			partnerType	Partner's type.	
			passengerReference	Reference to passenger for whom partner's service is being refunded	
			refundAmount	Refundable sum.	
			flightReference	Reference to the flight segment, if the partner product wasn't added for the entire flight.	
			orderId	Partner order id	
			compatibleProductCodes	Codes of services of the current partner, which are compatible with this service when purchased it	
			refundWithTicketOnlyServices	Array with service codes that can't be refunded without refunding the ticket.	In developing

errors	Information about errors appeared during query execution.	
ibeCode	Error code, see EN 8 2 Attachment: IBE error codes	
description	Code description, see. EN 8 2 Attachment: IBE error codes	
messages	Array of objects, containing errors from Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode.	
code	Sirena error code	
text	Sirena error text	

POST refund/partial method

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

Request format: /api/apiVersion/refund/partial

Request headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body:

```
{
  "pnrNumber": "string",
  "passengers": [
    {
      "isTicketRefund": boolean,
      "partnerProducts": [
        {
          "partnerId": "string",
          "partnerType": "string",
          "bookingStatus": "string",
          "code": "string",
          "amount": decimal,
          "currency": "string",
          "passengerReference": "string",
          "flightReference": "string",

```

```

    "isInfantProduct": boolean,
    "bookingId": "string",
    "orderId": "string",
    "isIssuedByRefundableOffice": boolean,
    "preselectedOption": {
      "preselect": boolean,
      "passengers": [
        {
          "referenceNumber": "string",
          "passengerType": "string",
          "flightReferences": [
            "string"
          ]
        }
      ]
    },
    "compatibleProductCodes": [
      "string"
    ],
    "internalError": "string",
    "internalErrorMessage": "string",
    "passengerAmounts": [
      {
        "referenceNumber": "string",
        "amount": decimal,
        "currency": "string"
      }
    ]
  },
  ],
  "services": [
    {
      "code": "string",
      "ticketNumber": "string"
    }
  ],
  "lastName": "string",
  "passengerReferenceNumber": "string",
  "relatedAdultReferenceNumber": "string",
  "luggageSeats": [
    {
      "referenceNumber": "string"
    }
  ],
  "passengerType": "string"
}
],
"isExchangeSegments": boolean
}

```

Request parameters:

Parameter	Description	Comments
pnrNumber	PNR number, unique 6-digit booking identifier	

passengers	An array of passengers for which refund sums needs to be calculated	
isTicketRefund	The sign that a ticket with services or only services is being returned	
partnerProducts	(optional) an array of partner services to be returned (the array parameters are similar to the GET api/Reservation response parameters)	
lastName	Passenger's last name	
passengerReferenceNumber	Passenger reference. Associated with field data -> passengers -> referenceNumber	
<i>luggageSeats</i>	(optional) array with information about luggage with separate air ticket and ssr CBBG.	In developing
passengerType	Passenger type. Possible types of passengers are presented in EN 8 1 Attachment: Dictionaries of passenger types .	
relatedAdultReferenceNumber	(optional) reference to the adult passenger with whom the child or infant is flying. The parameter is filled for the child and infant	
services	(optional) an array with information about purchased and refundable additional services of the airline	
code	Returned additional service code	
ticketNumber	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, if autorefund was unsuccessful.

Response parameters (short description):

Parameter	Description	Comments
success	Sign of successful request execution If errors array is empty, the value of field will be 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	Array with service codes that can't be refunded without refunding the ticket.	In developing
errors	Information about errors appeared during query execution.	

Valid request examples:

Example of requesting a full refund of a ticket with additional airline services and partner services:

```
{
  "pnrNumber": "1ZGLB0",
  "lastName": "KUZNETSOV",
  "passengers": [{
    "lastName": "KUZNETSOV",
    "passengerType": 0,
    "passengerReferenceNumber": "12",
    "luggageSeats": [],
    "isTicketRefund": true,
    "partnerProducts": [{
      "partnerId": 7,
      "partnerType": 3,
      "bookingId": "36a0173b-ae6-40e6-8d80-aaf3b154904b",
      "code": "SBERDOC"
    }, {
      "partnerId": 1,
      "partnerType": 1,
      "bookingId": "49753312-18df-4ac0-a68e-7c15ea8fe8b8",
      "code": "AVIADYNAMIC"
    }
  ]
},
],
```

```

        "services": [{
            "ticketNumber": "2624500003235",
            "code": "HTML"
        }
    ]
}

```

Example of request for refund of partner services without a ticket:

```

{
  "pnrNumber": "1ZR0ML",
  "lastName": "TESTOV",
  "passengers": [{
    "lastName": "TESTOV",
    "passengerType": 0,
    "passengerReferenceNumber": "12",
    "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": "1ZRTVO",
  "lastName": "TESTOV",
  "passengers": [{
    "lastName": "TESTOV",
    "passengerType": 0,
    "passengerReferenceNumber": "12",
    "isTicketRefund": true,
    "partnerProducts": [],
    "services": [{
      "ticketNumber": "2624552270079",
      "code": "PETC"
    }
  ]
}
]
}

```

Description of voluntary refund process

Voluntary refund - return at the client's own request to the airline.

Short description of voluntary refund business process.

1. Authorize in BMM (Booking Management Module) using PNR number and passenger's last name
2. Press the button "*Make a return*" → "*Voluntary refund*"
3. Select passengers for who refund is needed to be performed and press the button "*Calculate amount*"
4. POST api/refund/calculate request is sending, IBE calculates amounts for refund, bonuses and penalties
5. Press the button "*Confirm refund*", POST api/refund/partial request is sending
6. IBE performs a refund of services/tickets using Sirena and partner's API and validates the possibility of auto-refund registration
7. IBE registers an order for refund on the original PNR for which the payment was made
8. IBE checks the possibility of auto-refund for current order (using GOI service GET api/refund) and validates the contents of order and the amounts to be refunded
9. IBE sends a refund request to the bank and requests refund's status
10. Further, depending on the refund status:

Variation A: the order is accepted by the bank for return	Variation B: the order isn't accepted for return or an error has been received from the searching machine
<ol style="list-style-type: none"> 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 formalizes the return receipt. The check is sent to the client 3. When client re-enters the BMM, information is displayed that the order has been returned 	<ol style="list-style-type: none"> 1. Realizing automatically annulling seats from the booking (the same process as annulling by POST api/Annul request) 2. The IBE service generates a request for a refund in CRM 3. If the return fails, one of the possible error messages is displayed 4. The IBE service informs the user about a successfully created refund request with information about the request number 5. Within 30 days, the request in CRM is processed manually by the operator, and the amount to be returned is requested from the engine 6. After financial refund operation is performed, confirmation is sent to IBE (information about refund completion is being recorded to DB) 7. Refund completed

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:
 1. Tickets refund information entity;
 2. Airline's services refund information entity;
 3. 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. Complete of the refund session to the final state:
 1. 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.
 2. If refund session does not transfer to terminal state, during allotted time, IBE logs an error and transfers session to ManualAnalysis state.

For retrieving of refund session information and refund confirmation GET refund/pnr and PUT refund/pnr methods of GOI is being used. These methods are described in GOI documentation

Failed request and response examples

1. *Incorrect reference to passenger (14 instead of correct 12):*

Request example	Response example
<pre> { "pnrNumber": "1ZGLL0", "lastName": "TERENTEVA", "passengers": [{ "lastName": "TERENTEVA", "passengerType": 0, "passengerReferenceNumber": "14", "isTicketRefund": true, "luggageSeats": [], "services": [], "partnerProducts": [] }] } </pre>	<pre> { "errors": [{ "ibeCode": 1062, "messages": [{ "text": "Some passenger refund requested for doesn't exist in reservation" }, { "text": "The reservation doesn't have a ticket to refund or the request has an infant ticket to refund" }] }], "success": false } </pre>

2. *Passenger's reference is correct but incorrect passenger type (not 0):*

Request example	Response example
<pre> { "pnrNumber": "1ZGLL0", "lastName": "TERENTEVA", "passengers": [{ "lastName": "TERENTEVA", "passengerType": 1, "passengerReferenceNumber": "12", "isTicketRefund": true, "luggageSeats": [], "services": [], "partnerProducts": [] }] } </pre>	<pre> { "errors": [{ "ibeCode": 1062, "messages": [{ "text": "Some passenger refund requested for doesn't exist in reservation" }, { "text": "The reservation doesn't have a ticket to refund or the request has an infant ticket to refund" }] }], "success": false } </pre>

3. *Booking for 1 adult and 1 child. It is not possible to perform refund only for adult passenger:*

Request example	Response example
<pre>{ "pnrNumber": "1ZGDWS", "lastName": "BYKOV", "passengers": [{ "lastName": "BYKOV", "passengerType": 0, "passengerReferenceNumber": "12", "isTicketRefund": true, "luggageSeats": [], "services": [], "partnerProducts": [] }] }</pre>	<pre>{ "errors": [{ "ibeCode": 1062, "messages": [{ "text": "Some passenger refund requested for is not adult or after refund an infant will be left without adults" }] }], "success": false }</pre>

GET refund method

Description: allows to receive information about already performed refund.

Request format: /api/refund

Request headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request parameters:

Parameter	Description	Comments
pnrNumber	PNR number, unique 6-digit booking identifier	
lastName	Passenger's last name	
lastNameHash	Passenger last name hash	
<i>ticketNumber</i>	Ticket number of passenger	In developing

Response format

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

Response format:

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be true
data	Returned data
pnrNumber	PNR number, unique 6-digit booking identifier
createdDate	Date and time in GMT of refund.
crmRefundRequestId	Request ID in CRM.
errors	Information about errors appeared during query execution.
ibeCode	Error code, see EN 8 2 Attachment: IBE error codes
description	Code description, see. EN 8 2 Attachment: IBE error codes
messages	Array of objects, containing errors from Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode
code	Sirena error code
text	Sirena error text

Example of successful response:

```
{
  "data": [{
    "pnrNumber": "1ZGK8W",
```

```

        "createdDate": "2022-08-23T11:31:54.279264",
        "crmRefundRequestId": "1-5KYKNCR"
    }
],
"errors": [],
"success": true
}

```

6_24_api/reservation/AddContactInfo: contact info adding service (for agents)

API service	Reservation/AddContactInfo
Short description	Allows to add contact info to a passenger with or without a reference
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: modify_pnr

POST Reservation/AddContactInfo method

Description: Allow to add email or phone number to a passenger or add contact info without reference certain passenger

You can see added contact info in the response of GET Reservation/Advanced response

Request format: /api/apiVersion/Reservation/AddContactInfo

Request headers

Parameter	Description
X-API-Key	API key
X-Session	Session ID

Request parameters

Parameter	Description
reservationNumber	PNR of reservation
surname	Passenger's surname
contactInfo	Array with contact info

Parameter	Description
email	Email
phoneNumber	Phone number
passengerReference	Reference to a passenger (optional)

Reponse format

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

Response parameters

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be 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 Sirena response. Array can be empty, it is recommended to analyze errors by ibeCode.
code	Sirena error code
text	Sirena error text

6_25_api/reservation/advanced: service for retrieving SSR and remarks information contained in reservation (for agents)

API service	Reservation/Advanced
Short description	Allows to get all SSRs and remarks (with the RM type) from the booking to view them. Also to put the reservations with the necessary remarks in the queue, and remove the reservations with not interested for agents remarks from the queue
Supported HTTP(S)-methods	GET
Conterminous systems methods used	Sirena: order, pnr_status

GET api/reservation/Advanced method

Note: method is available on ADP

Description: allows to get information about partner's services at the booking using PNR or ticket number

Request format: api/apiVersion/Reservation/Advanced

Request header

Parameter	Description
X-API-Key	API key
X-Session	Session ID

Request parameters

Parameters must be passed: passenger's last name (LastName) + PNR (PnrNumber) or ticket number (TicketNumber)

Parameter	Description
PnrNumber	Reservation number.
TicketNumber	Ticket number, using instead of PNR
LastName	Passenger's last name

Response format

```
{
  "data": {
    "remarks": [
      {
        "type": "string",
        "text": "string",
        "referenceNumber": "string",
        "passengerReferences": [
          "string"
        ],
        "flightReferences": [
          "string"
        ]
      }
    ],
    "specialServiceRequests": [
      {
        "code": "string",
        "status": "string",
        "text": "string",
        "referenceNumber": "string",
        "passengerReferences": [
          "string"
        ],
        "flightReferences": [
          "string"
        ]
      }
    ],
    "contacts": [
      {
        "email": "string",
        "phoneNumber": "string",
        "passengerReference": "string"
      }
    ],
    "tickets": [
      {
        "number": "string",
        "passengerReference": "string",
        "flightReferences": [
          "string"
        ],
        "isInfantTicket": "true",
        "isIssuedByRefundableOffice": "bool",
        "issueDate": "datetime",
        "isVoidAvailable": boolean,
        "voidTimeLimitUtc": "datetime"
      }
    ],
    "outboundFareFamily": {
      "fareFamilyId": "integer",
      "code": "string",
      "shortCode": "string",

```

```

"category": "string",
"isSubsidized": "bool",
"options": [
  {
    "code": "string",
    "category": "string",
    "order": "integer",
    "showInPopup": "bool",
    "showInSummary": "bool",
    "showInUpsell": "bool",
    "showInFareIncluded": "bool",
    "showForAdult": "bool",
    "showForInfant": "bool",
    "showInBookingCart": "bool"
  }
],
"order": "integer",
"commercialFamily": "string",
"canRefund": "bool",
"hasUpsale": "bool",
"isLowCost": "bool"
},
"returnFareFamily": {
  "fareFamilyId": "integer",
  "code": "string",
  "shortCode": "string",
  "category": "string",
  "isSubsidized": "bool",
  "options": [
    {
      "code": "string",
      "category": "string",
      "order": "integer",
      "showInPopup": "bool",
      "showInSummary": "bool",
      "showInUpsell": "bool",
      "showInFareIncluded": "bool",
      "showForAdult": "bool",
      "showForInfant": "bool",
      "showInBookingCart": "bool"
    }
  ],
  "order": "integer",
  "commercialFamily": "string",
  "canRefund": "bool",
  "hasUpsale": "bool",
  "isLowCost": "bool"
},
"separateFareFamily": {
  "fareFamilyId": "integer",
  "code": "string",
  "shortCode": "string",
  "category": "string",
  "isSubsidized": "bool",
  "options": [
    {
      "code": "string",

```

```

        "category": "string",
        "order": "integer",
        "showInPopup": "bool",
        "showInSummary": "bool",
        "showInUpsell": "bool",
        "showInFareIncluded": "bool",
        "showForAdult": "bool",
        "showForInfant": "bool",
        "showInBookingCart": "bool"
    }
],
"order": "integer",
"commercialFamily": "string",
"canRefund": "bool",
"hasUpsale": "bool",
"isLowCost": "bool"
},
"tsmData": [
    {
        "passengerReference": "string",
        "price": "integer",
        "referenceType": "string",
        "referenceNumber": "string"
    }
],
"outboundBonus": {
    "bonus": "string",
    "replacedBonus": "string"
},
"returnBonus": {
    "bonus": "string",
    "replacedBonus": "string"
},
"appliedDiscounts": [
    {
        "flightReference": "string",
        "amount": "integer",
        "discountAmount": "integer",
        "id": "string",
        "type": "string",
        "passengerReference": "string",
        "serviceCode": "string",
        "promoNumber": "string",
        "serviceReference": "string",
        "usageType": "string",
        "promotionType": "string"
    }
],
"baseFarePrices": [
    {
        "currency": "string",
        "passengerReference": "string",
        "flightReferences": [
            "string"
        ],
        "isInfantFare": "bool",
        "totalPrice": "integer",

```

```

    "basePrice": "integer",
      "separatePrice": "integer",
    "outboundPrice": "integer",
    "returnPrice": "integer",
    "discountAmount": "integer",
      "separateDiscount": "integer",
    "outboundDiscount": "integer",
    "returnDiscount": "integer",
    "priceHidden": "bool",
    "taxDetailsInformation": [
      {
        "taxCategory": "string",
        "taxDetails": [
          {
            "rate": "string",
            "currencyCode": "string",
            "type": "string"
          }
        ]
      }
    ]
  ],
  "spentBonusesByPassengers": [
    {
      "amount": "integer",
      "commissionAmount": "integer",
      "passengerReference": "string"
    }
  ],
  "appliedCertificates": [
    {
      "amount": "integer",
      "number": "string"
    }
  ],
  "fareBasisCodesFlightRefs": [
    {
      "fareBasis": "string",
      "flightReferenceNumber": "string",
      "passengerReferenceNumber": "string",
      "expirationUtc": "datetime"
    }
  ],
  "familyTrip": "bool",
  "originOfficeId": "string",
  "originOfficeType": "string",
  "isPayForbiddenForAgency": "bool",
  "foodOnBoards": [
    {
      "flightNumber": "string",
      "departureAirport": "string",
      "arrivalAirport": "string",
      "commercialFamily": "string",
      "translations": {
        "additionalProp1": "string",
        "additionalProp2": "string",

```

```

        "additionalProp3": "string"
    },
    "ru": "string",
    "en": "string",
    "zh": "string"
}
],
"reservationRetrieveStatus": "string",
"ticketCouponStatuses": [
    {
        "ticketNumber": "string",
        "couponStatuses": {
            "additionalProp1": "string",
            "additionalProp2": "string",
            "additionalProp3": "string"
        },
        "couponStatusesBySegment": {
            "additionalProp1": "string",
            "additionalProp2": "string",
            "additionalProp3": "string"
        }
    }
],
"engineType": "string",
"isGroupReservation": "bool",
"isCompanyGroupReservation": "bool",
"isReceiptAvailable": "bool",
"isInvoluntaryRefundAvailable": "bool",
"isVoluntaryRefundAvailable": "bool",
"informationalServices": [
    {
        "code": "string",
        "status": "string",
        "text": "string",
        "petType": "string",
        "category": "string",
        "fareAmount": integer,
        "passengerQualifier": "string",
        "passengerReference": "string",
        "flightReference": "string",
        "referenceNumber": "string",
        "packReferenceNumber": "string",
        "ticketNumber": "string",
        "finalPrice": integer,
        "price": integer,
        "discountAmount": integer,
        "amadeusErrorCode": "string",
        "amadeusErrorMessage": "string",
        "internalError": "string",
        "internalErrorMessage": "string",
        "issueDate": "datetime",
        "issuedByThirdPartyOffice": "bool",
        "isIssuedByRefundableOffice": "bool",
        "serviceDuration": integer,
        "expireDateUtc": "datetime"
    }
],

```

```

"journey": {
  "outboundFlights": [
    {
      "aircraft": "string",
      "aircraftIataCode": "string",
      "referenceNumber": "string",
      "origin": "string",
      "destination": "string",
      "departureDate": "datetime",
      "arrivalDateUtc": "datetime",
      "departureDateUtc": "datetime",
      "arrivalDate": "datetime",
      "flightNumber": "string",
      "classOfService": "string",
      "commercialFamily": "string",
      "operatingCarrier": "string",
      "marketingCarrier": "string",
      "codeshareName": "string",
      "flightDuration": "string",
      "stopDuration": "string",
      "techStops": [
        {
          "arrivalDateTime": "datetime",
          "departureDateTime": "datetime",
          "airportCode": "string",
          "stopDuration": "string",
          "flightDurationUntilNextStop": "string"
        }
      ],
      "flightDurationUntilNextStop": "string",
      "arrivalTerminal": "string",
      "departureTerminal": "string",
      "hash": "string",
      "statuses": [
        "string"
      ],
      "isCancelledByAirline": "bool",
      "isCheckinWithoutPseudo": "bool"
    }
  ],
  "returnFlights": [
    {
      "aircraft": "string",
      "aircraftIataCode": "string",
      "referenceNumber": "string",
      "origin": "string",
      "destination": "string",
      "departureDate": "datetime",
      "arrivalDateUtc": "datetime",
      "departureDateUtc": "datetime",
      "arrivalDate": "datetime",
      "flightNumber": "string",
      "classOfService": "string",
      "commercialFamily": "string",
      "operatingCarrier": "string",
      "marketingCarrier": "string",
      "codeshareName": "string",

```

```

    "flightDuration": "string",
    "stopDuration": "string",
    "techStops": [
      {
        "arrivalDateTime": "datetime",
        "departureDateTime": "datetime",
        "airportCode": "string",
        "stopDuration": "string",
        "flightDurationUntilNextStop": "string"
      }
    ],
    "flightDurationUntilNextStop": "string",
    "arrivalTerminal": "string",
    "departureTerminal": "string",
    "hash": "string",
    "statuses": [
      "string"
    ],
    "isCancelledByAirline": "bool",
    "isCheckinWithoutPseudo": "bool"
  }
],
"separateFlights": [
  {
    "aircraft": "string",
    "aircraftIataCode": "string",
    "referenceNumber": "string",
    "origin": "string",
    "destination": "string",
    "departureDate": "datetime",
    "arrivalDateUtc": "datetime",
    "departureDateUtc": "datetime",
    "arrivalDate": "datetime",
    "flightNumber": "string",
    "classOfService": "string",
    "commercialFamily": "string",
    "operatingCarrier": "string",
    "marketingCarrier": "string",
    "codeshareName": "string",
    "flightDuration": "string",
    "stopDuration": "string",
    "techStops": [
      {
        "arrivalDateTime": "datetime",
        "departureDateTime": "datetime",
        "airportCode": "string",
        "stopDuration": "string",
        "flightDurationUntilNextStop": "string"
      }
    ],
    "flightDurationUntilNextStop": "string",
    "arrivalTerminal": "string",
    "departureTerminal": "string",
    "hash": "string",
    "statuses": [
      "string"
    ],
  },

```

```

        "isCancelledByAirline": "bool",
        "isCheckinWithoutPseudo": "bool"
    }
],
"outboundFareFamily": {
    "code": "string",
    "category": "string",
    "isSubsidized": "bool",
    "isLowCost": "bool",
    "isUnaccompaniedMinor": "bool"
},
"returnFareFamily": {
    "code": "string",
    "category": "string",
    "isSubsidized": "bool",
    "isLowCost": "bool",
    "isUnaccompaniedMinor": "bool"
},
"separateFareFamily": {
    "code": "string",
    "category": "string",
    "isSubsidized": "bool",
    "isLowCost": "bool",
    "isUnaccompaniedMinor": "bool"
},
"outboundDuration": "string",
"returnDuration": "string",
    "separateDuration": "string"
"origin": "string",
"destination": "string",
"departureDate": "datetime",
"returnDate": "datetime",
"outboundBaggageAllowances": [
    {
        "quantity": "integer",
        "weight": "integer",
        "passengerType": "string",
        "fareCode": "string"
    }
],
"returnBaggageAllowances": [
    {
        "quantity": "integer",
        "weight": "integer",
        "passengerType": "string",
        "fareCode": "string"
    }
],
"separateBaggageAllowances": [
    {
        "quantity": "integer",
        "weight": "integer",
        "passengerType": "string",
        "fareCode": "string"
    }
],

```

```

    "containsInternationalFlights": "bool"
  },
  "number": "string",
  "reservationDate": "datetime",
  "bookingTimeLimit": "datetime",
  "passengers": [
    {
      "id": "integer",
      "hasInfant": "bool",
      "referenceNumber": "string",
      "relatedAdultReferenceNumber": "string",
      "relatedInfantReferenceNumber": "string",
      "relatedChildrenReferenceNumbers": [
        "string"
      ],
      "disability": {
        "isDeaf": "bool",
        "isBlind": "bool",
        "hasGuideDog": "bool",
        "assistanceType": "string",
        "wheelchairType": "string",
        "commentary": "string"
      },
      "firstName": "string",
      "middleName": "string",
      "surname": "string",
      "passengerType": "string",
      "birthDate": "datetime",
      "sex": "string",
      "nationality": "string",
      "documentNumber": "string",
      "documentExpiryDate": "datetime",
      "loyaltyCard": "string",
      "email": "string",
      "phone": "string",
      "documentType": "string",
      "isAdult": "bool",
      "isChanged": "bool",
      "checkedIn": {
        "additionalProp1": "bool",
        "additionalProp2": "bool",
        "additionalProp3": "bool"
      },
      "checkInHasStarted": {
        "additionalProp1": "bool",
        "additionalProp2": "bool",
        "additionalProp3": "bool"
      },
      "checkedInSegment": {
        "additionalProp1": "bool",
        "additionalProp2": "bool",
        "additionalProp3": "bool"
      },
      "checkInHasStartedSegment": {
        "additionalProp1": "bool",
        "additionalProp2": "bool",
        "additionalProp3": "bool"
      }
    }
  ]
}

```

```

    },
    "hasPassengersDataErrors": "bool",
    "passengerDataErrors": [
      {
        "key": "string",
        "value": "string"
      }
    ],
    "endorsements": [
      {
        "endorsement": "bool",
        "isDisability": "bool",
        "endorsementDocNumber": "string",
        "endorsementDocDate": "datetime",
        "loyaltyPartnerCard": "string"
      }
    ],
    "accompanimentBookingsInfo": [
      {
        "number": string,
        "surname": string
      }
    ]
  ],
  "services": [
    {
      "code": "string",
      "status": "string",
      "text": "string",
      "petType": "string",
      "category": "string",
      "fareAmount": "integer",
      "passengerQualifier": "string",
      "passengerReference": "string",
      "flightReference": "string",
      "referenceNumber": "string",
      "packReferenceNumber": "string",
      "ticketNumber": "string",
      "finalPrice": "integer",
      "price": "integer",
      "discountAmount": "integer",
      "amadeusErrorCode": "string",
      "amadeusErrorMessage": "string",
      "internalError": "string",
      "internalErrorMessage": "string",
      "issueDate": "datetime",
      "issuedByThirdPartyOffice": "bool",
      "isIssuedByRefundableOffice": "bool"
    }
  ],
  "seats": [
    {
      "number": "integer",
      "title": "string",
      "code": "string",
      "status": "string",

```

```

    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": "integer",
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": "integer",
    "price": "integer",
    "discountAmount": "integer",
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "string",
    "internalErrorMessage": "string",
    "issueDate": "datetime",
    "issuedByThirdPartyOffice": "bool",
    "isIssuedByRefundableOffice": "bool"
  }
],
"luggageSeats": [
  {
    "referenceNumber": "string",
    "passengerReference": "string"
  }
],
"insurances": [
  {
    "number": "string",
    "passengerType": "string",
    "code": "string",
    "status": "string",
    "text": "string",
    "petType": "string",
    "category": "string",
    "fareAmount": "integer",
    "passengerQualifier": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "referenceNumber": "string",
    "packReferenceNumber": "string",
    "ticketNumber": "string",
    "finalPrice": "integer",
    "price": "integer",
    "discountAmount": "integer",
    "amadeusErrorCode": "string",
    "amadeusErrorMessage": "string",
    "internalError": "string",
    "internalErrorMessage": "string",
    "issueDate": "datetime",
    "issuedByThirdPartyOffice": "bool",
    "isIssuedByRefundableOffice": "bool"
  }
],
"partnerProducts": [

```

```

{
  "partnerId": "string",
  "partnerType": "string",
  "bookingStatus": "string",
  "code": "string",
  "amount": "integer",
  "currency": "string",
  "passengerReference": "string",
  "flightReference": "string",
  "isInfantProduct": "bool",
  "bookingId": "string",
  "orderId": "string",
  "isIssuedByRefundableOffice": "bool",
  "preselectedOption": {
    "preselect": "bool",
    "passengers": [
      {
        "referenceNumber": "string",
        "passengerType": "string"
      }
    ]
  },
  "compatibleProductCodes": [
    "string"
  ],
  "internalError": "string",
  "internalErrorMessage": "string"
}
],
"ticketFares": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": "integer",
    "totalAmount": "integer",
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ]
  },
  "passengerType": "string"
}
],
"ticketFaresWithoutZapOff": [
  {
    "referenceNumber": "string",
    "referenceQualifier": "string",
    "baseAmount": "integer",
    "totalAmount": "integer",
    "currency": "string",
    "passengerReferences": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string"
      }
    ]
  }
]

```

```

    ],
    "passengerType": "string"
  }
],
"hasSpecialConditions": "bool",
"isRefundAvailable": "bool",
"annulUnavailabilityReason": "string",
"isAnnulAvailable": true,
"isAgentReservation": true,
"accountCode": "string",
"tourCode": "string"
},
"errors": [
  {
    "ibeCode": "integer",
    "description": "string",
    "messages": [
      {
        "code": "string",
        "text": "string",
        "additionalInfo": {
          "ssrType": "string",
          "info": "string"
        }
      }
    ]
  }
]
},
],
"success": "bool"
}

```

Contacts array

Parameter	Description
contacts	Returned data
email	Email
phoneNumber	Phone number
passengerReference	Reference to a passenger

Remarks array

Parameter	Description
remarks	Returned data
type	Remark type (tag <type>)
text	Content of tag freeText
referenceNumber	Reference to a service

Parameter	Description
passengerReferences	Reference to a passenger (if exists)
flightReferences	Reference to a segment (is exist)

specialServiceRequests array

Parameter	Description
specialServiceRequests	Returned data
code	SSR code (tag <type>)
status	Status
text	Content of tag freeText
referenceNumber	Reference to a service
passengerReferences	Reference to a passenger (if exists)
flightReferences	Reference to a segment (if exists)

informationalServices array

Parameter	Description
informationalServices	An array with informational about info-services
code	Additional service code Possible values: reference codes of the add. services
status	The status of adding the service. Example: HD service successfully added
text	Text in ssr format for sending information. Example: DOG 1PC 8KG 45X35X25
petType	Type of pet. Possible values: DOG, CAT, PARROT, CANARY
category	Category of additional service Possible values: Reference codes add. services
price	The cost of additional service

Parameter	Description
issueDate	Date and time of service release
issuedByThirdPartyOffice	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued
isIssuedByRefundableOffice	Sign that the return of the service is not possible, since it is added by a third-party office
referenceNumber	SSR number. There is an unique SSR number for each service in PNR
passengerReferences	Reference to the passenger (if exists)
flightReferences	Reference to the segment (if exists)

tickets array (in case of request via ticket number)

Parameter	Description
tickets	An array with informational about tickets
number	Ticket number
passengerReference	Reference to the passenger
flightReferences	Reference to the segment
isInfantTicket	Sign of infant ticket
isIssuedByRefundableOffice	Sign that the return of the ticket is not possible, since it is added by a third-party office
issueDate	Date and time of ticket release
isVoidAvailable	Sign of ticket void availability
voidTimeLimitUtc	Date and time until ticket void is possible

6_26_api/reservation/AddSSR, api/reservation/AddRemark: services for adding SSR and remarks to reservation (for agents)

API service	Reservation/AddSSR, Reservation/ AddRemark
Short description	Service for add SSR and remarks to reservation via API IBE
Supported HTTP(S)-methods	POST

POST api/Reservation/AddSSR method

Note: method is available on ADP

Description: add SSR to the reservation

You can see added SSR in GET Reservation/Advanced response

Response format: api/apiVersion/Reservation/AddSSR

Response headers

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body

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

Request parameters

Parameter	Description
reservationNumber	Unique booking number (PNR)
surname	Passenger's surname

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, CTCR
passengerReference	Passenger's reference
segmentReference	Flight segment reference
freeText	Text

Response format

```

{
  "data": {
    "ssr": [
      {
        "code": string,
        "status": string,
        "text": string,
        "referenceNumber": string,
        "passengerReferences": [
          string
        ],
        "flightReferences": [
          string
        ]
      }
    ]
  },
  "errors": [
    {
      "ibeCode": integer,
      "description": string,
      "messages": [
        {
          "code": string,
          "text": string
        }
      ]
    }
  ],
  "success": bool
}

```

Response parameter

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

data	Returned data
ssr	Array of SSRs that were added to the reservation
code	SSR code
status	SSR status
text	freeText of SSR
referenceNumber	SSR reference number
passengerReference	Passenger's reference number
flightReference	Flight segment reference number
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 Sirena response. Array can be empty, it is recommended to analyze errors by ibeCode
code	Sirena error code
text	Sirena error text

Example of valid request

Request for adding SSR of transporting weapon and bike:

```
{
  "reservationNumber": "28NVFV",
  "surname": "Ivanov",
  "ssrs": [
    {
      "code": "WEAP",
      "passengerReference": "14",
      "segmentReference": "16",
      "freeText": "Weapon transporting"
    },
    {
      "code": "BIKE",
      "passengerReference": "14",
      "segmentReference": "16",
      "freeText": "Bike transporting"
    }
  ]
}
```

POST api/Reservation/AddRemark method

Note: method is available on ADP

Description: add remark to the reservation

Response format: api/apiVersion/Reservation/AddRemark

Response headers:

Parameter	Description
X-Api-Key	API key
X-Session	Session ID

Request body

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

Request parameters:

Parameter	Description
reservationNumber	Unique booking number (PNR)
surname	Passenger's surname
remarks	Remark information object
passengerReference	Passenger's reference
segmentReference	Flight segment reference
freeText	Text

Example of valid request:

```

{
  "reservationNumber": "28NVFV",
  "remarks": [
    {
      "passengerReference": "2",
      "segmentReference": "1",
      "freeText": "Here is free remark text"
    }
  ]
}

```

6_27_api/ticket/quota: service for retrieving information about ticket issuance quotas

API service	Ticket/Quota
Short description	Allows querying information about the current number of ticket blanks available for issuance, as well as details about the deposit (if applicable).
Supported HTTP(S)-methods	GET
Methods of related systems used	Sirena: get_et_quota

GET Ticket/Quota method

Description: Returns up-to-date information about the ticket issuance quotas and deposit (if applicable) for the PPR from which the request is submitted.

Request format: /api/apiVersion/Ticket/Quota

Request parameters: does not need to obtain a session ID in advance. API key (X-Api-Key) is necessary

Response format

```

{
  "data": {
    "quota": string,
    "deposit": string
  },
  "errors": [],
  "success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of successful query execution If errors array is empty, the value of field will be true
data	Returned data
quota	The number of available ticket blanks. It can be UNLIMITED, UNKNOWN, or a numerical value.
deposit	Deposit. It can be UNLIMITED, UNKNOWN, or a numerical value.
currency	Currency of the deposit (e.g., RUB).
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	Error code
text	Error text

Example of a successful response

```
{
  "data": {
    "quota": "UNLIMITED",
    "deposit": "UNLIMITED"
  },
  "errors": [],
  "success": true
}
```

Example of an unsuccessful request

```
{
  "errors": [
    {
      "ibeCode": 40230,
      "description": "Get ticket quota request failed, try later.",
      "messages": []
    }
  ]
}
```

```

    ],
    "success": false
}

```

6_32_api/SubsidiesConfirmationSession: service for getting the information about the status of creation of a subsidized reservation v2.3

API service	api/SubsidiesConfirmationSession
Short description	Service for getting the information about the status of creation of a subsidized reservation with passengers with disabilities or passengers with Kaliningrad's subsidies and Far Eastern Federal District, when a right for using the subsidy should be checked and confirmed.
Supported HTTP(S)-methods	GET, DELETE
Conterminous systems methods used	-

GET api/SubsidiesConfirmationSession method

Description: let to check information about the status of creating a subsidized reservation (for creating reservations at subsidized fares for passengers with disabilities and reservations with Kaliningrad's subsidies and Far Eastern Federal District).

Request format: /api/apiVersion/SubsidiesConfirmationSession

Request parameters

Parameter	Description
sessionId	Id of the session in which the subsidized reservation confirmation waiting session is performed
X-Api-Key	X-Api-Key - unique GUID of system using IBE API

Response format

```

{
  "data": {
    "pnrNumber": string,
    "status": integer,
    "subsidiesType": "string",
    "flights": [{
      "origin": string,
      "destination": string,

```

```

        "departureDate": datetime,
        "flightNumber": string
    }
],
"passengers": [{
    "id": integer,
    "firstName": string,
    "middleName": string,
    "surname": string,
    "passengerType": integer,
    "birthDate": datetime,
    "sex": integer,
    "nationality": string,
    "accompaniedPassengerId": integer,
    "errorCode": string,
    "status": string
}
],
"errorCode": string,
"bmmUrl": 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 successful request execution If errors array is empty, the value of field will be true
data	Returned data
pnrNumber	The pnr number of the created reservation (returned if the verification was successful and the reservation was created)

Parameter	Description
status	Reservation creation status See Attachments: dictionary of the statuses of sessions for creating subsidized reservations with confirmation
subsidiesType	Subsidy type (for example, Disability) See Attachments: dictionary of subsidy types
flights	An array of objects with information about the flights on which the booking was created
origin	The airport of departure
destination	Arrival airport
departureDate	Departure date
flightNumber	Flight number
passengers	An array of objects with information about passengers
id	The passenger's number from the booking
firstName	The passenger's name
middleName	Middle name of the passenger (may be missing)
surname	The passenger's last name
passengerType	The type of passenger
birthDate	Date of birth
sex	The gender of the passenger
nationality	Passenger's citizenship (country)
accompaniedPassengerId	The number (id) of the accompanied passenger. It's returned for accompanying passengers
errorCode	The error code returned when checking the passenger. It's returned if the check of the subsidized passenger ended with an error
status	Subsidized passenger verification status code. See Attachments: Attachment of subsidized passenger verification statuses
errorCode	The resulting error code for checking all passengers

Parameter	Description
bmmUrl	The link to BMM of the created reservation. It's returned if the booking has been successfully created
errors	Information about errors that occurred during the operation of the service
ibeCode	Error code
description	Error description
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

Examples of responses

The example of a successful response when a booking has been successfully created:

```
{
  "data": {
    "pnrNumber": "24GC8V",
    "status": 2,
    "subsidiesType": "Age",
    "flights": [
      {
        "origin": "DME",
        "destination": "HTA",
        "departureDate": "2024-03-22T19:40:00Z",
        "flightNumber": "93"
      },
      {
        "origin": "HTA",
        "destination": "DME",
        "departureDate": "2024-03-29T09:20:00Z",
        "flightNumber": "94"
      }
    ],
    "passengers": [
      {
        "id": 1,
        "firstName": "Лариса",
        "middleName": "Адамовна",
        "surname": "Ханипова",
        "passengerType": 25,
        "birthDate": "1956-05-25T00:00:00",
        "sex": 1,
        "nationality": "RU",
        "status": 2
      }
    ]
  }
}
```

```

    ],
    "bmmUrl": "https://sirena-bmm-
stage.uralairlines.ru/?pnr=24GC8V&lastName=Ханипова"
  },
  "errors": [],
  "success": true
}

```

The example of a successful response when the verification of subsidized passengers failed:

```

{
  "data": {
    "status": 3,
    "flights": [{
      "origin": "DME",
      "destination": "HTA",
      "departureDate": "2024-03-23T19:40:00Z",
      "flightNumber": "93"
    }
  ],
  "passengers": [{
    "id": 1,
    "firstName": "Мила",
    "middleName": "Андреевна",
    "surname": "Носкова",
    "passengerType": 23,
    "birthDate": "2011-10-20T00:00:00",
    "sex": 1,
    "nationality": "RU",
    "errorCode": 9111,
    "status": 3
  }
  ],
  "errorCode": 9111
},
"errors": [],
"success": true
}

```

DELETE api/SubsidiesConfirmationSession method

Description: let to cancel the confirmation session for creating a subsidized reservation (to create reservations at subsidized fares for passengers with disabilities)

Request format /api/apiVersion/SubsidiesConfirmationSession

Request parameters

Parameter	Description	Mandatory
sessionId	Id of the session in which the subsidized reservation confirmation waiting session is performed and which need to cancel.	Yes
X-Api-Key	X-Api-Key - unique GUID of system using IBE API	Yes

Response format

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

Response parameters

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be true
errors	Information about errors that occurred during the operation of the service
ibeCode	Error code
description	Error description
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

Example of a successful response

```
{
  "errors": [],
```

```

    "success": true
}

```

6_33_api/passenger/AddFiles: service for submitting attachments when creating a subsidy application for Kaliningrad, the Far Eastern Federal District, and Pensioners/Youth.

API service	Passenger/AddFiles
Short description	<p>A service for submitting attachments that confirm eligibility for a subsidy. Depending on the subsidy type:</p> <ul style="list-style-type: none"> • For Kaliningrad and Far Eastern Federal District subsidies, files are sent to the CRM system to verify eligibility. • For Youth/Pensioners subsidies, files are uploaded to an FTP server for document photo recognition.
Supported HTTP(S)-methods	POST
Methods of related systems used	<p>Sirena: order, add_remark, pnr_status</p> <p>CRM: ServiceRequestAttachment</p>

Method POST api/Passenger/AddFiles

Description:

At the process of booking for passengers eligible for Kaliningrad and Far Eastern Federal District subsidies, it is necessary to attach documents confirming the right to the subsidy.

When creating a booking for passengers eligible for the Kaliningrad and Far Eastern Federal District subsidies, supporting documents confirming the right to the subsidy must be attached. A request is created in the CRM, and the files are attached for review and verification. Once the right is confirmed on the CRM side, a booking number is assigned that is available for release.

When creating a booking for passengers with Pensioners/Youth subsidies, it is necessary to attach documents confirming the right to a subsidy. After the booking is created, the uploaded files are verified: scanned documents are checked for consistency with the passenger data provided in the request. If at least one passenger's data does not match, the booking is cancelled and an error code describing the discrepancy is returned to the API. If verification is successful,

the files are stored on the airline's FTP server for reporting purposes, and the booking becomes available for issuance.

Each subsidy type has its own list of required supporting documents. A maximum of 5 attachments per passenger is allowed. The total file size must not exceed 2.5 MB. Supported file formats: JPG, JPEG, PNG, PDF.

Request format /api/apiVersion/Passenger/AddFiles

The request includes ConfirmationSessionId (session ID from the GET SubsidiesConfirmationSession request) and details about the attached files.

- For Kaliningrad and Far Eastern Federal District subsidies, providing the session ID is mandatory.
- For Pensioners/Youth subsidies, session ID is not required.

Request header

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

Response format

```
{
  "data": {
    "isFilesAdded": boolean
  },
  "errors": [],
  "success": boolean
}
```

Response parameters

Parameter	Description
success	Sign of successful request execution If the errors array is empty, the value of field will be true
data	The data returned by the request
isFilesAdded	Parameter that all attached files were sent with the request
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 Sirena answer. The array may be empty, it is recommended to analyze the error by the ibeCode field

Parameter	Description
code	Sirena error code
text	Sirena error text

Response example

Succeed sending files with the request from approving a subsidy:

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

6_35_api/reservation/partners: obtain issued partner's services information service

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

GET api/reservation/partners method

Description: allows to obtain an information about partner's service in reservation

Request headers

Parameter	Description
X-Session	Session key

X-Api-Key	API key
-----------	---------

Request format: /api/apiVersion/reservation/partners/{bookingId}/details, where bookingId - unique ID of partner's service in current reservation.

Value can be obtained from the response to the request `api/services` in `partnerProducts` object:

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

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
    }]
  }
],
  "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
bookingId	Unique ID of partner's service in IBE. Matches value from request
description	Description of partner's service For example: "СТАНДАРТ" for STANDARTOW Aeroexpress tariff. Or "ТРЭБЕЛ" 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
flightReferences	Array of references to flights to which present service is added
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 Alfainsurance partner:

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

Examples of unsuccessful requests and responses:

Invalid bookingID sent:

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

POST api/reservation/partners/send method

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

Request format: /api/apiVersion/reservation/partners/{bookingId}/send, where bookingId - unique ID of partner's service in current reservation.

Request parameters are absent

Request headers:

Parameter	Description
X-Session	Session key
X-API-Key	API key

Response format:

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

Response example:

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

Examples of unsuccessful requests and responses:

Invalid bookingID sent:

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

Attempt to resend not issued ticket:

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

6_36_api/reservation/ArchivedBookingOrders: service of information about archived booking

API service	reservation/ArchivedBookingOrders
Short description	Allows to get information from archived booking
Supported HTTP(S)-methods	GET
Conterminous systems methods used	CRM: SBRBookingOrder

GET reservation/ArchivedBookingOrders method

Description: requests booking information even if it archived, passing PNR or ticket number

Request format: /api/v2.3/Reservation/ArchivedBookingOrders

Request headers: X-API-Key - unique GUID of system using IBE API

Request parameters:

Only PNR or ticket number may be transmitted. LastName parameter is optional.

Parameter	Description
PnrNumber	PNR number
TicketNumber	Ticket number
LastName	(Optional) passenger's surname

Response format:

```
{
  "data": {
    "id": "string",
    "email": "string",
    "formOfPayment": "string",
    "officeId": "string",
    "pnr": "string",
    "passengers": [
      {
        "fareBasis": {},
        "origAmount": string,
        "passengerNumber": integer,
        "ticketNumber": "string",
        "sex": integer,
        "firstName": "string",
```

```

        "surname": "string",
        "passengerDocument": "string",
        "cardNumber": "string",
        "birthDate": "datetime"
    }
],
"purchaseDate": "datetime",
"phone": "string",
"routes": [
    {
        "aircraft": "string",
        "arrivalDate": "datetime",
        "bookingClass": "string",
        "carrier": "string",
        "departureDate": "datetime",
        "destinationCityRus": "string",
        "destinationCode": "string",
        "flight": "string",
        "originationCityRus": "string",
        "originationCode": "string"
    }
]
},
"errors": [],
"success": boolean
}

```

Response parameters

Parameter	Description
success	Sign of successful request execution If errors array is empty, the value of field will be true
data	Array with order information
id	Order ID Example: 4-LFJGR
email	Passenger's e-mail address
formOfPayment	Form and sum of payment Example: VI28MG8BUVBRR/RUB6336
officeId	Office ID
pnr	PNR number of reservation
passengers	Array with passenger data
fareBasis	Array of objects with base fare information

	origAmount	Paid amount
	passengerNumber	Passenger number
	ticketNumber	Ticket number Example: 262 2400155172
	sex	Passenger's gender Possible values: see Attachment: dictionaries
	firstName	Passenger name
	surname	Passenger surname
	passengerDocument	Passenger document number
	cardNumber	Passenger card number
	birthDate	Passenger date of birth Format: YYYY-MM-DD
	purchaseDate	Order date Format: YYYY-MM-DDTHH:MM:SS
	phone	Passenger phone number
	routes	Array with route information
	aircraft	Aircraft's model
	arrivalDate	Arrival date Format: YYYY-MM-DDTHH:MM:SS
	bookingClass	Class of booking. Commonly it's a first symbol of fareBasis
	carrier	Carrier Example: U6
	departureDate	Departure date Format: YYYY-MM-DDTHH:MM:SS
	destinationCityRus	Destination city Example: Екатеринбург
	destinationCode	IATA-code of destination airport

	flight	Flight number
	originationCityRus	Origin city Example: Москва
	originationCode	IATA-code of origin airport
errors		Information about errors appeared during query execution

6_38_api/pricing: fares confirmation (pricing) service

API service	pricing
Short description	Pricing fares amount in current reservation session
Supported HTTP(S)-methods	POST
Conterminous system methods used	Sirena: Order, PnrStatus

Description: allows to pricing cost of fares in reservation

Request format /api/apiVersion/pricing

Request parameter: X-Session - session ID

Request body

```
{
  "outboundFareFamily": {
    "code": "string",
    "category": "string",
    "isLowCost": boolean,
    "isUnaccompaniedMinor": boolean
  },
  "returnFareFamily": {
    "code": "string",
    "category": "string",
    "isLowCost": boolean,
    "isUnaccompaniedMinor": boolean
  },
  "fares": [
    {
      "passengerType": "string",
      "flightIndex": integer,
      "flightDirection": integer,
      "fareBasis": "string",
      "isBreakpoint": boolean,
      "code": "string"
    }
  ]
}
```

```

],
"sessionMarker": "string",
"currency": "string",
"outboundDiscountId": "string",
"returnDiscountId": "string"
}

```

Request parameters

Parameter	Description
outboundFareFamily	Array with information about fares of outbound destination
code	Fare family code Example: ECOLIGHT, ECOFLEX, ECOPREM Possible values: see Dictionaries: fares
category	Fare family category Example: PR, EC, FL Possible values: see Dictionaries: fares
isLowCost	(optional) sign of low-cost fare
isUnaccompaniedMinor	(optional) sign of unaccompanied minor
returnFareFamily	Array with information about fares of return destination
code	Fare family code Example: ECOLIGHT, ECOFLEX, ECOPREM Possible values: see Dictionaries: fares
category	Fare family category Example: PR, EC, FL Possible values: see Dictionaries: fares
isLowCost	(optional) sign of low-cost fare
isUnaccompaniedMinor	(optional) sign of unaccompanied minor
fares	Array of objects with information about chosen fares
passengerType	Passenger type
flightIndex	Flight segment number

flightDirection	Flight direction Example: 0 - outbound, 1 - return
fareBasis	Name of base fare Example: WPROW или WECOW
isBreakpoint	Sign if fare cancels its action on current segment
code	Fare family code Example: ECOLIGHT, ECOFLEX, ECOPREM Possible values: see Dictionaries: fares
sessionMarker	Session marker
currency	Pricing currency
outboundDiscountId	(optional) discount amount for outbound flight
returnDiscountId	(optional) discount amount for return flight

Response format

```
{
  "data": {
    "ticketFares": [
      {
        "referenceNumber": "string",
        "referenceQualifier": "string",
        "baseAmount": decimal,
        "totalAmount": decimal,
        "currency": "string",
        "passengerReferences": [
          {
            "referenceNumber": "string",
            "referenceQualifier": "string"
          }
        ],
        "passengerType": "string"
      }
    ],
    "fareZapOff": {
      "uniquePromotionNumber": "string",
      "amount": decimal,
      "discountSsr": "string",
      "discountCardSsr": "string"
    }
  },
  "errors": [
    {
      "ibeCode": "string",
```

```

    "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
ticketFares	Information about confirmed prices by passengers
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
currency	Pricing currency
passengerReferences	Array of objects with information about passengers for which current ticket price is actual
referenceNumber	Passenger reference
referenceQualifier	Reference type Possible values: PA – for adults, PI – for infants
fareZapOff	(optional) array with information about discounts in reservation
uniquePromotionNumber	Unique discount number

	amount	Amount of discount in percents
	discountSsr	SSR with discount information For example, "DISCOUNT/DS03/1-1PSJTS5/", where DS03 – special discount percent format; 1-1PSJTS5 – ID of CRM discount campaign
	discountCardSsr	SSR with information about "Wings" card, for which discount has been added For 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 Sirens response. Array can be empty, it is recommended to analyze errors by ibeCode
	code	Error code
	text	Error text

Requests and responses examples

Successful pricing request for OW flight:

```
{
  "outboundFareFamily": {
    "category": "ECOFLEX",
    "code": "ECOFLEXRU",
    "isSubsidized": false
  },
  "returnFareFamily": null,
  "fares": [{
    "fareBasis": "WECOW",
    "isBreakpoint": true,
    "passengerType": 0,
    "flightDirection": 0,
    "flightIndex": 0
  }
],
  "currency": "RUB"
}
```

Response:

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

```

        "referenceNumber": "0",
        "referenceQualifier": "TST",
        "baseAmount": 15875.00,
        "totalAmount": 19357.00,
        "currency": "RUB",
        "passengerReferences": [
            {
                "referenceNumber": "12",
                "referenceQualifier": "PA"
            }
        ],
        "passengerType": 0
    }
],
"errors": [],
"success": true
}

```

7_Service API methods description

7_1_api/system/availability: service for monitoring the status of IBE

API service	api/system/availability
Short description	Allows to monitor the status of IBE and availability of integrated services
Supported HTTP(S)-methods	GET

GET system/availability method

Description: method allows to request the status of services on the backup IBE contour.

Request format /api/system/availability

Request parameters none

Response format

```

{
  "data": [
    {
      "service": "string",
      "severityLevel": integer,
      "isAvailable": boolean,
      "isEnabled": boolean
    },
    {
      "service": "string",
      "severityLevel": integer,
      "isAvailable": boolean,
      "isEnabled": boolean
    }
  ],
  "isAvailable": boolean
}

```

Response parameters

Parameter	Description
data	Array of objects with information about availability of services
service	Service name
severityLevel	Level of service severity Low = 1 Medium = 2 High = 3
isAvailable	Sign of service availability If connection is successful the value will be true
isEnabled	Sign of activity. If the service is disabled the value will be false
isAvailable	Sign of overall performance of IBE Will be <i>true</i> only when all services with severityLevel=3 are available

8 Attachments

8_1 Attachment: Dictionaries

Passenger types (other passenger types are under development)

paxType field value	Description	Passenger type
0	ADT, adult	Conventional
1	SRC, pensioner	Subsidized
2	YTH, youth	Subsidized
3	CHD, child	Conventional
4	INF, infant	Conventional
5	CNN, child	Subsidized/conventional
6	UMNR, child	Unaccompanied child
7	STR, adult	Subsidized for Far East Region, Kaliningrad's subsidy
8	JNN, child	Subsidized for Far East Region
9	MIL, adult	A passenger with military travel document of Military Ministry of Defence of the Russian Federation
10*	DIS, adult	Subsidized passenger with II or III disability stage
13	INS, infant	Infant with a seat
17	MMI, adult	Passenger of Ministry of Defence
18	MIR, adult	A passenger with military travel document of Ministry of Internal Affairs of the Russian Federation
19	MNN, child	Military man's (MIL) child
20	MNS, infant	Military man's (MIL) infant with a seat
21	MML, adult	Passenger flying with the military man (MIL) with cash payment
22*	DCD, child	Subsidized 2-11 years old child with disabilities
23*	DIC, child	Subsidized 12-17 years old child with disabilities

24*	DAC, adult	Accompanier for passenger with 'T' disability stage - SBK; for child with disabilities - DCD or DIC
25*	SBK, adult	Subsidized passenger with 'T' disability stage
26	CHR, child	Child 2-11 years old is official living at the Kaliningrad city or Kaliningrad region. Kaliningrad's subsidy
27	STU, student	Student 14+ years old is learning at the full-time form or military college at the Kaliningrad city or Kaliningrad region. Kaliningrad's subsidy

*** this type is not used when creating a reservation, it can be returned in a GET/reservation response**

Passenger sex

sex field value	Description
0	Male
1	Female

Document types

documentType field value	Description
0	Passport (general civil)
1	International passport
2	Birth certificate
3	Other type of document

Passenger assistance types

assistanceType field value	Description
0	Without assistance
1	Wheelchair is needed on the way to/from aircraft
2	Need assistance in getting on and off board
4	Both assistanceType values 1 and 2

Wheelchair types

wheelchairType field value	Description
0	Manual type
1	Electric type with dry battery
2	Electric type with liquid elements battery

Auxiliary services types

services.code field value	Category	Description
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)
EQPT	luggage	Sport equipment transportation option. Available at the "Promo" fare
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
ISMS	other	SMS notification service
UPQE	other	Service for upgrading the class of service.
TIME	other	Deferred payment service
AUCH	other	Automatic check-in service
CHNM	data	Change name service
URFL	insurance	Insurance, avia bundle
URTR	insurance	Insurance, travel bundle
PETC	pets	Pet transportation
CHML	meal	Child meal
HTML	meal	Hot meal
MOML	meal	Muslim halal meal

SNML	meal	Sandwich meal
SPML	meal	Special meal
CAML	meal	Dessert
KSML	meal	Kosher
BGML	meal	Burger
AVML	meal	Vegetarian Hindu
HDML	meal	Hindu meal
JDML	meal	Jain vegetarian meal
HNML	meal	Indian Cuisine
VDML	meal	Vegetarian meal
VGML	meal	Vegetarian meal
VJML	meal	Vegetarian jain
TCML	meal	Hot breakfast
GFML	meal	Gluten free
DDML	meal	Baby meal
BBML	meal	Baby meal
CPML	meal	Cheese platter
RPML	meal	Fish lunch
MPML	meal	Meat platter
GRML	meal	Greek salad
MTML	meal	Meat steak meal
UDML	meal	Seafood udon meal
SFML	meal	Seafood 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
ODML	meal	Asian Cuisine
ORML	meal	Asian Cuisine
MEML	meal	Italian Cuisine
GEML	meal	German Cuisine
ASML	meal	Central Asian Cuisine
CDML	meal	Business set
NYML	meal	New Year meal
ROML	meal	HAM and CHEESE ROLL
CTML	meal	Coffee and dessert
CCML	meal	Cappuccino coffee meal
CFML	meal	Americano coffee meal
SEAT	seat	Seat selection
BLOU	room	Business room
VLOU	room	VIP room
CVID	covidExpress	Rapid analysis for covid-19

Types of additional services supported by the engine (not sold in the engine)

Field serviceCategories.code	Category	Description
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 bulky carry-on baggage
XBAG	luggage	Oversized baggage
HBGS	luggage	Excess baggage up to 50 kg
HBGT	luggage	Excess baggage 32-50 kg
DBGS	luggage	Increase in baggage weight up to 32 kg
DBGT	luggage	Increase in baggage weight from 23 to 32 kg

Field serviceCategories.code	Category	Description
WBAG	luggage	registration of payment for excess weight of checked baggage from 10 to 23 kg
FRAG	luggage	Fragile luggage
SRXBLL	luggage	Payment for excess weight of hand luggage from 5 to 10 kg, not exceeding dimensions 55*40*20
SRXBLS	luggage	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
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
AVIH	pets	Animal in the hold
SVAN	pets	Animal in the cabin
EXST	seat	Additional free space nearby
STRQ	seat	Seat request

Field serviceCategories.code	Category	Description
GPST	seat	Group seat request, specify number and preferred location
RQST	seat	Seat request-include seat number or preference
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
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
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
URFL	insurance	Flight insurance, online
URTR	insurance	Insurance, travel package
URFO	insurance	Insurance, avia package
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
SPEQ	luggage	Sport equipment

Field serviceCategories.code	Category	Description
PTWP	luggage	Weapon and ammunition transportation at the luggage option, paid
FTWP	luggage	Weapon and ammunition transportation at the luggage option, for free

Types of animals

services.text field value	description
DOG	Dog
CAT	Cat
PARROT	Parrot
CANARY	Canary

Types of Sport equipment for transportation

services.text field value	description
BIKE	Bicycle
SKIS	Ski/Snowboard
SURFING	Surfboard/Water-skiing
FISHING	Fishing equipment
GOLF	Golf equipment
HOCKEY	Hockey equipment
SCOOTER	Scooter

Most widely spread Seatmap seat characteristics (in development)

characteristics field value	description
1A	Seat not allowed for infant
1B	Seat not allowed for medical
1C	Seat not allowed for unaccompanied child
1D	Restricted recline seat

1W	Window seat without window
7A	In front of lavatory seat
7B	Behind lavatory seat
A	Aisle seat
AL	Seat adjacent to lavatory
B	Seat with bassinet facility
CC	Center section seat
CH	Chargeable seat
E	Exit row seat
EK	Economy comfort seat
FC	Front of cabin class/compartment
G	Seat at forward end of cabin
I	Seat suitable for adult with an infant
IE	Seat not suitable for child
J	Rear facing seat
K	Bulkhead seat
L	Leg space seat
LS	Left side of aircraft
9	Middle seat (MS in Sirena)
O	Preferential seat
P	Comfort seat
PC	Pet cabin
Q	Seat in a quiet zone
RS	Right side of aircraft
S	Smoking seat
T	Rear/Tail section of aircraft
V	Seat to be left or offered last
W	Window seat
X	No facility seat (indifferent seat)

Service statuses

Status	Description
HN	Service not confirmed
HD	Service confirmed, EMD is required
HK	Service confirmed, EMD is not required
HI	Service paid (registered)
UN	Service denied

Promocode statuses (in development)

promotionStatus field value	description
0	New
1	Used
2	Sent
3	Saved
4	Received
5	Hold

Main fare options

Option code	Category	description
bestSeats	availablePaid	first rows in cabin of economy class (up to 15th row) if available at the moment of registration
upTo3bonuses	bonus	Bonus accrual up to 3% (CIS flights, fares: Promo, Econom, Premium-econom, Comfort, Business)
upTo5bonuses	bonus	Bonus accrual up to 5% (RF and IAL flights, fares: Promo)
upTo6bonuses	bonus	Bonus accrual up to 6% (RF and IAL flights, fares: Econom)

upTo7bonuses	bonus	Bonus accrual up to 7% (RF and IAL flights, fares: Premium-econom, Business-light, Business)
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
twoSuitcase23kg	luggage	Two suitcases 23 kg
twoSuitcase32kg	luggage	Two suitcases 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

Category	Description
PR	«Promo» tariff
EC	«Econom» tariff
FL	«Premium-Econom» tariff
BS	«Business» tariff

CF	«Business-Lite» tariff
PL	«Promo-light» tariff
LC1	«Bagless» tariff
LC2	«Luggage» tariff
SOC	«Subsidized» tariff

States of seats on check-in

Status	Description
0	Seat reservation cancelled
1	Seat not selected
2	The seat is reserved free during the online registration
3	The seat is reserved for a fee during online registration
41	The passenger is registered for a free seat
42	The passenger is registered for a paid seat
43	The passenger is registered by an agent
5	The seat is prepaid before registration
6	The passenger is in the process of registering for a seat
7	The seat was pre-added by the agent
8	Error during registration

States of sessions for creating subsidized booking with approving

Status	Description
0	Unknown
1	Waiting an approve
2	Confirmed
3	Error
4	Canceled
5	Expired
6	Waiting a decide by CRM side for the request

Status	Description
7	Rejected
8	Waiting Passenger Approval
9	Waiting Files Approval

Statuses of checking subsidized passengers (for creating booking with subsidized fares and required approving)

Status	Description
0	Unknown
1	Waiting an approve
2	Confirmed
3	Error
4	Skipped (an approve is not required)

The subsidy types values (for creating booking for subsidized fares)

Numeric / String value	Description
0 / Disability	People with disabilities
1 / Kaliningrad	Kaliningrad
2 / FarEast	Far Eastern District
3 / Age	Age subsidies

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
OST	Южная Осетия	South Ossetia
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

8_2_Attachment: Main IBE error codes

Error code	Description returned by IBE	Error Description
IBE errors		
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
1004	Session pool cleaning operation error	Error appears when saving a non-existent session (with an obvious lifetime)

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	Error appears when attempting to obtain information from non-existent session
1008	Data validation error	Validation error, might appear on all booking steps. For instance when invalid passenger data is being sent
10080	Validation warning	Validation error, might appear on all booking steps. For instance when invalid passenger data is being sent
10081	The seat map is not available because the check-in for the flight has started	An error occurs when attempting to request a seat map for a flight for which online check-in has already begun
10082	The ticket for this flight is not issued	The ticket on this flight hasn't been issued
10083	No checkin data	Check-in data missing
10084	Reservation must contains TST to complete this operation	The reservation must contain a TST to complete this transaction.
10085	Invalid data in passenger's document information	Invalid information in passenger's document
1009	Text is required	Mandatory text field should be filled
1010	Promotion is not found	Promo-code hasn't been found
1011	Operation can not be performed due to invalid payment order status	The operation could not be completed due to an invalid payment status. The error occurs when attempting to modify a booking in the same session after receiving a payment link, so that there are no two simultaneous bookings
1012	Unable to cancel additional service with issued ticket	Unable to cancel an additional service with an issued ticket. The error occurs when trying to cancel services using the api/services service, if the service being cancelled 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	Authentication error. The error occurs when both the Authorization header and the X-API-Key are sent to the request at the same time, which is not possible.
1015	File type not allowed	File type not supported. The error occurs when attempting to pass files to the booking engine methods that do not match the file types.
1016	File size is too large	File size exceeds allowed. The error occurs when attempting to pass large files to the booking engine methods.
1017	Request body is too large	Request body size exceeds allowed. The error occurs when the request body is too large.
1018	Unable to change more then maximum allowed symbols	Cannot change more characters. An error occurs when attempting to change more characters in passenger information than allowed. Currently used to check when changing the last name and first name.
1019	Unable to change element twice	The item has already been changed, the operation cannot be repeated. The error occurs when a repeated attempt is made to change passenger data. Currently used to check when changing the last name and first name
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.
10221	Redis connection error	Error while connection to Redis
10222	Redis system timeout	Redis response time-out
1031	Refund sessions store has failed to process an operation	Error of services refund. Occurs in case of unsuccessful attempt of read/store of refund's information in database
1032	Refund process is locked by another process	Refund process has been blocked by another process. Occurs when attempting to return one PNR at a time

1033	Refund is incomplete	Refund failed. The error occurs if one of the PNR refund processes is not in the final status. See the section describing the api/refund service.
10331	Database connection error	Error while connecting to database
1040	The ticket refund info was not found	No ticket refund data found. The error occurs when the status of tickets in the IBE database and in the PNR on the Amadeus side differs
1041	Some passengers are not exist in reservation	Some passengers are missing from the booking. The error occurs if the passengers provided in the request were not found in the booking
1042	The remaining passengers are of the wrong type	Incorrect type of remaining passengers to refund. This error occurs when the remaining passengers in the booking are of the wrong type. For example, when trying to cancel an adult passenger in a booking with 1 adult and 1 child
1043	The PNR was created not in IBE	PNR not created in IBE. The error occurs if the booking was not created in the booking engine (without using the IBE API)
1044	One of the specified refund items is not found or not available for the confirmation	One of the specified refund points was not found or is not available for confirmation. The error occurs if it is impossible to confirm the refund for one of the tickets (GOI-proxy service, PUT refund/pnr)
1045	Refund is not available for the reservation	Refund is unavailable for current reservation
1050	The partner refund process returned error	Error of partner's service refund. Occurs while unsuccessful attempt of partner's services refund
1051	The services refund process returned error	Error of air company services refund. Occurs while unsuccessful attempt of air company services refund
1052	The splitting reservation for refund process returned error	Error when returning split reservation
1053	There are services with bonus payment, refund process cannot be completed	The refund process cannot be completed because some services were paid for with bonuses.
1054	The passenger tickets refund process returned error	Error returning tickets for this passenger
1055	Refund process is incomplete, failed to send confirmation to payment service	Refund not completed due to failure to send confirmation to BPC

1060	The refund session has a problem	Problem with refund session. The error occurs when the booking refund could not be completed to the final status
1061	Unable to find a refund record based on passed passenger data	No refund record found based on the transferred passenger data
1062	Refund validation error	Error checking return
1063	IbeRefundCrmRequest	Error while returning. Error requesting to CRM
1079	Passenger not exist in reservation	Attempt of obtaining the information of passenger non-existed in the booking
1080	OrderInfo client returned error	Error retrieving order information. Error occurs when GOI on U6 side returns an error
1081	OrderInfo client returned unauthorized error	Authorization error when retrieving order information. The error occurs if the GOI service returned a 401 error (not authorized)
1082	Passenger not found in reservation	Attempt of obtaining the information of passenger not founded in the booking
1083	You cannot compose services with empty list of TSM	TSM list cannot be empty
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	Error occurs if selected reservation cannot be annulled
1087	Segment can't be cancelled	Attempt of delete a segment which cannot be cancelled
10871	Adding loyalty card is not possible for inactive segments	Adding of loyalty card is not possible for inactive segments
1088	Ibe only flights with open date	IBE error: only open-date flights available
1091	One of passengers' last name has incorrect format	One of passenger's surname has incorrect format
1092	Passenger documents in incorrect format	Invalid passenger document format. An error is returned if the passenger data is not in a valid format

1093	Passenger gender is required field	Passenger gender not selected. The error occurs when the passenger gender is not passed in the request
1094	Passenger birthdate is required field	Passenger's date of birth is not specified. The error occurs when the passenger's date of birth is not passed in the request
1095	Passenger firstname and surname in incorrect format	Incorrect format of passenger's first and last name. The error occurs if the passenger's last name is presented in an invalid format.
1096	Reservation prolongation is unavailable	Invalid API key for extending the booking session. The error occurs if adding a MIS segment is not available.
10961	Incorrect number of passengers	Incorrect number of passengers
10962	Incorrect booking travel locations	Incorrect booking of travel seats
1097	Passenger with infant is child	The passenger accompanying the child must be over 18 years old. The error occurs if there are no adult passengers in the request to create a booking with an infant
1098	Airport code not found	The error appears if the airport code not found.
1099	Failed to update or retrieve refund job data	Unable to update or retrieve refund information
11000	The operation was canceled	Operation canceled
1500	Failed to receive exchange rates.	Unable to receive information for api/exchangerates. Method uses API CB RF for receiving of data
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.
1988	UA flight availability calendar service error	Error during attempt to obtain flight schedule service
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
9004	Unknown IBE error. Null Ok response.	The error occurs if the response to the request to create a booking is returned empty.
OrderInfo errors		
20000	Incorrect login or password.	Occurs when invalid login or password is passed to request.
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 OrderInfo proxy service did not find the requested reservation or was unable to load it.
4001	Catalogue of services is unavailable	Service catalogue is unavailable
4002	OrderInfo proxy service has failed to prepare partners products for issue operation	OrderInfo proxy service cannot release partner products
4002	Invalid service code	Invalid service code
4003	OrderInfo proxy service has failed to issue partners products	Proxy server error when releasing a partner product
4003	Pack already contains the added service	The service being added is already present
4004	OrderInfo proxy service has failed to decrease fop amount on insurance amount	OrderInfo proxy service cannot reduce the cost by the insurance amount
4004	Cannot select more than one package for a passenger	It is not possible to select more than one package per passenger
4005	Pack without included service is not allowed	Package without added services is not allowed
4005	Issue service has failed to issue services	Error releasing service for already released service
40051	Issue service has failed to issue free services	Error when issuing free services
40052	It is not possible to issue free services or tickets when the reservation or other services are not paid	It is not possible to issue free services or tickets if the reservation or other services have not been paid for.
40053	Issue service has failed to issue. Invalid Total Amount	Service release not possible. Incorrect total cost
40054	Issue service has failed to issue free tickets	Error issuing free tickets
4006	Can't issue service or partner without paid tickets	It is not possible to release a service or partner without paid tickets

4007	OrderInfo proxy service has failed to change passenger data	Error while proxy server trying to change passenger data
4008	Issue proxy service has failed to issue tickets	Proxy service error for already issued tickets
4009	Issue service has failed to change name	Error when changing name
4010	Passenger ticket not found	Ticket for this passenger hasn't been found
4011	Issue service has failed to enqueue autocheckin	The issuing service was unable to queue automatic registration
4100	OrderInfo service has failed to process request	The error occurs if the GOI-proxy was unable to fulfill the request
4101	Invalid request	Incorrect request
Exchange errors		
11001	Unable to process with exchange. Need services refund before ticket exchange.	It is not possible to complete the exchange of tickets without prior refund of additional services.
11002	The ticket exchange info was not found	No information found about ticket exchange
11003	Unable to process with exchange. Cannot split linked passengers.	Unable to complete exchanges. Unable to split related passengers.
11004	Unknown segment reference	Unknown flight segment reference (segmentReference parameter)
11005	Exchange not started	The exchange of this ticket hasn't been initialized
11006	Payment needed	Payment for exchange required
11007	Error while confirming exchange	Error confirming exchange
11008	Error is occurred while processing exchange variants	Error processing exchange options
11009	Exchange is forbidden for this reservation	Exchange is not allowed for this reservation.
11010	Exchange	Unknown error while exchanging
11011	Segment reference and the last segment reference are in the wrong order	Segment reference and last segment reference refer to incorrect order
Errors while ticket refund		
1133	No tickets to refund	No tickets available for refund
1134	Voluntary refunds are not possible because one of the passengers checked in for the flight	Voluntary refund is not possible as one of the passengers is already registered for the flight. It is necessary to cancel the registration in advance

1135	Annulation are not possible because one of the passengers checked in for the flight	The flight cannot be cancelled because one of the passengers is already registered for the flight. It is necessary to cancel the registration in advance
Certificate errors		
11101	The certificate balance is zero	The certificate's balance is empty
11102	The certificate is expired	Certificate has been expired
11103	The certificate status is not open	Attempt to make an operation with the certificate with wrong status
11104	The certificate surname mismatch error	Passenger's surname doesn't match with certificate
11105	The reservation or services cannot be paid by the certificate	Selected reservation or services cannot be paid by the certificate
11106	The certificate currency is invalid	Incorrect certificate currency
11107	The certificate surname mismatches with the reservation surname	Certificate's surname mismatches with the reservation one
11108	The certificate amount is greater than needed to pay for the reservation and services	Certificate amount is greater than needed to pay
11109	The certificate does not contain information about the name of the done	There is no information about naming
11110	Error while applying discount	Error while applying the certificate
11111	Error while applying discount. Certificate not found	Error while applying the certificate: certificate doesn't found
11112	Error while issuing certificate for residual amount	Error while certificate's issuing
11113	Operation available only for covid certificates	Current operation available for COVID-certificates only
11114	The certificate refund is failed	Error while refund of certificate
Errors while getting the boarding pass		
12000	Send boarding pass is failed	Error while sending the boarding pass
12001	Wrong passenger	Incorrect passenger
12002	The boarding pass is not available for the passenger	The boarding pass is not available for current passenger
12003	The boarding pass temporary is not available	The boarding pass is temporary unavailable

12020	An adult passenger count should be equal or more than an infant count	The number of adult passengers must be greater than or equal to the number of infants.
12021	The reservation should contain at least one passenger over than 18 years old	The booking must include at least one passenger over 18 years of age.
12022	Accompaniment booking not found	No accompanying person found in the booking
12023	Segments in accompaniment booking do not match with requested booking segments	The segments in the booking support do not match the requested booking segments
12050	The airport info is not found in the database	No airport information found in the database
12070	Operation is not available for segment with open date	Operation not available for open date segment
12080	Information services should be added for all segments in the same direction	Information services should be added for all segments in the same direction
12090	Failed to add airShopSessionId into booking	Unable to add airShopSessionID to booking
13000	Failed to send upgrade completed email. Email not found	Unable to send email: email address not found
20000	Incorrect login or password	Occurs when an incorrect login or password is provided to obtain a GOI token.
20001	Channel creation timeout is exceeded	An attempt to establish a connection to SIG took too long and resulted in a timeout error.
20100	Unhandled Fare Family	Unsupported tariff family
CRM-side errors		
1200	External API Unknown error	Unknown error on the CRM side
1201	External API Connectivity error	Error connecting to CRM
1202	External API Connection timeout exceeded	Connection timeout to CRM exceeded
1203	Promocode error	Promo code error on the CRM side
1204	Personal information agreement confirmation error	Error confirming personal data on the CRM side
1205	Promocode error	Error checking promo campaign on the CRM side
12051	Promo code for services is temporarily unavailable	Promo code for services is temporarily unavailable
12052	Promotion has not started yet	The promotional campaign has not started yet
1206	Calculate discount error	Error in calculating discounts
1207	External API error	CRM API Error

1208	Calculate discount list error	Error calculating discount list
1209	On this phone already created involuntary refund	A request for a forced refund has already been created for this phone number.
1210	Unknown involuntary refund error	Unknown error during involuntary refund
1211	Involuntary refund not found	No data found on involuntary refund
1212	External Core API returned error	CRM API Core Error
1213	External API session token expired	External CRM API Key has expired
1214	External API session token expired	CRM API Key Expired
1215	Food on board service has wrong result	Incorrect data for the "On-board Meals" service
1216	CRM FTP returned error	FTP error on CRM side
1217	Not enough permissions for this request	Insufficient rights for this request
1218	Wrong profile data provided	Error - loyalty card data mismatch
1219	Reached maximum number of failed requests attempts for the session	Maximum number of failed request attempts within a session reached
1220	External Core API is too busy and doesn't respond	External Main API is too busy and not responding
Errors that occur during payment		
1300	Unknown Payment gate error	Unknown payment gateway error
1301	Payment gate hash value is incorrect	Invalid payment gateway hash value
1302	Unknown Payment History error	Unknown payment history error
1303	Payment History Login returned error	Error authorizing for payment history
1304	Payment History By Pnr returned error	Error requesting payment history by PNR number
1305	Payment service returned error	Payment service error
1306	Url for perform payment method is empty	Payment link missing
1307	OrderId for perform payment method is empty	There is no order number for payment
1308	Payment service login error	No authorization in the payment service
1309	Missing payment setting	Payment type not found
1310	Services cannot be paid for	Services cannot be paid for
1320	Payment email report upload failed	Error sending payment information to e-mail

Errors that occur when issuing a check		
1400	Receipt service returned error	Error while attempt of receiving a check
1401	Receipt send email service returned error	Error sending check to e-mail
1402	Receipt timeout exceeded	The waiting time for receiving the check has expired
1403	Receipt error. There are no issued tickets in the booking	Receipt Receive Error. There are no tickets issued in the booking.
1404	Receipt send error. There are no contacts in the booking	Receipt Error. No contacts in the booking
1405	Receipt send not available booking	Sending a receipt for a non-existent reservation
1406	Receipt error. Some tickets refer to inactive flight segments	Error in receipt. Some tickets are for inactive flight segments
Errors on TAIS side		
1500	Failed to receive exchange rates	Error of receiving exchange rates. It is impossible to get information by api/exchangerates method. Method uses API of Central bank of Russia for getting information
1600	Mail service error	Email sending service error
1700	Tais API returned error	API error on TAIS side
1701	Tais connection timeout	TAIS connectivity time-out
1702	Tais connectivity error	TAIS connectivity error
1703	Tais Ftp server return error	FTP-error on TAIS side
1710	Sofi report upload failed	Error loading Sofi report
SIG side errors		
1750	Unknown SIG error	Unknown error on SIG side. The request specifies a date in the past (in terms of the departure airport time zone)
1751	SIG connection timeout exceeded	SIG connectivity time-out
1752	SIG connectivity error	The number of services does not match the number of passengers in the order, since there is a non-removed service
1753	SIG AirBookRQ function returned error	AirBookRQ request error on SIG side
1754	SIG AirTicketRQ function returned error	AirTicketRQ request error on SIG side

1755	Bad Request. Check parameters of request	Incorrect request. It is necessary to check request parameters
1756	SIG returned an incorrect PaxType for passenger	Incorrect passenger's type (PaxType parameter)
1800	Unknown social network error	Unknown error while attempting of connection using social open network
1801	Social network connectivity error	Connection error while attempting of connection using social open network
1802	Social network connection timeout exceeded	Time-out of connection using social open network
1803	External not linked	External connection error
1810	External system temporary unavailable	External system is temporary unavailable. The error occurs when SIG's request "breaker" is triggered.
1811	Total request process did not complete within the timeout	External system total Timeout. An error occurs when the total number of requests processed has not completed within the allotted timeout
Alfa-insurance errors		
1850	Unknown Alfa insurance service error	Unknown error on Alfa-insurance side. Response time-out
1851	Alfa insurance service PoliciesQuote method returned an error	PoliciesQuote request error
1852	Alfa insurance service PoliciesCreate method returned an error	PoliciesCreate request error
1853	Alfa insurance service PoliciesConfirm method returned an error	PoliciesConfirm request error
1854	Alfa insurance service SaleWithoutInsurance method returned an error	SaleWithoutInsurance request error
Aeroexpress errors		
1900	Aeroexpress services return error	Unknown Aeroexpress error
1901	Aeroexpress connection timeout exceeded	Connectivity time-out
1902	Aeroexpress connectivity error	Connectivity error
1903	There is no available aeroexpress tariff for the request	Additional fares are unavailable
SeaLine errors		

1930	Unknown SeaLine error	Unknown error
1931	There is no any product in SeaLine	No services available
1932	SeaLine calculation method returned error	Calculation error
1933	SeaLine create order method returned error	Error while creating an order
1934	SeaLine confirm order method returned error	Order confirmation's error
Partner's services errors		
1940	Sberhealth services return error	SberHealth side error
1950	Unknown Partner	Unknown partner
1951	Empty partner products list	There are no services of current partner
1952	Partner product wasn't issued	Required partners service hasn't been issued
1953	Unknown partner product booking ID	Unknown partners service ID
1954	Error while trying issue partner product	Error while adding selected partner's service
1955	Error while trying reissue partner product	Repeated error when trying to add partner's service
1956	Error while trying reissue partner product. Maximum attempts amount reached	Repeated error when trying to add partner's service. Number of attempts exceeded
1957	A partner type is not available	Unavailable partners service was selected
1958	Partner products reissue not possible, PNR not found	Repeated error while adding affiliate services: PNR not found
1959	One of partner products is not available	One of selected partner's products is unavailable
1960	Partner product is not currently available on additional purchase	Selected partner's service is unavailable for adding
1961	Error in VAT calculation	Error while VAT calculation
1962	Partner service returned error	Error while obtain partners service information
1963	Incorrect flight or passenger reference in the request	Incorrect passenger or flight reference
1964	Flight tickets was not issued	Repeated error: ticket wasn't issued
1965	Partner product was not paid	Partners product has no fee
1966	Payment and partner product amount mismatch	Repeated error with payment amount mismatch
1967	Partner product not found in reservation	Repeated error - service not found in reservation

1968	Statuses in DB and reservation mismatch	Statuses in the database and in the booking do not match
MaoCloud errors		
1970	Maocloud unknown error	Unknown MaoCloud error
1971	Maocloud API general error	General error on MaoCloud API's side
1972	Maocloud API unauthorized	Client not authorized to MaoCloud API
1973	Maocloud API wrong params	Invalid parameters of request to MaoCloud API
1974	Maocloud refund error	Error on refund on MaoCloud side
Change name errors		
1975	No passenger details have been changed	There are no changes in the passenger data
1976	The service of changing passenger data was added by another office	The passenger data change service was performed by another agent
1977	Missing Text property in the passenger data change service's info	Data lost when changing name
1978	No more services can be added to reservation while an unpaid CHNM service is present	Unable to add services until name change payment is made
Yandex.OCR errors		
1980	Unknown Yandex OCR server error	Image processing error: unknown API error
1981	Passenger's passport wasn't found	Image processing error: passport data not found
1982	Request limit to Yandex OCR API exceeded	Image processing error: attempts limit exceeded
1983	Invalid request parameters	Image processing error: invalid request parameters
1984	Yandex OCR server internal error	Image processing error: internal server error
1985	Couldn't find operation of recognition by Id	Image processing error: Unable to recognize ID of document recognition operation
1986	Incorrect or expired Api-key	Image processing error: API key is invalid or expired
1987	Internal server error on recognition documents	The documents did not pass verification
Seatmap errors		
2000	Unknown IBE Connector error	Unknown IBE connector error. Unhandled exception, cause unknown

2001	Seat is unavailable for this passenger type	The seat is not available for this type of passenger
2002	Seat is unavailable	Chosen seat is unavailable
2003	Seat is not compatible with pack	Current seat is not compatible with the service package
2004	Pack that is compatible with seat contains error	Seat incompatibility error
2005	PetInExitRow	Pet cannot be placed in the emergency exit row
2006	Can't book several passengers for one seat	It is not possible to add multiple passengers to one seat
2007	Cannot choose pet and comfortable seat at the same time	It is not possible to add a pet and comfort seat at the same time
2008	Invalid passenger reference	Invalid passenger reference
2009	Seat up to 16 row cannot be provided for passengers with pets	Seats up to row 16 are not available for passengers with pets
2010	The seat added at the stage of check-in cannot be changed	A seat that has already been checked in cannot be changed
2011	There is no service or seat with this reference in the reservation	No service or seat with this link found in the booking
2012	Unable to delete issued services	It is impossible to delete already paid services
2013	It is not possible to delete the Unaccompanied Child service	Unable to remove the "Unaccompanied Child" service
2014	Unable to delete a seat added during the online registration process	It is not possible to delete a seat added during the online registration process
2015	The passenger seat has already been selected. To change a passenger seat, first delete the previous seat	The passenger seat is already selected. To change the passenger seat, first delete the previous seat
2016	The passenger seat was already issued	Current seat has already been chosen
2017	The child's seat should be close to an adult	The child's seat should be next to the adult
2018	Unable to check seat due to connectivity error. Try later.	Error checking availability of seat (connection error). Try again later
2019	The passenger seat is unavailable because a related passenger seat unavailable	Current seat is unavailable because the seat associated with it is unavailable
2100	Unknown DMS error	Unknown error on DMS side
2101	DMS connection timeout exceeded	DMS connectivity error

Errors while retrieving airport data		
2200	Airports service error	Unable to get information from api/airports method. This method uses u6.ru API to get data
2201	Airports cache is empty	Api/airports method did not return data
2202	Airports API returned invalid result	Api/airports method returns invalid data
2203	Airports subsidize API returned invalid result	Airport's API error
Mindbox errors		
2300	Mindbox API returned error	General error of MindBox API
2301	Mindbox API returned authorization error	Error of authorization on MindBox API
2302	Mindbox API returned validation error	Error of request validation of MindBox API
2303	Mindbox API is is temporarily unavailable	Error appears when MindBox server is unavailable
2304	Mindbox API return bad request with message \"Operation ... not found\"	Bad request: operation cannot be found
Sirena errors		
3000	Unknown Soap receiver service error	Unknown SOAP receiver error
30001	General integration error	General error on Sirena side
30002	Connectivity error	Error while connecting with Sirena server
30003	PNR not found. No match for record locator	Requested PNR wasn't found
30004	Office configuration error	Error of office configuration
30005	PNR retrieve function returned error	The user entered an incorrect ticket number when requesting a refund
30006	Discount provider error	Error while calculating discount on Sirena side
30007	Pnr is blocked	PNR is blocked by another office
30008	PNR was changed	Requested PNR has been changed
30009	General integration error	General error while integration with Sirena server
30010	Temporarily unavailable. Try later	Sirena server is temporarily unavailable
30023	Connectivity error. Invalid private key	Connectivity error because of invalid authorization key
300021	Connectivity error. Try later	General Sirena connectivity error. Try later
300022	Connectivity error. Invalid user id	Connectivity error because of invalid user ID

300024	Connectivity error. Public key not received	Connectivity error while receiving a public key
300025	Connectivity error. Try later	Connectivity error
300026	Connectivity error. Try later	Connectivity error because of response time-out
300027	Connectivity error. Try later	Connectivity error
300028	Connectivity error. Try later	Connectivity error
300029	Connectivity error after successful payment. Please try to get payment information later	Connectivity error after successful payment. Try to get information about payment later
300030	Connectivity error. Try later	Connectivity error
300031	Connectivity error. Try later	Connectivity error
300032	Answer decompressing error. Please try later	Error while forming response
300051	Display ticket function returned error	Error while ticket display
300052	Retrieve remote pnr function returned error	Error while accessing PNR remotely
300053	Claim pnr function returned error	Error while obtaining PNR number
300054	Display ticket is not found	Ticket didn't found for displaying
3000201	Connectivity error. Try later	Connectivity error
3000202	Error on clear cache after symmetric key exception	Error while clearing a cache
3000203	Connectivity error. Try later	Connectivity error
3001	Unknown insurance code	Unknown insurance code
3002	Catalogue of insurance services is unavailable	Error of insurance services availability
301001	Error on pricing flights. Empty flights brands	Error on pricing flights
302001	Error on cancel the booking	Error on booking cancellation
302002	Error on cancel the booking. The booking has already been issued	The booking has already been issued
31004	An error occurred while requesting catalogue	Error on requesting catalogue
31005	No services returned from a provider	Error requesting services from provider
31006	Can't add seat	Error while seat adding
31007	Seat is not available	Current seat is unavailable
31008	An error occurred while requesting catalogue	Error on requesting catalogue

31009	Change passenger data is unavailable	Passenger's data changing is unavailable
31100	Error adding additional services to reservation	Error adding additional services to reservation
31101	Error when deleting a service from reservation	Error when deleting a service from reservation
31102	Can not remove issued service	Can not remove issued service
31103	Service not found during remove	Removing service didn't found
311000	Error when issuing a ticket	Ticket issuing error
311001	Error when releasing services	Services releasing error
311002	Error when releasing partner services	Partner's services releasing error
311003	Amounts payable validation error	Error checking payment amount
311004	Error when changing passenger data	Error while changing passenger data
311005	Error when removing remark from reservation	Error when deleting a remark from a booking
311006	Error when removing ssr from reservation	Error removing SSR from booking
311007	Error when adding ssr to reservation	Error adding SSR to booking
311008	Error when adding ssr to reservation	Error adding SSR to booking
311009	Add contact failed	Error adding contact details
311011	An error occurred while adding passengers to the booking	Error adding passengers to booking
3110001	Error when canceling ticket issue	Error canceling ticket issue
3110010	Reached maximum number of failed requests attempts for the session	The session has reached its limit of failed requests
3110011	Error when cancel services issue	Error canceling issued services
3110041	Error when canceling name changing	Error when undoing name change
3110071	Ssr already added	SSR already added
31210	Not enough quota for tickets	Not enough ticket quotas
31220	Not enough quota for emds	Not enough emds quotas
31221	An error occurred while issuing services	Services issuing error
31300	Final status can only be changed to Open	Final status can only be changed to "open"
31301	Coupon status change failed	Error when changing coupon status

32003	Infant must be with adult	Infant must be accompanied by an adult
32004	Ticket not found	Ticket not found
32100	Error while voiding tickets	Ticket voiding error
33000	General refund error	General error while refund
33001	Services not found	Services not found
33002	Pnr is blocked	PNR is unavailable
33003	Refund not priced	Refund not priced
33004	Refund amount not valid	Incorrect refund amount
33005	Penalty greater than fare on returned ticket	Fine exceeds refund amount
33006	Error occurred while returning services	Error occurred while returning services
33007	Cannot refund tickets paid with more than one documents	It is impossible to return tickets paid for with more than one document
33008	Error is occurred while refunding reservation with the invalid segment state	Error returning booking with invalid segment state
33009	The operation must be done manually	Operation must be performed manually
33303	Unable to find specified ticket	Unable to find a ticket with the given specifications
34001	Version mismatch	Version mismatch
35000	Booking request returned error	Error while requesting reservation
35001	Impossible to book reservation with infant	Unable to book reservation for infant
35002	Cannot reserve seats	Error while attempt of reserving seats
35003	Booking is forbidden	Sirena has limited the remote control's functionality due to too frequent requests.
35004	Cannot book and price PNR	The selected fare combinations have ended, you need to search again. Passengers are shown a pop-up.
35005	An error occured while creating land segment booking	Error creating segment reservation
35006	An error occured while booking. Try select another date or flight number	Error booking. Try another date or flight number
35010	Certificate booking request returned error	Certificate error when booking
35011	Failed creating ssr request for certificate	Error while creating SSR for selected certificate

35012	Certificate ssr request returned error	Error while requesting SSR for selected certificate
35013	Failed creating manual pricing request for certificate	Error while creating a request for manual certificate pricing
35014	Certificate manual pricing request returned error	Error while execution of request for manual certificate pricing
35015	Certificate manual price mismatch error	Manually calculated price mismatching error
35016	Certificate not found	Requested certificate wasn't found
35017	Certificate service section not found	It is impossible to found a section of selected service
35018	Certificate service not ready for issue	Selected service is not available for issue
35019	Issue certificate service failed	Error while issuing a certificate
35020	Certificate request by number returned error	Error while requesting certificate by number
35022	Certificate service number mismatch error	Error appears if certificate number mismatches
35023	Certificate company name mismatch error	Company name mismatch error in certificate
35024	Certificate request didn't returned price	No price in certificate
35025	Certificate with specified number not found	No certificate found with this number
35026	An error occurred while restoring the certificate	Error restoring certificate
35027	An error occurred while restoring the certificate for refund	Error restoring certificate for refund
35028	An error occurred while refunding the certificate	Error refunding certificate
35029	Certificate with specified number not found	No certificate found with this number
35030	Certificate validity date is not found	Certificate expiration date not found
35031	An error occurred while refunding the certificate, please try again later	General error of certificate refund. Try later
35032	An error occurred while sending the certificate. No contacts found	Error sending certificate. Contact details not found
35040	An error occurred while requesting info about reservation	Error while requesting reservation data
35041	An error occurred while unarchiving pnr	Error while unarchiving PNR from reservation
35100	Error while getting EMD info	Error retrieving EMD data

35101	EMD info response is empty	Empty response on EMD request received
36000	Queue provider error	Queue provider error
36001	Pnr access error	Error accessing PNR information
36002	Queue access error	Queue access error
37000	Wrong data format	Invalid data format
37100	An error occurred while searching exchange	Error while exchange searching
37101	There are no passengers with the requested references in the reservation	There are no passengers with the requested parameters in the booking
37102	There are no segments with the requested references in the reservation	The booking does not contain segments with the requested parameters
37103	An error occurred while processing booking segment information	Error retrieving flight segment booking data
37104	Exchange not allowed for this booking. Use refund instead	Exchange not available for this booking. Please use refund instead of exchange.
37105	Exchange not allowed for this booking. One of the passengers is checked in for the flight	Exchange not available for this booking. One of the passengers is already checked in
37106	Exchange not allowed for this booking. One of the segments is unavailable for exchange	Exchange not available for this booking. One of the flight segments cannot be replaced
37110	Apply exchange provider returned error	Exchange error while attempt to change ticket on a same one
37120	Error occurred while canceling exchange	Error when canceling exchange
37200	Searching in past returning empty flights	Error while attempt of searching the flights on past date
37250	The seat number is incorrect	Incorrect seat number (layout errors)
37270	Cannot annul reservation service	It is not possible to cancel the booking service
39999	Obsolete method. Please, read the documentation	Deprecated method. See Sirena's documentation
40100	Error occurred while requesting seat map	Error while requesting seat map
40101	Error occurred while requesting seat map price	Error while requesting seat prices
40102	No price for current passengers	Sirena's response does not include a price for some of the passengers in the booking.
40103	Flight not found	Flight not found

40104	No seat map for such segment	There is no seat map available for this segment.
40105	SeatMap aircraft has changed error	Aircraft seating arrangement has changed - error
40106	Calendar out of reach	The calendar is unavailable
40107	There are too many passengers in the request	Too many passengers in the request
40200	Incorrect response data format, please try again later	Invalid response data format, please try again later
402201	Incorrect or not expected Sirena type of response for get currency rates request	Incorrect or unexpected response type from Sirena to request for exchange rates
40220	Sirena currency rates request error	Error requesting exchange rate from Sirena
40230	Get ticket quota request failed, try later	Request for ticket quota failed, please try again later
4200	CallCenter proxy has failed to process request	Error while requesting call center proxy
4300	Trying to request a non-existent setting	Attempt to request a setting that does not exist
4400	Flight not found	Flight not found
Sirena SOAP service errors		
401000	An error occurred while requesting external service	Error while requesting external Sirena services
401001	Connection timeout is exceeded while requesting external service	Connection timed out while requesting external Sirena services
401002	Connectivity error occurred while requesting external service	Connection error while requesting external Sirena services
401003	Configuration error. Please contact feedback	Configuration error
401010	An error occurred while authorization external service	Error authorizing to external Sirena services
401011	An error occurred while requesting to checkin service	Error while requesting check-in services
401012	The requested flight segment was not found or is not active in the check-in system	The requested flight segment was not found or is not available for check-in.
401013	An error occurred while requesting to checkin service	Errors while requesting check-in services
401014		
401015		

401016		
401017	The requested passenger was not found in the check-in system	The requested passenger was not found in the check-in system
401018	An error occurred while requesting to checkin service	Errors while requesting check-in services
401019		
401020		
401021		
401022	No passengers found in the check-in system. There may be delays in execution, please try again later	Passengers not found in check-in system. There may be a delay in receiving data. Try again later.
401023	An error occurred while passenger reservation	Error during reservation
401024	Refused seat found in the check-in system. Web registration is not available	An unavailable seat was found in the registration system. Online registration is unavailable.
401025	No ticket found in the check-in system. Web registration is not available	No tickets found in the registration system. Online registration is not available.
401026	Refused seat found in the check-in system. Web registration is not available	An unavailable seat was found in the registration system. Online registration is unavailable.
401027	An error occurred while requesting to checkin service	Errors while requesting check-in services
401028		
401029		
401030	An error occurred while enrolling the associated infant	There was an error registering the corresponding infant
401031	An error occurred while setting payment status	An error occurred while setting the payment status

401032	Passenger not found in check-in service	Passenger not found in DCS. After creating a booking in Sirena, the data is not immediately pulled into DCS
401033	Web registration is not available	The error (SirenaSoapPassengerAgentTransitBortChanging) is passed to the registrationUnavailableCode parameter to hide the possibility of online registration and display the corresponding message to the passenger
Sirena check-in services errors		
401100	An error occurred while get checkin information	Error retrieving registration information
4011001	An error occurred while get seatmap information	Error retrieving seat map information
4011002	Online check-in is closed	Online registration is closed
401103	The passenger has services or restrictions that restrict access to online check-in	The passenger has services or restrictions that limit access to online check-in
401104	The passengers has services or restrictions that restrict access to online check-in	Passengers have services or restrictions that limit access to online check-in
401105	Online check-in is not available on codeshare flights	Online check-in is not available on codeshare flights.
401106	Can not cancel codeshare registration	Unable to cancel check-in for codeshare flight
401107	Segment isn't found	Segment not found
401123	Reservations are not available at the moment, please try again in a few minutes	Reservations are currently unavailable
401124	One of the segments of the transfer flight is not available for check-in	One of the connecting flight segments is unavailable for check-in
401125	Checkin reservation time renewal failed	Failed to update registration time
401126	Checkin reservation failed	Error while reserving registration
401127	Unable to replace preselected seat	It is not possible to change a pre-selected seat
401160	No requested passenger in reservation	The requested passenger was not found in the booking
401161	Online check-in for this flight is not available	Online check-in is not available for this flight.
401162	Ticket not issued	Ticket not issued

401163	Flight is expired	Flight time has expired
401166	Passenger seat not reserved	The passenger's seat is not reserved.
401167	The passenger seat has an invalid format	Incorrect passenger seat format
401168	The passenger is already registered	The passenger is already registered
401169	Passenger seat not purchased	No passenger seat purchased
401170	Missing one of the segments of the transfer flight	One of the connecting flight segments has been missed
401181	An error occurred while requesting to checkin service	Error while requesting registration service
401182	An error occurred while seatmap pricing	Error in calculating the cost of seats
401191	The passenger is not registered	The passenger is not registered or passenger's information didn't found
401192	An error occurred while requesting to checkin service	Error while requesting registration service
401193	The seatmap doesn't contain passenger's preselected/reserved seats, likely cause of the airplane reconfiguration	The seat map does not show the passenger's pre-selected/reserved seats, likely due to aircraft reconfiguration
401194	Cancellation of agency registration is prohibited	Cancellation of agency registration is prohibited
401195	Cancellation of paid registration is prohibited	Cancellation of paid registration is prohibited
401196	Web check-in is not available due to different luggage weight limits	Online check-in is not available due to various baggage weight restrictions
401197	Web registration is not possible on an unavailable seat	Web registration is not possible for an inaccessible location
401198	Requested flight canceled	The requested flight has been cancelled
Errors of Sirena services for getting seatmap on check-in		
401200	An error occurred while get seatmap information	Error while requesting seat map data
401201	There is no information about the flight in the online check-in system	There is no flight information in the registration system
401202	No cabin is assigned for the flight in the checkin-system	Associated cabin missing in flight check-in system
Errors of Sirena services for obtaining a boarding pass		

401400	An error occurred while get boarding pass information	Error retrieving boarding pass details
401401	Boarding pass is unavailable for passenger with pet	Boarding pass not available for passenger with pet
401402	Boarding pass is unavailable for subsidized segment	Boarding pass not available for subsidized flight segment
401403	No contacts for sending boarding pass	There are no contact details to send the boarding pass.
401404	Boarding pass unavailable	Boarding pass not available
40300	All autocheckin services is in failed state	All automatic registration services are not responding
Sirena NDC errors		
403000	Sirena Ndc Error	Common Sirena NDC error
403001	Sirena Ndc requests are forbidden for this office	Request is unavailable for current user (authorization error)
403002	Timeout for response from third-party service	Timeout during connection to 3rd party services
403003	Unable to connect to third-party service	Error during connection to 3rd party services
403004	Not available for NDC booking. Use claim	Request is unavailable without reservation claim
Errors of codeshare flights		
40400	Error while creating codeshare booking	Error creating a codeshare flight booking
40401	Email has no attachments	There are no attachments in the letter
40402	Wrong attachment type	Invalid attachment type
40403	Unknown carrier	Unknown carrier
40405	Missing error email setting	Incorrect email settings
40410	Error while parsing file	Error parsing file
40411	Error while parsing file header	Error parsing file header
40412	Missing required field	A required field is missing
40413	No records found in file	No records found in file
40420	Error while parsing record	Error parsing record
40421	Missing passenger name	Passenger name missing

40422	Missing passenger name part	Part of passenger name is missing
40423	Unknown passenger courtesy title	Unknown courtesy title for passenger
40424	Invalid birth date format	Invalid date of birth format
40425	Missing passenger document	Passenger document missing
40426	Missing passenger document part	Part of the passenger's document is missing
40427	Unknown flight	Unknown flight
40428	Invalid departure date format	Invalid departure date format
40429	Invalid arrival date format	Invalid arrival date format
40430	Unknown passenger type	Unknown passenger type
40431	Unknown fare	Unknown tariff
40500	Error while sending documents	Error sending documents
40501	Missing email setting	Missing email settings
40502	External pnr not found	External PNR not found
Other additional services errors		
5000	Certificate is invalid	The certificate is not valid
5001	A partner type is not available for current App Type	Partner type is not available for the current application type
5002	The service must be added for the entire route, but was not added in one of the segments	The service should have been added for the entire route but was not added to any of the segments
5003	The passenger must be older than {InsuranceProductsAgeRestrictions.MinCustomerAge} age	The passenger must be over the age required to receive the insurance product according to the terms of the insurer
5004	Unable to book Amadeus insurance along with partner insurance product	Unable to book Amadeus insurance together with a partner insurance product
5005	Can't add veterinary without a pet in PNR	Unable to add a vet without a pet to the PNR
5006	Can't be sold to non-citizens of the Russian Federation	Service cannot be sold to a non-citizen of the Russian Federation
5007	According to the business rules these insurances cannot be purchased in one order	Due to business conditions, these insurances cannot be purchased in one order.
5008	The product is not available for chosen fares	The product is not available for the selected tariffs
5009	The service is missed but required	The required service is not available

5010	Unable to purchase the service after registering for a destination	Unable to purchase service after registration
5011	Current api key doesn't support Auto Checkin service	The current API key does not support automatic registration.
5012	All passengers should have seats to add Auto Checkin service	For automatic check-in, each passenger must have a seat selected.
5013	Only reservations with regular flights and inner country flights supports autocheckin	Automatic check-in is supported only for scheduled and domestic flights.
5014	Reservations contain services that incompatible with Auto CheckIn	Bookings contain services that are not compatible with the automatic check-in feature
5015	Auto checkin service expired	The automatic registration service has expired.
5016	Unequal number of autocheckin on all segments in the same direction	Inconsistency in the number of auto-registrations on all segments in one direction
5017	A partner type is not available	The selected partner is not available
5018	Not available on codeshare bookings	Not available for codeshare flights
6000	The reservation contains unaccompanied minor without UMNR service	Booking includes unaccompanied minor without UMNR service.
6001	Price must be in RUB currency	The price must be in rubles
6004	Nothing to pay error	Error "no payment options available"
6005	Operation not supported for this passenger type	Operation not supported for this passenger type
6006	Passenger cannot pay the booking. Please use the office where you ordered the booking	The passenger is unable to pay for the booking. Please use the office where you made the booking
6007	Passenger cannot pay the booking. Please contact the call center	The passenger is unable to pay for the booking. Please contact the call center
6008	Business lounge services return error	Error while receiving business lounge booking service
6009	Business lounge services not available for standalone children, a adult is mandatory	The business lounge booking service is not available for children without adults
"Time to think" service errors		
6002	Time to think expired	Time for the "Time to Think" service has expired
6003	Time to think is not confirmed	The service "Time to Think" is not confirmed
6010	Error while sending Time to think info	Error sending information about the service "Time to Think"

6011	Contacts not found while sending Time to think info	Contact details for sending data within the framework of the service "Time to think" were not found
6012	Tickets already bought while sending Time to think info	Tickets have already been purchased as part of the "Time to Think" service
6013	Service not found while sending Time to think info	Service not found when sending data "Time to think"
8003	The time to think service should be added to all passengers in the booking, except for infants without seat	The "Time to Think" service must be added to all passengers in the booking, except infants without a seat
8004	The Time to Think service cannot be added to an infant	The "Time to Think" service cannot be added to an infant
API Weather errors		
6100	Weather service return success: false. It is possible airport not found	Error retrieving weather data. Airport not found
6101	Weather service is temporarily unavailable	The weather data service is temporarily unavailable
6102	Incorrect weather service response format	Incorrect weather request format
Errors of getting flights status		
7000	Flight statuses search general error	General error searching for flight status
7001	OnlineDashboard / Flight Status API is temporarily unavailable	Online board / Flight status API is temporarily unavailable
BPC side errors		
8000	Request to BPC did not complete	BPC request not completed
8001	Payment gateway connectivity error	Error connecting to payment gateway
8002	Payment gateway connection timeout is exceeded	Payment gateway connection timeout
9001	Received markups report error	Error while getting markup report
9002	Applied markups report error	Error in reports on applied markups
9003	Seat map is not available due to incorrect billing	Seat map unavailable due to incorrect billing
9004	Unknown IBE error. Null Ok Response	An error occurs if the response to a request to create a booking is returned empty

9005	An error occurred while requesting an external service	Error while requesting external services
9006	Unreachable code executed	Invalid code executed
9007	Error while validation SitCloseCondition	Error validating parameter SitCloseCondition
9050	At the moment, booking changes are not available. Ticketing required	Reservation changes are currently unavailable. Tickets are required
Subsidized flights errors		
9100	Subsidized API returned error	General API error
9101	All passengers do not have quotas available to book the subsidized fare	Not all passengers have quotas for booking a subsidized fare
9102	Some passengers do not have quotas available to book the subsidized fare	Some passengers do not have quotas to book a subsidized fare
9103	Subsidized API timeout error	API connection timeout
9104	Subsidized API connectivity error	API connection error
9105	Subsidies booking creation is forbidden for user	Booking a subsidized flight is prohibited for the user
9106	Subsidized fare mismatch on reservation creation	Subsidized rate mismatch when creating a booking
9107	Unknown subsidized fare	Unknown subsidized tariff
9110	Need to wait for documents confirmation	It is necessary to wait for confirmation of documents
9111	Identity of some passengers were not confirmed	Some passengers have not been identified
9112	Documents of some passengers were not confirmed	Some passengers' documents have not been verified
9113	Some passengers subsidies dates does not match flight dates	Subsidy dates for some passengers do not coincide with flight dates
9114	Error while creating booking from subsidies session	Error creating subsidized flight booking
9115	Error while annulling booking from subsidies session	Error canceling subsidized flight reservation
9116	Error while adding a crm request id remark	Error on the CRM side when adding a remark to a booking

9120	Error when sending subsidies confirmation status email	Error sending subsidy status confirmation to email
9121	Error when sending subsidies confirmation status email. Contacts missing	Error sending confirmation of subsidies to email. Contact details missing
9130	Error when sending subsidies confirmation request	Documents for subsidies have not been verified
9131	Error when sending subsidy files	Error sending subsidy confirmation files
9140	Subsidies confirmation session is not exists	Subsidy document confirmation session missing
9141	Subsidies confirmation session is not waiting for passenger files	Missing subsidy document confirmation session, missing files
9142	Subsidies confirmation session is already exists	Subsidy confirmation session already created
9150	Error while uploading subsidy files to external storage	Error loading documents confirming the availability of subsidies to external storage
SMS notification service errors		
9200	Re-booking is not available as there is a paid service	Re-booking is not available as it is a paid service
9201	Current api key doesn't support sms service	The current API key does not support SMS notifications
9202	Phone number (text field) is missing or incorrect	The phone number in the text field is missing or incorrect
9203	Sms service cannot be added to an infant	SMS notification cannot be added to the infant
9204	Sms service cannot be added to the soon departing flight	SMS notifications cannot be added to an upcoming flight

8_3 Attachment: Partner's services.

Partner codes

Partner code	Title
1	Alfainurance
2	Sealine insurance
3	Service
4	Aeroexpress

7	Telemedicine (SberHealth)
8	Baggage packing
11	Alfastrah (Anticovid)
12	Sealine (Insurance AntiVirus)
13	Business room
21	Alfamedicine

Partner types

Partner type	Decryption
1	Insurance
2	Aeroexpress
3	Medicine
4	Packing

Aeroexpress service codes

№	Service code	Tariff decryption
1	STANDARTOW	Standard
2	STANDARTRT	Round trip
3	FAMILYRT	Family: Round trip
4	BUSINESSOW	Business
5	BUSINESSRT	Business: Round trip

Tariff description can be found on official Aeroexpress website: <https://aeroexpress.ru/aero/prices.html>

Alfainsurance 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	SLPROMODYNAMIC	Promo with dynamic cost calculation	Yes
6	SLTRAVEL	Travel	Yes

Sealine service codes (COVID insurance)

№	Service code	Tariff decryption
1	SLANTICOVID	Insurance AntiCovid

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

Luggage packing service codes

№	Service code	Tariff decryption	Availability
1	SWRP	Standart packing	Yes
2	DWRP	Double packing	Yes

Business rooms service codes

№	Service code	Tariff decryption	Availability
1	BLN	Business room	Yes
2	VLN	VIP room	Yes

8_4_Attachment: service SSR

IBE adds PNR SSR 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/LESSX.XXXX/SegmentReference/PassengerReference

Examples:

DISCOUNT/DS20/1/1-5KYLTGV/LESS34.6741/S12/P12

DISCOUNT/DS07/0/1-4NJKZL1/LESS6.9954/S12/P12

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
LESSX.XXXX	X.XXXX - actual percentage of discount from base fare (taking into account the rounding of the discount amount to a multiple of 5 down)	Example: LESS6.9954 LESS34.6741
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	-

SSR with information about actual amount of discount from base fare

Format: PREVBASEDIFF/XXX

Example:

PREVBASEDIFF/305

Field	Description	Comment
PREVBASEDIFF	Constant	-
XXX	Actual amount of discount in rubles after rounding down to a multiple of 5.	-

SSR with information about "Wings" card

Description: SSR is added in tariff booking with discount

Format: type = "FQTV", text="U6XXXXXXXXXXXX" (max length of the field text = 18 chars)

Example: text = U61000761775

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/LESSX.XXXX/SegmentRefernece/PassengerReference

Example: PROMO/WWJ0MTMV/1-48MLQR4/DS04/LESS3.7344/S12/P12

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
LESSX.XXXX	X.XXXX - actual percentage of discount from the base fare (taking into account the rounding of the discount amount to a multiple of 5 down)	Example: LESS3.7344
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	-

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:

PTR/PartnerId/ServiceId/Code/OrderId/Currency/Amount/Status/RefundAmount/PassengerReference/InfantFlag/SegmentReference/ppr

Example (Aeroexpress):

"PTR/4/65241A17717E4BEEB378B82763795A03/STANDARTOW/520501/RUB/500.00/1/0.0
0/P2///26250311"

Field	Description	Comment
PTR	Constant	-
PartnerId	Partner's unique identifier	Possible types described in Attachment 3.
ServiceId	Unique order's id	GUID
Code	Partner's service code	Possible types described in Attachment 3
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	A link to a segment is not added if the partner service is added to all segments. Currently only used for baggage packing partner. Example: S12
IssuedOffice	Ppr of the office under which the partner was added to the booking	-

Service SSRs for subsidized fares

Description: SSRs are added on subsidized fare bookings.

Format: type = "SPAX", text="SUBSIDIZED TARIFF".