

EN_Internet Booking Engine (Sirena-Travel)

API documentation

Document version	v2.3.94 (Sirena) 2.3 - recommended API version 94 – sprint number where recent changes were introduced
Supported API versions	v2.2, v2.3

Change history

Date (dd.mm.yy)	Version	Changelog	Author
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 /Reservation/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>	Gudumak S.E.

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>
------------	--------	---	--

26.07.2022	2.3.88	<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>
------------	--------	--	--

07.07.2022	2.3.87	<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>
------------	--------	--	---------------------------------------

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.

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.

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.

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: 1. Adding flights information 2. Adding passenger's data 3. 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	<i>GET api/partners/types</i>	Optional Retrieve list of available partners.
8	<i>GET api/partners/[id]</i>	Optional Retrieve list of services available for specific partner. Where id is partner's number (presented in attachment)

9	<i>POST api/partners</i>	Optional. Adding of partner's service to booking.
10	<i>POST api/payment</i>	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 paid 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

API methods description

Obtain session service

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

POST session method

Description: new session creation

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

Request format /api/apiVersion/session

Request header

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

Request paramaters

No parameters

Response format

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

Response parameters

Parameter	Description
sessionKey	Session key Used for storing of booking context

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

Response sample

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

Obtain minimal prices service (unavailable for agents)

API service	calendar
Short description	Obtaining of minimal ticket prices by dates. Mostly used for price display in calendar
Supported HTTP(S)-methods	GET
Conterminous systems methods used	Sirena: calendar_e2e SIG: SIG_AirShop, SIG_AirMinPrice CRM: CalculateDiscountList2

GET calendar method

Request format: /api/apiVersion/calendar

Request samples:

One way flight, search through Sirena:

/api/apiVersion/calendar

?orig=LED&dest=MOW&dep=20220817&mh=false&disc=false&cache=0&twp=false&adt=1
&chd=0&inf=0&src=0&yth=0&umnr=0&str=0&jnn=0&gcf=0&jcb=0

Two way flight, search through SIG:

/api/apiVersion/calendar

?orig=LED&dest=DME&dep=20220610&ret=20220620&adt=1&chd=0&inf=0

Request parameters

IMPORTANT: only non-subsidised (ADT/CHD/INF) set of passengers types should be used in request.

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 Sample: 20180801	
ret	date	Return date	Формат yyyyMMdd Пример: 20180905	
adt	integer	number of Adult passengers	Integer number in interval [0; 9]. Overall number of passengers must not exceed 9	
chd	integer	number of Child passengers		
inf	integer	number of Infant passengers		
src	integer	number of Senior citizen passengers		
yth	integer	number of Youth passengers		
cnn	integer	number of passengers of «subsidised child» type		
<i>str</i>	integer	number of passengers of subsidized adult for the Far East		In developing
<i>jnn</i>	integer	number of passengers of subsidized children for the Far East		In developing
<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>mtb</i>	boolean	obtain values grouped by months false – standard request, true – request by months	false	
<i>twp</i>	boolean	Request a value for a period of +-7 days from the current date or not. false - standard request, true - request for values +-7 days from the current date	false	
<i>cache</i>	string	Determines whether cache data should.	0	In developing
<i>disc</i>	boolean	Service parameter, allows to control discount querying. Default value - true. disc = false used in querying information for swiper interface elements. disc = true used for querying of information for calendar with discounts available by months.	-	In developing
<i>accountCode</i>	string	Optional parameter. Used to display prices for specific account codes.	ACCadd005	
<i>meta</i>	boolean	Sign of metasearch.		
<i>currency</i>	string	Currency of recommendations.	EUR	

Response format

```
{
  "data": {
    "prices": [{
      "price": number,
      "fareAmounts": [{
```

```

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

```

Response parameters

Parameter	Description	Notes
-----------	-------------	-------

success			Sign of query execution success. If errors array is empty, field value will be set to true	
data			Returned data	
	range		Date interval for which data was returned	
	currency		Currency of recommendations.	
	prices		Array of objects with price information	
		price	Minimal price for one way flight for date specified in prices.date field	
		date	One way flight date with minimal price returned by service Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-03T00:00:00	
		flightsNumbers	Array of strings with flights number's information for specific flight direction. Contains several values if flight contains transfers.	
		fareAmounts	Array of objects with ticket price information for specific passenger	
		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	
		discount	Object with information about discount available for present flight Discount may absent for certain transfers if flights are in the list of exceptions. Exceptions stored on CRM and IBE side.	In developing
		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	
			fareFamilies	Strings array with information about fareFamilies for which discount may be applied.	
			returnPrices	Array of objects, which contains minimal prices for return flight dates which matches with dates specified in prices.date.	
			price	Return flight price for returnPrices.date date	
			date	Date of return flight with minimal price returned by service	
			flightsNumbers	Array of strings with flights number's information for specific flight direction. Contains several values if flight contains transfers.	
			fareAmounts	Array of objects with ticket price information for specific passenger Analogue of data.prices.fareAmounts. For one way flight array will be empty.	

				paxType	Passenger type. Possible values: Attachments, passenger types	
				paxCount	Amount of passengers of specific type	
				taxAmount	Tariffs and fees	
				totalAmount	Full ticket price including tariffs and fees	
errors					Information about errors appeared during query execution.	
				ibeCode	Error code.	
				description	Code description.	
				messages	Array of objects. Array can be empty, it is recommended to analyse errors by ibeCode.	
				code	Error code	
				text	Error text	

Error request and response samples

Request:

api/apiVersion/calendar?orig=BAX&dest=VVO&dep=20220521&ret=&adt=1&chd=0&inf=0&src=0&yth=0&cnn=0&mth=false&disc=false&cache=0

Parameter dep contains date in the past.

Response:

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

}

Obtain flights service: api/flights

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 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: non-subsidised (ADT/CHD/INF) set of 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 Sample: 20180801	
ret	date	Return date	Формат yyyyMMdd Пример: 20180905	
adt	integer	number of Adult passengers	Integer number in interval [0; 9]. Overall number of passengers must not exceed 9	
chd	integer	number of Child passengers		
inf	integer	number of Infant passengers		
src	integer	number of Senior citizen passengers		
yth	integer	number of Youth passengers		
cnn	integer	number of passengers of «subsidised child» type		
<i>str</i>	integer	number of passengers of subsidized adult for the Far East		In developing
<i>jnn</i>	integer	number of passengers of subsidized children for the Far East		In developing

<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>mode</i>	string	Search mode (optional parameter, affects search via SIG)	mono - only mono-recommendations are requested, for example, PR there and PR back; full - all recommendations are requested.	For the Sirena search now it uses mode=Mono, full - in developing
<i>ofn</i>	string	Array of one-way flight numbers (optional parameter, affects search via SIG)	60	In developing
<i>rfn</i>	string	Array of return flight numbers (optional, affects search via SIG)	371,392	In developing
<i>accountCode</i>	string	Personal account client code (optional parameter)	ACCadd005	
<i>meta</i>	boolean	Sign of metasearch.		In developing
<i>currency</i>	string	Currency of recommendations.	EUR	
<i>cabin</i>	string	Class of service	Economy, Business	In developing
<i>SearchAllBrands</i>	bool	Search for all brands		In developing

Response format

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

```

"airShopSessionId": string,
"currency": string,
"outboundFlights": [flightGroup],
"returnFlights": [flightGroup],
"fareFamilies": [{
    "fareFamilyId": integer,
    "code": string,
    "shortCode": string,
    "category": string,
    "isSubsidized": boolean,
    "order": integer,
    "commercialFamily": string,
    "canRefund": boolean,
    "hasUpsale": boolean "options": [{
        "code": string,
        "category": string,
        "order": integer,
        "showInPopup": boolean,
        "showInSummary": boolean,
        "showInUpsell": boolean,
        "showInFareIncluded": boolean,
        "showForAdult": boolean,
        "showForInfant": boolean
    }]
}]
},
],
"foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string "commercialFamily": string,
    "translation": {
        "ru": string,
        "en": string,
        "zh": string
    }
}]
},
],
"alternativeFlights": [{
    "departure": string,
    "arrival": string
}]
},
],
"discountConstraints": [{
    "fareFamilies": [string],
    "flights": [string],
    "excludedFlights": [{
        "flightNumber": string,
        "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
}],
"returnFareAndAmountDetails": [{
    "bookingCode": string,
    "fareBasis": string,
    "isBreakpoint": boolean,
    "paxType": integer,
    "paxCount": integer,
    "taxAmount": integer,
    "totalAmount": integer
}],
"outboundBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer
}],
"returnBaggageAllowances": [{
    "quantity": integer,
    "weight": integer,
    "passengerType": integer
}],
"discount": {
    "id": string,
    "name": string,
    "amount": integer,
    "discountType": integer,
    "usageType": integer,
    "startDate": datetime,
    "endDate": datetime
},
"promotion": {
    "number": string,
    "amount": integer,

```

```

        "promotionStatus": string,
        "promotionType": string,
        "description": string,
        "startFlightDate": datetime,
        "endFlightDate": datetime,
        "startDate": datetime,
        "endDate": datetime,
        "isUnique": boolean,
        "campaignId": string,
        "directions": [{
            "id": string,
            "origin": string,
            "destination": string
        }
    ],
    "services": [{
        "code": string
    }
],
    "families": [{
        "code": string
    }
],
    "hasConflict": boolean,
    "excludedFlights": [{
        "flightNumber": string,
        "departureAirport": string,
        "arrivalAirport": string
    }
]
    }
    }
    },
    "moneyWallCoefficient": integer,
    "containsInternationalFlights": bool
},
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
]
    }
],
"success": boolean
}

```

flightGroup object

```

{
    "flightGroupId": integer,
    "flights": [{

```

```

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

```

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

```

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

```

Response parameters

Parameter		Description	Notes
success		Sign of query execution success. If errors array is empty, field value will be set to true	
data		Returned data	
	airShopSessionId	SIG search session ID	
	currency	currency in which prices are represented	
	moneyWallCoefficient	Coefficient for calculating the monthly installment payment	
	containsInternationalFlights	Sign that one of the airports is located outside the territory of the RF	
	outboundFlights	Array of flightGroup objects with information about outbound flights	
	returnFlights	Array of flightGroup objects with information about return flights. For one way flight array will be empty	
	fareFamilies	Array of objects with information about fare families	
	fareFamilyId	Fare family ID	
	code	Fare family code, for ex: ECOLIGHTRU/ECOFLEXRU/ECOPREMRU etc.	
	shortCode	Shortened code for fare family identification for example for fare families mentioned above: PR/EC/FL etc.	

	category	Fare family category, for ex: ECOLIGHT/ECOFLEX/ECOPREM Possible values: see Attachment 2, Main fare family categories	
	order	fare families display order	
	commercialFamily	fare family type for ex: U6ECONOMY or U6BUSINESS	
	isSubsidized	Sign of subsidised tariff	
	<i>canRefund</i>	Sign of refundable tariff	For Sirena: in developing
	hasUpsale	Sign indentifying whether present tariff can be upgraded to next one in price category. Used for management of Upsale block display in frontend.	
	options	Array of fareFamiliesOptions objects with information about tariff options. Possible values: see Attachment 2, tariff options	
	<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 analyse 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 <u>transfer</u> 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 passenger nationalities 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
	codeshareName	Full aviacompany title performing code-share flight Example: Czech Airlines for operatingCarrier = OK For operatingCarrier = U6, present field is empty
	hash	service field
	techStops	Array of objects with information about technical stops
	arrivalDateTime	Date and time of arrival to airport where technical stop is performed Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-26T05:30:00
	departureDateTime	Date and time of departure from airport where technical stop is performed Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-26T08:00:00
	airportCode	Airport code
	stopDuration	Technical stop duration
	flightDurationUntilNextStop	Flight duration until next stop

Object fareFamiliesOptions

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

data.fareFamilies.options

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

Object fareDetails

Contains information about tariff details

data.recommendations.outboundFareDetails or data.recommendations.returnFareDetails	
bookingCode	In general case it is first symbol in fareBasis
fareBasis	Base tariff title Example: WPROW или WECOW
isBreakpoint	Flag identifying whether tariff ends on present segment.
paxType	Passenger type. Possible values: Attachments , passenger types
paxCount	Amount of passengers of specific type
taxAmount	Tariffs and fees
totalAmount	Full ticket price including tariffs and fees

isSale	A parameter that defines the sales rate
--------	---

Object BaggageAllowances

Contains information about baggage allowances

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

totalPrice field value examples

One way flight, no discounts

totalPrice is equal to value outboundPrice

```

▼ recommendations: [{outboundFlightGroupId: 0, outboundFareFamilyId: 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 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":
      "eJxdj1EKwjAQRK8yFxB2N5uk7SUUf/odpGBAFKr44a8XKHhh042VKpkEJpO8WRFchzGnU36kW76c
      uy6Nh2O+D2AFIZZNYKcsGiDYcLHlULAni0TQekiAR7/bb3s4+1JXAyERani54AhVaPPNmNibqfnc+
      alwoY0ssVhYi1X/Ele0eVSh2AbyzgjGWIVr+BVTtPD/nlU5PKfXG/geN0w="
    },
    {
      "key":
      "eJxdj1EKwjAQRK8yFxB2N5uk7SUUf/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 the success of the request If the errors array is empty, then the field value will be true

Method GET flights/allFlightStatusesForDay

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

Request format: /api/apiVersion/flights/allFlightStatusesForDay

Request parameters

Parameter	Description
date	Flight date
departureAirport	Departure airport
arrivalAirport	Arrival airport

Request example

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

Response body

```
{
  "data": {
    "flightStatuses": [
      {
        "fnNumber": integer,
        "fnSuffix": string,
        "fnCarrier": string,
        "dayOfOrigin": datetime,
        "departureAirportActual": string,
        "arrivalAirportActual": string,
        "legState": string,
        "legType": string,
        "aircraftRegistration": string,
        "aircraftType": string,
        "departureAirportSched": string,
        "arrivalAirportSched": string,
        "departureDTSchedUTC": datetime,
        "arrivalDTSchedUTC": datetime,
        "departureDTSchedLocal": datetime,
        "arrivalDTSchedLocal": datetime,
        "estimatedArrivalDate": datetime,
        "offBlockDT": datetime,
        "airborneDT": datetime,
        "landingDT": datetime,
        "onBlockDT": datetime,
        "updatedDT": datetime,
        "checkinSinceDTUTC": datetime,
        "checkinUntilDTUTC": datetime,
        "checkinSinceDTLocal": datetime,
        "checkinUntilDTLocal": datetime,
      }
    ]
  }
}
```

```

        "departureDTUTC": datetime,
        "departureDTLocal": datetime,
        "arrivalDTUTC": "2021-10-27T07:40:00",
        "arrivalDTLocal": "2021-10-27T12:40:00",
        "isFlightDelayed": boolean
    }
]
},
"errors": [
    {
        "ibeCode": integer,
        "description": string,
        "messages": [
            {
                "text": string
            }
        ]
    }
]
},
"success": boolean
}

```

Response parameters

Parameter		Description
data		The data returned by the request
	flightStatuses	Array of objects with information about flights
	fnNumber	Flight number
	fnSuffix	Flight suffix
	fnCarrier	Airline operating the flight
	dayOfOrigin	Departure date of the scheduled flight
	departureAirportActual	IATA code of the airport of departure - actual
	arrivalAirportActual	IATA-code of the airport of arrival - actual
	legState	Flight status
	legType	Тип рейса
	aircraftRegistration	Flight type
	aircraftType	IATA aircraft type code
	departureAirportSched	IATA code of the scheduled departure airport
	arrivalAirportSched	IATA code of the scheduled arrival airport
	departureDTSchedUTC	Scheduled departure time UTC

		arrivalDTSchedUTC	Scheduled arrival time UTC
		departureDTSchedLocal	Scheduled departure time Local
		arrivalDTSchedLocal	Scheduled arrival time Local
		estimatedArrivalDate	Estimated time of arrival
		offBlockDT	Departure time from gate
		airborneDT	Take off time
		landingDT	Boarding time
		onBlockDT	Time of arrival at the parking lot
		updatedDT	Date of last update of flight information
		checkinSinceDTUTC	Check-in start for the flight UTC
		checkinUntilDTUTC	Check-in end for the flight UTC
		checkinSinceDTLocal	Check-in start for the flight UTC
		checkinUntilDTLocal	Check-in end for the flight UTC
		departureDTUTC	Departure time UTC
		departureDTLocal	Departure time Local
		arrivalDTUTC	Arrival time UTC
		arrivalDTLocal	Arrival time Local
		isFlightDelayed	Flight delay sign
errors			Information about errors that occurred during the operation of the service.
		ibeCode	Error code, see Attachments 1. IBE error codes
		description	Error description, see Attachments 1. IBE error codes
		messages	Array of objects containing errors from the partner's response. The array may be empty, it is recommended to analyze the error by the 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 body

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

Request body parameters

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

Request body example

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

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

Booking management service: api/reservation

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.

Request format /api/apiVersion/reservation

Request body

```
{
  "flights": {
    "outboundFlights": [reservationFlight],
    "returnFlights": [reservationFlight],
    "adults": integer,
    "children": integer,
    "infants": integer,
    "pensioners": integer,
```



```

    "youths": integer,
    "subsidizedChildren": integer,
    "origin": string,
    "destination": string,
    "passengersCount": integer
  },
  "passengers": {
    "passengers": [passengerData],
    "userAgreement": boolean
  },
  "fares": {
    "accountCode": "string",
    "tourCode": "string",
    "outboundFareFamily": {
      "code": string,
      "category": string,
      "isSubsidized": boolean
    },
    "returnFareFamily": {
      "code": string,
      "category": string,
      "isSubsidized": boolean
    },
    "fares": [{
      "passengerType": "Adult",
      "flightIndex": integer,
      "flightDirection": integer,
      "fareBasis": string,
      "isBreakpoint": boolean
    }
  ],
  "sessionMarker": string,
  "currency": string
},
"expectedPrice": integer,
"searchPrice": integer,
"airShopSessionId": string
}

```

reservationFlight object

```

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

```

passengerData object

```

{
  "id": integer,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": integer,
  "birthDate": datetime,
  "sex": integer,
  "nationality": string,
  "documentNumber": string,
  "documentExpiryDate": datetime,
  "loyaltyCard": string,
  "loyaltyCardValidTo": datetime,
  "serviceClassUpOrdered": boolean,
  "email": string,
  "phone": string,
  "documentType": integer,
  "disability": {
    "isDeaf": boolean,
    "isBlind": boolean,
    "hasGuideDog": boolean,
    "assistanceType": integer,
    "wheelchairType": integer,
    "commentary": string
  },
  "endorsements": [{
    "endorsement": boolean,
    "isDisability": boolean,
    "endorsementDocNumber": string,
    "endorsementDocDate": datetime,
    "loyaltyPartnerCard": string
  }]
}

```

Request parameters

Parameter	Description	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, 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 tariffs	
	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 tariffs for return flight	
	code	Fare family code Data is taken from the response api/flights: data.fareFamilies.code	
	category	Fare family category Data is taken from the response api/flights: data.fareFamilies.category	
	isSubsidized	Sign of a subsidized tariff	Optional parameter

fares		An array with information about tariffs for directions "outbound" and "return"	
	fareBasis	<p>Base tariff title</p> <p>Data is taken from the response <code>api/flights</code>:</p> <p><code>data.recommendations.outboundFareDetails.fareBasis</code> and <code>data.recommendations.returnFareDetails.fareBasis</code></p>	
	isBreakpoint	<p>Flag identifying whether tariff ends on present segment.</p> <p>Data is taken from the response <code>api/flights</code>:</p> <p><code>data.recommendations.outboundFareDetails.isBreakpoint</code> and <code>data.recommendations.returnFareDetails.isBreakpoint</code></p>	
	passengerType	<p>Passenger type</p> <p>Possible values: see Attachment 2. Passenger types</p>	
	flightDirection	<p>Flight direction</p> <p>0 - outbound, 1 - return</p>	
	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	

reservationFlight

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

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

passengerData Request object

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

nationality	Nationality (country) 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-Яa-я]{2})[A-Za-z]{2,4})\d{6}\$. <p>For documents of other countries there is no special format rules.</p>	
documentType	Document type		
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)		
<i>serviceClassUpOrdered</i>	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 assistance type Possible values: dictionary of assistance types		
wheelchairType	Wheelchair type Possible values: dictionary of wheelchair type		
commentary	Comment (optional)		
endorsements	Array with information about documents to buy tickets on the subsidized fares or to apply discounts on partner loyalty cards.		

endorsement	The sign that an endorsement is adding to booking.		
isDisability	Disability endorsements.		
endorsementDocNumber	Subsidy document number. E.g disability certificate, ID number or other document confirming the status of a large family.		
endorsementDocDate	Subsidy document issue date.		
loyaltyPartnerCard	Loyalty partner card number.		

Response format

```
{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "returnFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string,
      },
      "returnFareFamily": {
        "code": string,
      },
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "returnDate": datetime
    },
    "number": string,
    "reservationDate": datetime,
    "passengers": [{
      "id": integer,
      "hasInfant": boolean,
      "referenceNumber": string,
      "firstName": string,
      "middleName": string,
      "surname": string,
      "passengerType": string,
      "birthDate": datetime,
      "sex": integer,
      "nationality": string,
      "documentNumber": string,
      "documentExpiryDate": datetime,
    ]
  }
}
```

```

"email": string,
"phone": string,
"documentType": integer,
"isAdult": boolean,
"isChanged": boolean,
"checkedIn": {
    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,
    "endorsementDocDate": datetime
}]
},
],
"services": [],
"seats": [],
"insurance": {},
"luggageSeats": [{
    "referenceNumber": string,
    "passengerReference": string
}
],
"partnerProducts": [],
"ticketFares": [{
    "baseAmount": integer,
    "passengerReferences": [{
        "referenceNumber": string,
        "referenceQualifier": string
    }],
    "passengerType": integer,
    "referenceNumber": string,
    "referenceQualifier": string,
    "totalAmount": 10964,
    "currency": string
}],
"ticketFaresWithoutZapOff": [{

```

```

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

```

fullReservationFlight object

```

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

```

```

],
"flightDurationUntilNextStop": string,
"arrivalTerminal": string,
"departureTerminal": string,
"hash": string,
    "isCancelledByAirline": boolean
}

```

Response parameters

Parameter	Description	Notes
success	Sign of the success of the request. If the errors array is empty, then the field value will be true.	
data	The data returned by the request.	
number	Unique booking number (PNR).	
reservationDate	Date and time of booking creation Format: yyyy-MM-ddTHH:mm:ss Example: 2018-06-20T15:44:00	
passengers	An array of the passengerData objects with passenger information	
services	An array of objects with information about additional services. Filled in when using the api/services service and will be described below	
seats	An array of objects with information about the selected locations. Filled in when using the api/services service and will be described below	
<i>insurance</i>	Object with information about the selected insurance. Filled in when using the api/services service and will be described below	In developing
<i>luggageSeats</i>	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	
	origin	Initial airport of departure	
	destination	Initial airport of destination	
	departureDate	Departure date	
	returnDate	Return date	
	ticketFares	An array of objects with information about tariffs after applying the discount	
	baseAmount	Base rate	
	passengerReferences	An array of objects with links to passengers	
	referenceNumber	Reference number	
	referenceQualifier	Link type, for example "PA"	
	passengerType	Passenger type Possible values: see Attachment 2. Passenger types	
	referenceNumber	Reference number	
	referenceQualifier	Link type, for example "TST"	
	totalAmount	The total amount of the tariff	
	currency	Reservation currency	

Parameter		Description	Notes
	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	
	<i>hasSpecialConditions</i>	Sign of the additional conditions	
	<i>isRefundAvailable</i>	Sign of the availability of the return	
	<i>annulUnavailabilityReason</i>	Reason for unavailability of booking cancellation	
	<i>isAnnulAvailable</i>	Indication of availability of booking cancellation	
	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	

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	

Parameter		Description	Notes
	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	
<i>arrivalTerminal</i>		Terminal number at the point of arrival	
<i>departureTerminal</i>		Terminal number at the point of departure	
hash		Unique hash of the flight	
<i>isCancelledByAirline</i>		Sign of cancellation of flights by the airline	

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)	

Parameter	Description	Notes
relatedInfantReferenceNumber	Reference to the infant passenger in the booking with whom the adult flies (it returns only for adult)	
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	<p>Whether the passenger is an adult</p> <p>true for ADT/SRC/YTH type passengers</p> <p>false for CHD/INF type passengers</p>	
birthDate	Date of birth	
sex	<p>Passenger gender</p> <p>Possible values: see dictionary "gender of passengers"</p>	
nationality	<p>Nationality (country)</p> <p>Possible values: dictionary of countries</p>	
documentNumber	Document number	
documentType	<p>Document type</p> <p>Possible values: dictionary of document types</p>	

Parameter		Description	Notes
documentExpiryDate		Date until which the document is valid Mandatory only for a foreign passport. For other documents, +10 years from the current date is automatically set	
email		Email	
phone		Phone	
checkedIn		Passenger check-in on flight coupon	
checkedInSegment		Passenger check-in on flights	
checkInHasStarted		Starting check-in on flight coupon	
checkInHasStartedSegment		Starting check-in on flights	
<i>hasPassengersDataErrors</i>		The sign of errors in passengers data.	
<i>passengerDataErrors</i>		Array with detailed information of errors in passengers data.	
	key	The key.	
	value	The value.	

1. Creation of one-way booking, direct.



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

<p>Recommendation:</p> <pre> { "outboundFlightGroupId": 0, "outboundFareFamilyId": 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 } </pre>	<p>Using the outboundFlightGroupId field, determine the flight number that needs to be added and take information on it from the api/flights response, data -> outboundFlights:</p> <pre> { "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "LED", "departureTerminal": "1", "arrivalAirport": "DME", "departureDateTime": "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" } </pre> <p>Substitute the highlighted values, for the flights array and get:</p> <pre> "flights": { "outboundFlights": [{ "origin": "LED", "destination": "DME", "departureDate": "2022-07-16T00:45:00.000Z", "flightNumber": "192", "classOfService": "W" }], </pre>
---	---

	<pre>"passengersCount": 1 }</pre> <p>The classOfService value matches the bookingCode from the recommendation.</p> <p>The passengersCount value depends on the number of passengers added to the booking.</p> <p>Using the outboundFareFamilyId field, determine the number of the fare that will need to be added and take information on it from the api/flights response, data -> fareFamilies:</p> <pre>{ "fareFamilyId": 2, "code": "PR", "category": "ECOLIGHT", "isSubsidized": false, "options": [...], "order": 150, "commercialFamily": "U6ECONOMY", "canRefund": true, "hasUpsale": true, "isLowCost": false }</pre> <p>Substitute the highlighted values, for the fares array and get:</p> <pre>"fares": { "outboundFareFamily": { "category": "ECOLIGHT", "code": "PR", "isSubsidized": false }, "returnFareFamily": null, "fares": [{ "fareBasis": "WPROW", "isBreakpoint": true, "passengerType": 0, "flightDirection": 0,</pre>
--	---

	<pre> "flightIndex": 0 }], "sessionMarker": null } </pre>
--	--

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

ВЫБРАТЬ ДРУГОЙ РЕЙС

С. Петербург LED

06:30

U6-123

5 ч 00 мин

Сочи AER

11:30

от 705 ₺

ПРОМО

705 ₺

ВАШИ ВОЗМОЖНОСТИ

ЭКОНОМ

2 705 ₺

⊘ недоступно

ПРЕМИУМ-ЭКОНОМ

5 205 ₺

⊘ недоступно

БИЗНЕС

24 200 ₺

⊘ недоступно

Direction return

Сочи AER

▼ 08:20

U6-524

8 ч 00 мин



[1 пересадка, U6-256](#)

С. Петербург LED

16:20

от 1 331 ₽

ПРОМО

1 331 ₽

[ВАШИ ВОЗМОЖНОСТИ](#)

<p>Recommendation:</p> <pre> { "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", </pre>	<p>Using the outboundFlightGroupId and returnFlightGroupId fields, we determine the flight numbers and take information on them from the api/flights response, data -> outboundFlights and data -> returnFlights, respectively.</p> <p>Flight outbound:</p> <pre> { "flightGroupId": 0, "flights": [{ "flightId": 0, "departureAirport": "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" } </pre> <p>Flights return:</p> <pre> { "flightGroupId": 3, "flights": [{ "flightId": 0, "departureAirport": "AER", </pre>
---	--

<pre> "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 } </pre>	<pre> "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" } </pre> <p>Substitute the highlighted values, for the flights array and get:</p> <pre> { "flights": { "outboundFlights": [{ </pre>
---	---

	<pre> "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, "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. </pre>
--	---

	<p>The passengersCount value depends on the number of passengers added to the booking.</p> <p>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:</p> <p>Substitute the highlighted values, for the flights fares and get:</p> <pre> "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 }], </pre>
--	--

	<pre>"sessionMarker": null }</pre> <p>The fareBasis values are taken from the recommendation.</p>
--	---

Tariff information should be sent as follows:

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

```

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

```

Method GET reservation

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

Request format /api/apiVersion/reservation

Examples of requests

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

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

Request parameters

Parameter	Description	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, for 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
    }
  ],
  "outboundFareFamily": fareFamilyObject,
  "returnFareFamily": fareFamilyObject,
  "tsmData": [{
    "passengerReference": string,
    "price": number,
    "referenceType": string,
    "referenceNumber": string
  }
],
  "appliedDiscounts": [{
    "flightReference": string,
    "amount": integer,
```

```

        "discountAmount": decimal,
        "id": string,
        "type": string,
        "passengerReference": string,
        "serviceCode": string,
        "promoNumber": string,
        "serviceReference": string,
        "usageType": integer,
        "promotionType": integer
    }
],
"baseFarePrices": [{
    "currency": string,
    "flightReferences": [
        "string"
    ],
    "passengerReference": string,
    "isInfantFare": boolean,
    "totalPrice": number,
    "basePrice": number,
    "priceHidden": boolean,
    "taxDetailsInformation": [{
        "taxCategory": string,
        "taxDetails": [{
            "rate": string,
            "currencyCode": string,
            "type": string
        }
    ]
}]
},
"foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string
    "commercialFamily": string,
    "ru": string,
    "en": string,
    "zh": string
}],
"reservationRetrieveStatus": integer,
"ticketCouponStatuses": [{
    "ticketNumber": string,
    "couponStatuses": {
        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,
        "origin": string,
        "destination": string,
        "departureDate": datetime,
        "returnDate": datetime,
        "outboundBaggageAllowances": [{
            "quantity": integer,
            "weight": integer,
            "passengerType": integer
        }
    ],
        "returnBaggageAllowances": [{
            "quantity": integer,
            "weight": integer,
            "passengerType": integer
        }
    ]
},
    "number": string,
    "reservationDate": datetime,
    "passengers": [passengerObject],
    "services": [{
        "code": string,
        "status": string,
        "text": string,
        "petType": string,
        "category": string,
        "passengerReference": string,
        "flightReference": string,
        "referenceNumber": string,
        "packReferenceNumber": string,
        "price": number,
        "issueDate": datetime,
        "issuedByThirdPartyOffice": boolean
    }
]

```

```

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

```

fareFamily object

```

{
  "fareFamilyId": integer,
  "code": string,

```



```

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

```

fullReservationFlight object

```

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

```

passenger object

```

{
  "id": integer,
  "hasInfant": boolean,
  "referenceNumber": string,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": integer,
  "isAdult": boolean,
  "birthDate": datetime,
  "sex": integer,

```

```

    "nationality": string,
    "documentNumber": string,
    "documentExpiryDate": datetime,
    "email": string,
    "phone": string,
    "documentType": integer
}

```

Response options

Parameter		Description	Notes
success		<p>Sign of the success of the request.</p> <p>If the errors array is empty, then the field value will be true</p>	
data		The data returned by the request	
	tickets	Information about issued tickets	
	flightReferences	Flight segment	
	number	<p>Ticket number.</p> <p>Example: 262 2439494143</p>	
	passengerReference	A reference to the passenger to whom this ticket belongs.	
	isInfantTicket	An sign that this passenger has a baby	
	issueDate	Date and time of issuing the tickets.	
	outboundFareFamily	An array of farFamily objects with information about the selected fares for the “outbound” flight	
	returnFareFamily	An array of farFamily objects with information about the selected fares for the "return" flight	

Parameter		Description	Notes
	tsmData	An array of objects with information about additional services added to PNR	
	passengerReference	Reference to the passenger to whom this service relates	
	price	Service cost	
	referenceType	The type of reference to the entry in the PNR, usually "SSR"	
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR	
	appliedDiscounts	Discounts applied in this PNR.	
	flightReference	Reference to the segment for which the discount was applied	
	amount	Discount rate as a percentage	
	discountAmount	The absolute value of the discount in rubles	
	id	ID of the campaign in CRM for which the discount was applied	
	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.	

Parameter			Description	Notes
		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)	
		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.	

Parameter					Description	Notes
				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 English 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.	

Parameter		Description	Notes
	ticketNumber	Ticket number.	
	couponStatuses	Statuses of flight coupons.	
	couponStatusesBySegment	Statuses of flight coupons for flight segments.	
	engineType	<p>The system in which the reservation was created.</p> <p>Can take values:</p> <p>0 - Unknown,</p> <p>2 - Tais,</p> <p>4 - Sirena</p>	
	<i>isGroupReservation</i>	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	<p>The office under which the reservation was created.</p> <p>For example, SVXU608AZ</p>	

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

Parameter			Description	Notes
		returnBaggageAllowances	An array of objects containing detailed information about baggage allowances for the “return” route	
		<i>containsInternationalFlights</i>	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	
	services		Array of objects with information about additional services added to the booking	
		code	Additional service code Possible values: dictionary of additional codes services	
		status	Service adding status. Example: HK - service added successfully	
		text	Text in ssr format for sending pet information. Example: DOG 1PC 8KG 45X35X25	
		petType	Pet type Possible values: DOG, CAT, PARROT, CANARY	
		category	Additional service category Possible values: dictionary of additional service codes	

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

Parameter		Description	Notes
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR.	
	packReferenceNumber	Reference to the service package in which the location was added	
	issueDate	Date and time of issuing the service.	
	<i>issuedByThirdPartyOffice</i>	A sign that the service was issued by a third-party office in relation to the office under which the booking was issued.	
	<i>insurance</i>	Object with information about the insurance added to the booking	In developing
	number	Insurance document number	
	code	Insurance code Possible values: dictionary of additional services (category insurance)	
	status	Service adding status. Example: HK - service added successfully	
	referenceNumber	SSR number. Each service has a unique SSR number in the PNR.	
	packReferenceNumber	Reference to the service package in which the service was added	
	issueDate	Date and time of issuing the service.	

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.	
	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	
	hasSpecialConditions		The sign if one of the following services has been added to the booking : PETC, CBBG, WEAP, UMR, MAAS, MEDA, EXST, STCR, WCHR, WCHC, WCHS, WCB, WCBW, WCMP, WCOB, DEAF, BLND, DEPA, DEPU	
errors			Information about errors that occurred during the operation of the service	
		ibeCode	Error code	

Parameter		Description	Notes
	description	Error description	
	messages	An array of objects containing errors. The array may be empty, it is recommended to analyze the error by the <code>ibeCode</code> 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
aircraft	Board number
aircraftIataCode	IATA aircraft code

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

outboundBaggageAllowances and returnBaggageAllowances

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

passengerData

Parameter	Description
id	Passenger's serial number (numbered from 1)
hasInfant	An indication that this passenger has a baby

Parameter	Description
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)
firstName	Passenger's name
middleName	Passenger's patronymic (may be absent)
surname	Passenger's surname
passengerType	Passenger type. Possible values: dictionary of passenger types
isAdult	The sign if the passenger is an adult true for passengers type ADT/SRC/YTH false for passengers type CHD/INF
birthDate	Date of birth
sex	Passenger gender Possible values: dictionary gender of passengers
nationality	Nationality (country) Possible values: dictionary of countries
documentNumber	Document number

Parameter	Description
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

Method DELETE reservation

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 parameters

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

Response format

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


Response parameters

Parameter		Description
success		Sign of the success of the request. If the errors array is empty, then the field value will be true
errors		Information about errors that occurred during the operation of the service.
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from 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 v 2.3:

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

Response format

```
{
  "data": {
    "journey": {
      "outboundFlights": [fullReservationFlight],
      "outboundFareFamily": {
        "code": string
      }
    },
  },
}
```

```

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

```

```
}
```

fullReservationFlight object

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

fareBasisCodesFlightRefs object

```
{  
  "fareBasis": string,  
  "flightReferenceNumber": string,  
  "passengerReferenceNumber": string,  
  "expirationUtc": datetime  
}
```

Response parameters (short format)

Parameter	Description	Notes
success	Sign of query execution success. If errors array is empty, field value will be set to true	
data	Returned data	
<i>isGroupReservation</i>	IT/BT reservation sign	In developing
<i>isCompanyGroupReservation</i>	Group booking sign	In developing
number	PNR number	
reservationDate	Booking creation date and time	
bookingTimeLimit	Date and time until which the unpaid booking will exist (rules are set in the settings in IBE)	
passengers	Array of passengerData objects with information about passengers	

services	Array of objects with information about auxiliary services.	
seats	Array of objects with information about selected seats.	
insurances	Array of objects with information about selected insurances.	In developing
partnerProducts	Array of objects with information about partner's services.	
journey	Information about flights for present booking.	
isAgentReservation	Agent booking sign to disable the booking management module	In developing
isPayForbiddenForAgency	Sign that payment is not available for an unpaid booking	
fareBasisCodesFlightRefs	Array of objects with fare basis info	
errors	Information about errors appeared during query execution.	

insurances structure description

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

Itinerary receipt receive and send to email service (unavailable for agents)

API Service	receipt
Short description	Allows to receive Itinerary receipt for present booking and send it to email.
Supported HTTP(S)-methods	GET, POST
Conterminous systems methods used	Sirena: pnr_status, order

GET receipt method

Description: method requests information about current reservation in Sirena and returns PDF with itinerary receipt.

Note: there are two ways of using this service:

1. Before the requesting the method need to request the information about reservation by using the GET api/reservation.
2. Without additional methods need to pass the query parameters: pnr and surname.

Request format /api/apiVersion/receipt

Request parameters

For the first way: header X-Session

For the second way:

Parameter	Description
pnr	PNR number of reservation which need to download
surname	Surname of one of the passengers in reservation.

Response format

PDF-file with itinerary receipt.

Failed request and response examples

If the booking information is not found in the context and the pnr and surname parameters weren't passed in the request.

{

```

"errors": [
  {
    "ibeCode": 1008,
    "description": "Data validation error.",
    "messages": [
      {
        "code": "Pnr",
        "text": "'Pnr' must not be empty."
      },
      {
        "code": "Surname",
        "text": "'Surname' must not be empty."
      },
      {
        "code": "Context.BookingData",
        "text": "Invalid reservation state"
      }
    ]
  }
],
"success": false
}

```

POST receipt method (in developing)

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

Request body

```

{
  "email": string
}

```

Request parameters

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

Response format

```

{
  "errors": [{
    "ibeCode": number,
    "description": string,

```

```

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

```

Response parameters

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

Auxiliary services management service: api/services, api/catalogue

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 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 query execution success. If errors array is empty, field value will be set to true	
data	Returned data	
services	Array of objects with information about airline's available services	

		code	Service code Possible values	
		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	
		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	
	order	Display index in auxiliary services page. 1 - page top	
	fullDirection	Service for directions	
	onlyForRussianCitizens	Service only for russian citizens	
	partnerProducts	Array of objects with information about partners services	
	partnerId	Partners unique ID in IBE. Alfastrakh – 1, Sealine insurance – 2, Aeroexpress – 4	
	partnerType	Partner type Insurance – 1, Aeroexpress – 2	
	code	Partner's service code. Each partner has its own unique set of codes.	
	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	

POST services method

Description: allows you to add services to the booking.

Request format: /api/apiVersion/services

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
	text	Animal type
	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
	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.

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>
<p>Changed the seatServices.flightReferences.referenceNumber array of type string to seatServices.flightReferences.referenceNumbers array of type string.</p>	
<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>
<p>In version 2.3, the partnerProducts array has been removed.</p>	
<p>In the array services[].flightReferences[] added information about available quotas for returned service (this array is returned only for service with the code = "PETC")</p>	

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

Service management of additional services v2.3

Method structure is described in api/reservation version 2.3.

Examples of requests and responses

Error adding airline service (invalid service code)	
<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>
Error when adding airline service (invalid reference to passenger)	

<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: `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 body

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

Request parameters:

Parameter	Description
services	Array of objects with information about added services

Parameter		Description
	code	Additional service code
	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	Link to paid services. Taken from the GET api/reservation response.
seats		Array with objects about seats
	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	Link to paid seats. Taken from the GET api/reservation response.

Method for removing services from booking `api/services/remove`

Description: a method for removing services from an order, provided that no ticket has been issued for the service and the flight segment for which the service was added has not yet departed

Request format `api/apiVersion/services/remove`

Request body same as request body `api/services/add`

Method POST `api/services/favorite`

Description: allows you to get your favorite services for passengers

Request fromat `/api/apiVersion/services/favorite`

Request parameters

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

Request body example

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

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

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

Request body example

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

Service for obtaining a cataalog of servivces 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: get information about catalog of additional services without PNR

Request format: api/apiVersion/CatalogueStandalone

Request body


```

{
  "flights": {
    "outboundFlights": [{
      "origin": string,
      "destination": string,
      "departureDate": datetime,
      "arrivalDate": datetime,
      "flightNumber": string,
      "classOfService": string,
      "marketingCarrier": string,
      "operatingCarrier": string,
      "flightIndex": integer
    }
  ],
  "returnFlights": [{
    "origin": string,
    "destination": string,
    "departureDate": datetime,
    "arrivalDate": datetime,
    "flightNumber": string,
    "classOfService": string,
    "marketingCarrier": string,
    "operatingCarrier": string,
    "flightIndex": integer
  }
],
},
"fares": {
  "outboundFareFamily": {
    "code": string,
    "category": string,
    "isSubsidized": boolean
  },
  "returnFareFamily": {
    "code": string,
    "category": string,
    "isSubsidized": boolean
  },
  "fares": [{
    "passengerType": string,
    "flightIndex": integer,
    "fareBasis": string,
    "isBreakpoint": boolean
  }
],
},
"passengers": [{
  "id": integer,
  "firstName": string,
  "middleName": string,
  "surname": string,
  "passengerType": string,
  "birthDate": datetime,
  "sex": string,
  "loyaltyCard": string,
  "disability": {

```

```

        "isDeaf": boolean,
        "isBlind": boolean,
        "hasGuideDog": boolean,
        "assistanceType": integer,
        "wheelchairType": integer,
        "commentary": string
    },
    "hasInfant": boolean,
    "nationality": string
}
],
"currency": string
}

```

Request parameters

Parameter		Description	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	
fares		Tariff information object	
	outboundFareFamily	Object with information on fares for "outbound" directions	
	code	Fare family code. Data is taken from api/flights response: data.fareFamilies.code	
	category	The category of the tariff family. Data is taken from api/flights response: data.fareFamilies.category	
	isSubsidized	Sign of a subsidized tariff.	In developing
	returnFareFamily	Object with information on fares for the "return" route	
	code	Fare family code. Data is taken from api/flights response: data.fareFamilies.code	

Parameter		Description	Note
	category	The category of the tariff family. Data is taken from api/flights response: data.fareFamilies.category	
	isSubsidized	Sign of a subsidized tariff.	In developing
	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) (required parameter)	
	firstName	Passenger's name	
	middleName	Passenger's patronymic	
	surname	Passenger's surname	
	passengerType	Passenger type (required parameter)	
	birthDate	Date of birth	
	sex	Passenger gender	

Parameter	Description	Note
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	
currency	Search currency	

StandaloneInputFlight

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

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

Response format

```

{
  "data": {
    "services": [{
      "code": string,
      "flightReferences": [{
        "referenceNumber": 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]
  }]
}

```

```

    }
  ],
  "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
},
"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 success of the request. If the errors array is empty, then the field value will be true	
data				The data returned by the request	
	services			An array of objects with information about available airline services	
		code		Service code	
		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	
			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	

Parameter				Description	Note
			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	
			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>	

Parameter		Description	Note
	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	
	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	
	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	

Parameter		Description	Note
	portions	Service field	In developing
	type	Service field	
	number	Service field	
	flightReferenceNumbers	Service field	
moneyWallCoefficient		Coefficient for calculating the monthly installment payment	
errors		Information about errors that occurred during the operation of the service.	
	ibeCode	Error code	
	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	

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: receives information about seatmap scheme in aircraft.

Request format: api/apiVersion/seatmap

Request example

api/apiVersion/seatmap?referenceNumber=1&origin=LED&destination=DME&departureDate=20220723&flightNumber=80&classOfService=E&commercialFamily=U6ECONOMY

Request parameters

Information about flight for which seatmap is needed

Parameter	Mandatory	Description
referenceNumber	Yes	Reference to flight segment
origin	No	Origin airport
destination	No	Destination airport
departureDate	No	Date and time of departure
flightNumber	No	Flight number
classOfService	No	Information about selected tariff
commercialFamily	No	Fare family type, for instance: U6ECONOMY or U6BUSINESS
currency	No	Currency in which search results will be returned

Response format

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

Response parameters

Parameter				Description
success				Sign of query execution success. If errors array is empty, field value will be set to true
data				Returned data
	isEmpty			Flag identifying whether data is returned from 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	Row number for seat
			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
			isSeatUnavailable	Seat availability flag

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

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 body

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


Параметры запроса

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	
sex	Passenger gender Possible values: see dictionary "gender of passengers"	
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 assistance type Possible values: dictionary of assistance types	
wheelchairType	Wheelchair type Possible values: dictionary of wheelchair type	
commentary	Comment (optional)	
hasInfant	An sign that this passenger has a baby	
nationality	Nationality (country) Possible values: dictionary of countries	
origin	Departure airport	
destination	Destination airport	

Parameter	Description	Note
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).	
currency	Search currency	

Формат ответа

```

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

```

Response parameters

Parameter				Description
success				<p>Sign of the success of the request.</p> <p>If the errors array is empty, then the field value will be true.</p>
data				The data returned by the request.
		isEmpty		<p>Flag identifying whether data is returned from sirena</p> <p>This field can be used to determine possibility of selecting seats on present flight segment</p>
		compartmentDetails		Array of objects with information about seats grouped by rows
			number	Row number
		seats		Array of objects which contains detailed information about seats
			seat	Seat. Possible values: A-F
			number	Row number for seat
			description	<p>Seat description</p> <p>9 – central seat</p> <p>A – aisle seat</p> <p>W – window seat</p>
			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
			isSeatUnavailable	Seat availability flag
			hasInfant	Flag identifying that passenger travels with infant

Parameter				Description
			classDesignator	Information about seat class. Possible values: ECONOMY, BUSINESS
		passengersTotalData		An array of objects with information about the cost of seats in the context of passengers
			passengerReference	Passenger reference
			passengerPriceLevels	An array of objects of the cost of seats available to the passenger
			price	Seat cost
			priceLevel	Seat value category
errors				Information about errors appeared during query execution.
			ibeCode	IBE error code
			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

Cancellation service

API service	annul
Short description	Allows to cancel tickets and receive information on canceled bookings
Supported HTTP-(S) methods	GET, POST
Используемые методы смежных систем	Sirena: Getting information on booking - order Cancel (part or all) PNR seats (without tickets refund) - release_seats

POST api/annul method

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

Request format: /api/apiVersion/Annul

Request parameters: X-Session is passed in the header

Response format:

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

Response parameters:

Parameter	Description
success	Sign of the success of the request. If the errors array is empty, then the field value will be true
errors	Information about errors that occurred during the operation of the service.

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

Response format:

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

Response parameters:

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

Partial cancellation service

API service	annul/partial
Short description	Allow to cancel certain tickets from reservation
Supported HTTP-(S) methods	POST
Used methods of related systems	Sirena: Getting information on booking - order Cancel (part or all) PNR seats (without tickets refund) - release_seats

POST api/annul method

Description: allow to cancel tickets for certain passengers in reservation

Request format: /api/apiVersion/Annul/Partial

Request parameters: X-Session is passed in the header

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 success of the request. If the errors array is empty, then the field value 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 analyse errors by ibeCode.
	code	Error code
	text	Error text

Auxiliary partner's services management service

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

GET partners/types method

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

Validated via API-key passing in method.

Request Format /api/apiVersion/partners/types

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

Request parameters no parameters

Response format

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

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	partnerId	Unique partner's code
	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 analyse errors by ibeCode.
	code	Sirena error code
	text	Sirena error text

Response example (Website application type)

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

Response example (mobile application type)

```
{
  "data": [
    {
      "partnerId": 4,
      "name": "Aeroexpress"
    }
  ]
}
```

```

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

```

GET partners/id method

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

Request format /api/apiVersion/partners/id

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

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

All information will be taken from booking context.

Request parameters

id - partner's code

Response format

```

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

```

```

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

```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	partnerId	Unique partner's code
	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
	preselectedOption	An array of settings for pre-selecting an partners service (in the basket at the third step)
	preselect	Sign of preselection of a partner service
	passengers	An array of passengers for whom a partner service is preselected
	referenceNumber	Passenger reference

		passengerType	Passenger type, see Attachment 2. Dictionary of Passenger Types
		compatibleProductCodes	An array of partner service codes that can be added to a booking at the current rate
		internalError	Internal engine error code
		internalErrorMessage	Description of an internal engine error
		errors	Information about errors appeared during query execution.
		ibeCode	Error code, see EN_Attachment 1. IBE error codes
		description	Code description, see. EN_Attachment 1. IBE error codes
		messages	Array of objects, containing errors from 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
      }
    ]
  }, {
    "partnerId": 7,
    "partnerType": 3,
    "bookingStatus": 0,
    "code": "SBERVET",
    "amount": 300.00,
    "currency": "RUB"
  }
],
  "errors": [],
  "success": true
}
```

POST partners method

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

Request format /api/apiVersion/partners

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

Request format

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


Request parameters

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

Response format matches GET api/reservation method response.

Response example

```
{
  "data": {
    "journey": {
      "outboundFlights": [{
        "aircraft": "32N",
        "referenceNumber": "1",
        "origin": "LED",
        "destination": "DME",
        "departureDate": "2020-01-22T14:35:00",
        "arrivalDate": "2020-01-22T16:25:00",
        "flightNumber": "60",
        "classOfService": "E",
        "operatingCarrier": "U6",
        "marketingCarrier": "U6"
      }
    ],
    "outboundFareFamily": {
      "code": "ECOFLEXRU"
    },
    "origin": "LED",
    "destination": "DME",
    "departureDate": "2020-01-22T14:35:00"
  },
  "number": "VY9NCD",
  "passengers": [{
    "id": 1,
    "hasInfant": false,
    "referenceNumber": "2",
    "firstName": "TEST",
    "middleName": "TESTOVICH",
    "surname": "TESTOV",
  ]
}
```

```

    "passengerType": 0,
    "birthDate": "1988-08-11T00:00:00",
    "sex": 0,
    "nationality": "RUS",
    "documentNumber": "14001232244",
    "documentExpiryDate": "2029-12-25T00:00:00",
    "email": "TEST@YANDEX.RU",
    "phone": "79510002233",
    "documentType": 0,
    "isAdult": true,
    "isChanged": false
  }
],
"services": [],
"seats": [],
"partnerProducts": [{
  "partnerId": 1,
  "partnerType": 1,
  "bookingStatus": 1,
  "code": "TRAVEL",
  "amount": 847.00,
  "currency": "RUB",
  "passengerReference": "2",
  "isInfantProduct": false,
  "bookingId": "94cf3c93-b6aa-49c5-8de1-f3fa2749b5e2"
}, {
  "partnerId": 4,
  "partnerType": 2,
  "bookingStatus": 1,
  "code": "STANDARTOW",
  "amount": 500.00,
  "currency": "RUB",
  "passengerReference": "2",
  "isInfantProduct": false,
  "bookingId": "abf3bf5c-acc3-442d-98fa-df317ab439e2",
  "orderId": string
}
],
"hasSpecialConditions": false
},
"errors": [],
"success": true
}

```

Examples of invalid requests and responses

Request	Response
---------	----------

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

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

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: when voiding not all the tickets from the booking split happens and all tickets that are voided are transferred to separated booking, the PNR of which is returned in the response of POST /ticket/void. 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 body

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

Request parameters

Parameter	Description
TicketNumbers	Array of air tickets and additional services to void
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	
	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
success		Sign of query execution success. If errors array is empty, field value will be set to true	

Services for working with queues (for ADP environment)

Service for placing PNR to queues.

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

POST queue/placePNR method

Description: placing the PNR in queues

Request format /api/apiVersion/queue/placePNR

Request parameters

Parameter	Description
queueNumber	Queue number in which need to place the booking
pnr	Booking number which need to be queued

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 query execution success. If errors array is empty, field value will be set to 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

Examples of successful requests and responses

Request

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

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
}

```

Service for retrieving the list of bookings from the queue

API service	queue/list
Short description	The service allows to retrieve the list of bookings from the queue
Supported HTTP-(S) methods	POST
Used methods of related systems	Sirena: queue_show

POST queue/list method

Description retrieving the list of bookings from the queue

Request format /api/apiVersion/queue/list

Request parameters

Parameter	Description
queueNumber	Number of the queue for which the list of bookings is requested to be returned

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 success of the request. If the errors array is empty, then the field value 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 any) 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

Examples of successful requests and responses

Request

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

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
}
```

```
    "success": false
}
```

Service for retrieving the count of records

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

POST queue/counttotal method

Description retrieving the count of records

Request format /api/apiVersion/queue/counttotal

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

Параметр		Описание
success		Sign of the success of the request. If the errors array is empty, then the field value 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

Examples of successful responses

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
}

```

Service for removing PNR from the queues

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

POST queue/removeItem method

Description removing the PNR from the queues

Request format /api/apiVersion/queue/removeItem

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 query execution success. If errors array is empty, field value will be set to 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

Examples of successful requests and responses

Request

```

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

```

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

Discount retrieving service (unavailable for agents)

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

POST discounts method

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

Request format /api/apiVersion/discounts

Request body

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

Request parameters

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

Response format

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

```

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

```

Response parameters

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

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

Obtain recommendations for rebook /api/Exchange/Search (for ADP environment)

API service	/Exchange/Search
Short description	Obtain list of available recommendations for rebook
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: exchange_pricing

Method POST Exchange/Search

Note: allow to get information about the cheapest option for exchange with non-negative amount for surcharge

Request format: api/apiVersion/Exchange/Search

Request headers

X-Api-Key - unique GUID for the systems, using IBE API.

Request body

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

Request parameters

Parameter	Description
reservationNumber	Number of a reservation
lastName	Surname of a passenger

originDestination		An array of data about the part of the flight to be exchanged	origin	/IATA-code of origin airport
			destination	IATA-code of destination airport
	flightNumber	Number of flight		
	classOfService	Class of service		
	departureDate	Departure date		
	segmentReference	Reference to the segment to be exchanged		
	passengerReference	Reference to the segment whose tickets to be exchanged		

Response format

```
{
  "data": {
    "flights": [flightGroup],
    "fareFamilies": [
      {
        "fareFamilyId": integer,
        "code": string,
        "category": string,
        "isSubsidized": boolean,
        "options": [
          {
            "code": string,
            "category": string,
            "order": integer,
            "showInPopup": boolean,
            "showInSummary": boolean,
            "showInUpsell": boolean,
            "showInFareIncluded": boolean,
            "showForAdult": boolean,
            "showForInfant": boolean,
            "showInBookingCart": boolean
          }
        ],
        "order": integer,
        "commercialFamily": string,
        "canRefund": boolean,
        "hasUpsale": boolean,
        "isLowCost": boolean
      }
    ],
    "recommendations": [
```

```

{
  "totalPrice": float,
  "totalAddCollectPrice": float,
  "breakdowns": [
    {
      "ticketNumber": string,
      "passengerReference": string,
      "paxType": integer,
      "baseFareAmount": float,
      "taxAmount": float,
      "totalAmount": float,
      "addCollectAmount": float
    }
  ],
  "flightGroupDetails": [
    {
      "flightGroupId": integer,
      "fareFamilyId": integer,
      "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
    }
  }
],
  "moneyWallCoefficient": float
},

```

```

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

```

flightGroup object

```

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

```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
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
	code	Fare family code, for ex: ECOLIGHTRU/ECOFLEXRU/ECOPREMURU etc.
	category	Fare family category, for ex: ECOLIGHT/ECOFLEX/ECOPREM Possible values: see Attachment 2, Main fare family categories
	order	Fare families display order
	commercialFamily	f\Fare family type for ex: U6ECONOMY or U6BUSINESS
	isSubsidized	Sign of subsidised tariff
	canRefund	Sign of refundable tariff
	hasUpsale	Sign indentifying whether present tariff can be upgraded to next one in price category. Used for management of Upsale block display in frontend.
	options	Array of fareFamiliesOptions objects with information about tariff options. Possible values: see Attachment 2, tariff options
foodOnBoards		Array of objects with information about meals available on the flight
	departureAirport	Airport of departure

Parameter		Description
	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
	recommendations	Array of recommendations objects for flights
	totalPrice	Total ticket price including all applied discounts
	totalAddCollectPrice	Amount to be paid in case of rebook
	breakdowns	Array of tickets that are available for exchange
	ticketNumber	Number of issued ticket
	passengerReference	Reference to a passenger
	paxType	Type of a passenger
	baseFareAmount	Base price
	taxAmount	Tariffs and fees
	totalAmount	Full ticket price including tariffs and fees
	AddCollectAmount	Amount to be paid in case of rebook
	errors	Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Sirena error code
	text	Sirena error text

flightGroup object

Contain information about flights

data.flights	
flightGroupId	Index number used in recommendations
flightGroupDuration	Total flight time: summed up of duration values of each <u>transfer</u> and technical stop.
hash	Service field
flights	Array of objects with information <u>about transfers</u>
flightId	Flight Id
departureAirport	Airport of departure
departureTerminal	Terminal in airport of departure
arrivalAirport	Airport of arrival
departureDateTime	Departure date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T17:25:00
arrivalDateTime	Arrival date and time Format: yyyy-MM-ddTHH:mm:ss Example: 2018-05-27T20:50:00
aircraft	Full aircraft title For ex.: Airbus A320
aircraftIataCode	IATA aircraft code
flightDuration	Duration of flight
flightNumber	flight number
marketingCarrier	Marketing carrier Example: U6, S7
operatingCarrier	Operating carrier (one actually performed flight) Example: U6, OK
hash	service field

Involuntary refund of tickets service (unavailable for agents). Simplified scheme.

Api service	InvoluntaryRefund
Short description	Allows to create involuntary refund request
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: order, pnr_status CRM: Create refund request - ServiceReqInsert_Input

POST InvoluntaryRefund method

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

Request format: /api/apiVersion/InvoluntaryRefund

Request parameters:

Parameter	Mandatory	Description	Comments
pnrNumber	yes	PNR number, unique 6-digit booking identifier	
lastName	yes	Passenger's last name	
passengers	yes	Array of objects with information about passengers.	
lastName	yes	Passenger's last name	
passengerReferenceNumber	yes	Reference to passenger Relates to field data → passengers → referenceNumber	
passengerType	yes	Passenger's type. Possible values: see EN_8_1_Attachment: Dictionaries of passenger types	

	relatedAdultReferenceNumber	no	Reference to the adult passenger with whom the child or infant is flying. The parameter is filled for the child and infant	
	relatedInfantReferenceNumber	no	Reference to the infant with whom the adult is flying. The parameter is filled for the adult who flies with infant.	
	<i>luggageSeats</i>	no	Array with information about services for which the separate tickets were issued.	In developing
	<i>referenceNumber</i>	yes	Reference to the service for which the separate ticket was issued. Maps to a field in the response GET Reservation data → luggageSeats → referenceNumber	In developing
	refundReason	yes	1 - due to flight cancellation 2 - other reason	
	contact	yes		
	firstName	yes	Passenger's first name	
	surname	yes	Passenger's last name	
	middleName	no	Passenger's middle name	
	email	yes	Passenger's email	
	phone	yes	Passenger's phone number	
	comment	yes	Applies only if refundReason = 2 . Comment field in which passengers specifies reason of involuntary refund.	

Request examples:

1. "Due to flight cancellation" case:

-----WebKitFormBoundary9aHnxXNCpJSjJbBQ
Content-Disposition: form-data; name="refundDetails"

```
{
  "pnrNumber": "1ZGD27",
  "lastName": "ZAKHAROV",
  "passengers": [{
    "lastName": "ZAKHAROV",
    "passengerType": 0,
    "passengerReferenceNumber": "12",
    "luggageSeats": [],
    "relatedInfantReferenceNumber": "14"
  }, {
    "lastName": "TIKHOMIROV",
    "passengerType": 5,
    "passengerReferenceNumber": "16",
    "luggageSeats": [],
    "relatedAdultReferenceNumber": "12"
  }, {
    "lastName": "SMIRNOVA",
    "passengerType": 4,
    "passengerReferenceNumber": "14",
    "luggageSeats": [],
    "relatedAdultReferenceNumber": "12"
  }
],
  "refundReason": 1,
  "comment": "",
  "contact": {
    "firstName": "ARTEM",
    "surname": "ZAKHAROV",
    "email": "testuser@arcadialab.ru",
    "phone": "2147483647"
  }
}
```

-----WebKitFormBoundary9aHnxXNCpJSjJbBQ--

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

-----WebKitFormBoundarys1xVgQ9Blka64T30
Content-Disposition: form-data; name="files"; filename="Справка о
порудоспособности.pdf"
Content-Type: application/pdf

-----WebKitFormBoundarys1xVgQ9Blka64T30
Content-Disposition: form-data; name="refundDetails"

```
{
  "pnrNumber": "1ZGD27",
  "lastName": "ZAKHAROV",
  "passengers": [{
```

```

        "lastName": "TIKHOMIROV",
        "passengerType": 5,
        "passengerReferenceNumber": "16",
        "luggageSeats": [],
        "relatedAdultReferenceNumber": "12"
    },
    ],
    "refundReason": 2,
    "comment": "Не смог явиться на рейс по причине болезни",
    "contact": {
        "firstName": "MIKHAIL",
        "surname": "TIKHOMIROV",
        "email": "testuser@arcadialab.ru",
        "phone": "4294967296"
    }
}
-----WebKitFormBoundaryslxVgQ9Blka64T30--

```

Response parameters (short form):

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
crmRefundRequestId	Refund request number in CRM. Example: 1-46MJ19D.
autoRefundCompleted	Sign of autorefund processing
errors	Information about errors appeared during query execution.

Simplified involuntary refund scheme description

1. Client opens BMM (booking management module) using PNR number and passenger's last name.
2. Selects option "Make a return", switched to "Induced" tab.
3. Selects refund reason. By default "Other reason" is set. In this case client should describe reason of involuntary refund and upload document, confirming refund reason. In case of "Due to flight cancellation" it is not necessary.
4. Selects passengers for whom refund is needed to be performed.
5. Fill personal data fields: first/last name, phone number, email.
6. Client selects "Confirm refund" option, api/InvoluntaryRefund method is called.

7. In case of successful creation of refund request, information with refund request ID in CRM is displayed to client.
8. In case of failure one of possible errors is displayed to client.

No refund sum calculation is performed.

No history of involuntary requests is stored in IBE DB.

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

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

Involuntary ticket refund service for agents using.

POST involuntaryRefund/moneyless method

API service	InvoluntaryRefund/moneyless
Short description	Let to process involuntary refund of tickets and services on ADP
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

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

Request format: /api/apiVersion/InvoluntaryRefund/Moneyless

Request parameters:

Parameter		Mandatory	Description
pnrNumber		yes	Unique booking number (PNR).
passengers		yes	Array with information about passengers whom needs to process the involuntary moneyless refund
	isTicketRefund	yes	The sign that a ticket with services or only services is being refunded.
	partnerProducts	no	An array of partner services to be returned (the array parameters are similar to the apiVersion/partners/id response parameters)

	lastName	yes	Passenger's last name
	passengerReferenceNumber	yes	Passenger reference. Associated with field data -> passengers -> referenceNumber
	relatedAdultReferenceNumber	no	Reference to the adult passenger with whom the child or infant is flying. The parameter is filled for the child and infant
	relatedInfantReferenceNumber	no	Reference to the infant with whom the adult is flying. The parameter is filled for the adult who flies with infant.
	passengerType	yes	Passenger type. Possible types of passengers are presented in EN_8_1_Attachment: Dictionaries of passenger types.
	services	no	An array with information about purchased and refundable additional services of the airline.
	code	no	Returned additional service code.
	ticketNumber	no	Ticket number of the returned additional service.

Response format and parameters are the same with POST api/refund/partial method response format and parameters.

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.
Поддерживаемые HTTP(S)-методы	POST, GET
Используемые методы смежных систем	<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 to receive breakdown of refund sums for each passenger.

Request format: /api/apiVersion/refund/calculate

Request parameters:

Parameter	Mandadatory	Description	Comments
pnrNumber	yes	PNR number, unique 6-digit booking identifier	
lastName	no	Passenger's last name	
passengers	yes	Array of passengers for which refund sums needs to be calculated	
lastName	yes	Passenger's last name	

	passengerReferenceNumber	yes	Reference to passenger Linked with data → passengers → reference Number	
	relatedAdultReferenceNumber	no	Reference to the adult passenger with whom the child or infant is flying. The parameter is filled for the child and infant	
	relatedInfantReferenceNumber	no	Reference to the infant with whom the adult is flying. The parameter is filled for the adult who flies with infant.	
	<i>luggageSeats</i>	no	Array with information about luggage with separate air ticket and ssr CBBG.	In developing
	<i>referenceNumber</i>	yes	Ссылка на услугу, для которой оформлен отдельный авиабилет. Соотносится с полем в ответе GET Reservation data → luggageSeats → referenceNumber.	In developing
	passengerType	yes	Passenger type. Possible values: see EN_8_1_Attachment: Dictionaries of passenger types	
	isInvoluntaryMoneyless	no	Indicates whether the calculation of refundable amounts is for performing POST api/InvoluntaryRefund/Moneyless refund on ADP.	

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 query execution success. If errors array is empty, field value will be set to true	
data			Returned data	
	passengerRefundDetails		Array of objects with refund details information for each passenger.	
		hasInfant	Flag identifying whether passenger travels with infant	
		passengerType	Passenger type Possible values: 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	Number of refunded	
		refundTotal	Total refund sum, 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 a user.	
				taxCategory	Tax category For refundable taxes "R" is returned, for non-refundable taxes - "NRT".	
				taxDetails	Array of objects with information about tariffs and fees details.	
				rate	Tariff/fee sum	
				currencyCode	Currency code	
				type	Tariff/fee type	
				<i>luggageSeatsRefundDetails</i>	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 sum, 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	
		<i>refundWithTicketOnlyServices</i>	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	<p>Array of objects, containing errors from Sirena response.</p> <p>Array can be empty, it is recommended to analyse errors by ibeCode.</p>	
	code	Sirena error code	
	text	Sirena error text	

POST refund/partial method

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

Request format: /api/apiVersion/refund/partial

Request parameters:

Parameter	Mandadtory	Description	Comments
pnrNumber	yes	PNR number, unique 6-digit booking identifier	
passengers	yes	An array of passengers for which refund sums needs to be calculated	
isTicketRefund	yes	The sign that a ticket with services or only services is being returned	
partnerProducts	no	An array of partner services to be returned (the array parameters are similar to the GET api/Reservation response parameters)	
lastName	yes	Passenger's last name	
passengerReferenceNumber	yes	<p>Passenger reference.</p> <p>Associated with field data -> passengers -> referenceNumber</p>	

		<i>luggageSeats</i>	no	Array with information about luggage with separate air ticket and ssr CBBG.	In developing
		passengerType	yes	Passenger type. Possible types of passengers are presented in EN_8_1_Attachment: Dictionaries of passenger types.	
		relatedAdultReferenceNumber	no	Reference to the adult passenger with whom the child or infant is flying. The parameter is filled for the child and infant	
		relatedInfantReferenceNumber	no	Reference to the infant with whom the adult is flying. The parameter is filled for the adult who flies with infant.	
		services	no	An array with information about purchased and refundable additional services of the airline	
		code	yes	Returned additional service code	
		ticketNumber	yes	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 query execution success. If errors array is empty, field value will be set to true	
data	Returned data	
passengerRefundDetails	Array with refund information Full analogue of data in response of api/refund/calculate method	
crmRefundRequestId	Refund request number in CRM. Example: 1-46MJI9D.	
<i>refundWithTicketOnlyServices</i>	Array with service codes that can't be refunded without refunding the ticket.	In developing
errors	Information about errors appeared during query execution.	

Requests 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": "1ZROML",
  "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 is initiated by airline's client own will.

Short description of voluntary refund business process.

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

General technical information.

1. For each refund (initiated by api/refund method call) a record with unique refund session ID is created in DB.
2. 3 entities can be linked with refund session: a) tickets refund information entity, b) airline's services refund information entity c) partner's services refund information entity

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

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

These methods are described in GOI documentation.

Failed request and response examples

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

<p>Lets consider that in 1ZGLLO booking adult passenger (of type 0) is added and correct reference to passenger - 12.</p> <p>An error will appear in case of:</p> <ol style="list-style-type: none"> 1. Incorrect reference to passenger is passed. 2. Incorrect type of passenger is passed. 	<p>Request for point 1.</p> <pre>{ "pnrNumber": "1ZGLLO", "lastName": "TERENTEVA", "passengers": [{ "lastName": "TERENTEVA", "passengerType": 0, "passengerReferenceNumber": "14", "isTicketRefund": true, "luggageSeats": [], "services": [], "partnerProducts": [] }] }</pre> <p>Request for point 2.</p> <pre>{ "pnrNumber": "1ZGLLO", "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" }], "success": false }] }</pre>
---	---	--

<p>Booking for 1 adult and 1 child</p> <p>It is not possible to perform refund only for adult passenger.</p>	<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 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 query execution success. If errors array is empty, field value will be set to 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
}

```

Authorize in CRM: api/user (unavailable for agents)

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

Method POST user

Description: authorisation in CRM

Request format /api/apiVersion/user

Request parameters

Parameter	Description
login	User login
password	User password
token	Auth token

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

Request examples

Authorisation with login and password

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

Authorisation with token

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

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

2. Authorize with login and token

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

Response format

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

```

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

```

```

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

```

Response parameters

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

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

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

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

Card

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

Booking payment service (unavailable for agents)

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

POST payment method

Description: forms link for redirection to booking payment form

Request format /api/apiVersion/payment

Request body

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

Request parameters

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

Response format

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

Response parameters

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

Successful requests and response examples

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

POST payment/email method

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

Request format /api/apiVersion/payment/email

Request body

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

Request parameters

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

Response format

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

```

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

```

Response parameters

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

Successful requests and response examples

```

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

```

GET payment method

Description: allows to check booking payment status

Request format /api/apiVersion/payment

Query sample:

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

Request parameter

Parameter	Description
orderId	Checking order number

Response format

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

Response parameters

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

		text	Error text
--	--	------	------------

Successful requests and response examples

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

Failed requests and response examples

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

GET payment/info method

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

Request format /api/apiVersion/Payment/Info

Query sample:

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

Request parameter

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

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

Response format

```
{
  "data": {
    "orderId": string,
    "errorCode": string,
    "mobileErrorCode": string,
    "status": string,
    "pnr": string,
    "currency": string,
    "remainMilisecs": decimal,
    "amount": decimal,
    "language": string
  },
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string,
      "additionalInfo": {
        "ssrType": string,
        "info": string
      }
    }]
  }]
},
"success": boolean
}
```

Response parameters

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

	errorCode	Error code that returns from BPC. If the request finishes without errors then the value is "OK" (it is the same with mobileErrorCode).
	mobileErrorCode	Error code that returns from BPC. If the request finishes without errors then the value is "OK".
	status	The status that returns from BPC.
	pnr	Code of reservation.
	currency	Three-letter currency code.
	remainMilisecs	The remaining time in milliseconds during which payment will be available.
	amount	The amount to be paid in the minimum currency units.
	language	Two-letter language code.
errors		Information about errors appeared during query execution.
	ibeCode	IBE error code.
	description	IBE error description.
	messages	Array of objects, containing conterminous system's errors. Array might be empty. It is recommended to analyze ibeCode field.
	code	Error code.
	text	Error text.
	additionalInfo	Additional information about errors.
	ssrType	Ssr type.
	info	Information.

Successful requests and response examples

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

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

Failed requests and response examples

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

Online check-in service

API service	Checkin
Short description	Allows to check online check-in for your flight from the booking management module and, if available, go to registration service
Supported HTTP-(S) methods	GET

GET Checkin method

Description: allows to get data on the possibility of registration.

Return the time of start and end of registration, remaining time until start or end of registration.

Request format /api/apiVersion/checkin

Request parameters

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

Response format

```
{
  "data": {
    "segments": [
      {
        "canRegister": bool,
        "flightReference": string,
        "webCheckinEndDate": datetime,
        "webCheckinEndDateUtc": datetime,
        "webCheckinStartDate": datetime,
        "webCheckinStartDateUtc": datetime,
        "webCheckinEndRemainingTime": integer,
        "webCheckinStartRemainingTime": integer
      }
    ],
    "passengers": [
      {
        "passengerReference": strgin,
        "flightReference": string,
        "ticketNumber": string,

```

```

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

```

Response parameters

Parameter		Description
success		Sign of the success of the request. If the errors array is empty, then the field value will be true
data		Online check-in information
	segments	Array of flight segments
	canRegister	Sign of registration availability
	flightReference	Flight reference
	webCheckinEndDate	Local date of checkin end
	webCheckinEndDateUtc	UTC date of checkin end
	webCheckinStartDate	Local date of checkin start
	webCheckinStartDateUtc	UTC date of checkin start
	webCheckinEndRemainingTime	Remaining time until end of registration
	webCheckinStartRemainingTime	Remaining time until start of registration
passengers		Passenger list
	passengerReference	Passenger reference
	flightReference	Flight reference

	ticketNumber	Passenger ticket number
	registered	Sign of registration
errors		Information about errors that occurred during the operation of the service
	ibeCode	Error code, see Attachment 1 IBE error codes
	description	Error description, see Attachment 1 IBE error codes
	messages	An array of objects containing errors from Sirena answer. The array may be empty, it is recommended to analyze the error by field ibeCode.
	code	Sirena error code
	text	Sirena error text

Examples of successful responses

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

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

Response when registration is not open yet:

```
{
```

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

Change passenger's data service

API service	passenger
Short description	Allows to change passenger's data in booking
Supported HTTP(S)-methods	PUT
Conterminous systems methods used	Sirena: modify

PUT passenger method

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

Implemented logic:

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

Request format: /api/apiVersion/passenger

Request parameters

Parameter	Mandatory	Description
referenceNumber	yes	Reference to passenger
passengerType	yes	Passenger's type
surname	no	Last name
firstName	no	First name
middleName	no	Middle name
email	no	Email
phone	no	Phone
nationality	no	Nationality
sex	no	Sex
birthDate	no	Date of birth
documentNumber	no	Document number
callbackUrl	no	Url for retrieve payment status
engine	no	engine for BMM

lang	no	<p>Interface language for the BPC payment form.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ru - Russian; • en - English; • cn - Chinese.
redirectLang	no	<p>Redirection language to BMM after payment.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • ru - Russian; • en - English; • zh - Chinese; • uz - Uzbek; • ky - Kyrgyz; • fr - French.
returnUrl	no	Return address after payment.

Response format

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

Response parameters

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

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

Request and response examples

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

Valid request	<p>Passenger in booking</p> <pre>{ "id": 12, "hasInfant": false, "referenceNumber": "12", "firstName": "ANTON", "middleName": "ANDREEVICH", "surname": "SAVIN", "passengerType": 0, "birthDate": "1999-10- 25T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "4444111222", "documentExpiryDate": "2049-12-31T00:00:00", "email": "SOBAKA010101@MAIL.RU", "phone": "79531568787", "documentType": 0, "isAdult": true, "isChanged": false }</pre> <p>Request</p> <pre>{ "referenceNumber": "12", "passengerType": 0, "documentNumber": "4444111222", "email": "sobaka010101@mail.ru", "phone": "79531568786", "sex": 0, "birthDate": "1999-10- 25T00:00:00Z", "lang": "ru", "redirectLang": "ru" }</pre>	<pre>{ "data": { "orderId": "0c42e8e6-5f91- 4e04-b4d6-347d840014fa", "paymentFormURL": "https://web.rbsuat.com /amadeuspsp/static/payment /payment.html ?orderId=0c42e8e6- 5f91-4e04-b4d6-347d840014fa &systemId=mainwebst &language=ru &hash=661a02522e176c52d97d7e 6436a44b3115ea8ealbfc9ca3 e10788780d1feaf09"}, "errors": [], "success": true}</pre>
---------------	--	--

Contact info add service Reservation/AddContactInfo (for ADP environment)

API service	Reservation/AddContactInfo
Short description	Allow to add contact info to a passenger 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 parameters

Parameter	Description
reservationNumber	PNR of reservation
surname	Passenger's surname
contactInfo	Array with contact info
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 query execution success. If errors array is empty, field value will be set to true
errors		Information about errors appeared during query execution.
	ibeCode	Error code, see EN_Attachment 1. IBE error codes
	description	Code description, see. EN_Attachment 1. IBE error codes
	messages	Array of objects, containing errors from Sirena response. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Sirena error code
	text	Sirena error text

Service for retrieving SSR and remarks information /api/version/Reservation/Advanced (for ADP environment)

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

POST AddSSR method

Note: method is available on ADP

Description: adds SSR to reservation

Request format: api/apiVersion/Reservation/Advanced

Request header

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

Request paramaters description

Параметр	Описание
PnrNumber	Reservation number.
TicketNumber	Ticket number, if one is used instead of PNR number, to retrieve reservation.
LastName	Passenger last name.
LastNameHash	Passenger last name hash.

Response parameters

```
{
  "data": {
    "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",
        "issueDate": "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"
},
"tsmData": [
    {
        "passengerReference": "string",
        "price": "integer",
        "referenceType": "string",
        "referenceNumber": "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",
        "outboundPrice": "integer",
        "returnPrice": "integer",
        "discountAmount": "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",
"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"
}
],
"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"
    }
],
"outboundFareFamily": {
    "code": "string",
    "category": "string",
    "isSubsidized": "bool",
    "isLowCost": "bool",
    "isUnaccompaniedMinor": "bool"
},
"returnFareFamily": {
    "code": "string",
    "category": "string",
    "isSubsidized": "bool",
    "isLowCost": "bool",
    "isUnaccompaniedMinor": "bool"
},
"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": "bool"
},
"number": "string",
"reservationDate": "datetime",
"bookingTimeLimit": "datetime",
"passengers": [
    {
        "id": "integer",
        "hasInfant": "bool",
        "referenceNumber": "string",
        "relatedAdultReferenceNumber": "string",
        "relatedInfantReferenceNumber": "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"
  }
]
},
],
"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"
  }
],
"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"
  }
],
"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"
  }
],
"partnerProducts": [
  {
    "partnerId": "string",
    "partnerType": "string",
    "bookingStatus": "string",
    "code": "string",
    "amount": "integer",
    "currency": "string",
    "passengerReference": "string",
    "flightReference": "string",
    "isInfantProduct": "bool",
    "bookingId": "string",
    "orderId": "string",
    "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 block

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

Remarks block

Parameter		Description
remarks		Returned data
	type	Remark type (tag <type>)
	text	Content of tag freeText
	referenceNumber	Reference to a service
	passengerReferences	Reference to a passenger (if exists)
	flightReferences	Reference to a segment (is exist)

specialServiceRequests block

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)

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

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

POST AddSSR methodx

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

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

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	Reference number
segmentReference	Flight segment reference
freeText	Text

Формат ответа

```
{
  "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 query execution success. If errors array is empty, field value will be set to 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 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 analyse errors by ibeCode
	code	Sirena error code
	text	Sirena error text

Service of tickets rebook /api/Exchange (for ADP environment)

API service	/Exchange
Short description	Service of tickets exchange
Supported HTTP(S)-methods	POST
Conterminous systems methods used	Sirena: exchange_pricing, divide_order, exchange_confirm

POST Exchange method

Note: method is available on ADP

Description: allow to make a rebook of ticket

Request format: api/apiVersion/Exchange

Request headers

X-Api-Key - unique GUID for systems using IBE API.

Request body

```

{
  "pnrNumber": string,
  "lastName": string,
  "travelParts": [
    {
      "segmentReference": string,
      "segmentReferenceLast": string,
      "fareFamily": {
        "code": string,
        "category": string,
        "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 a 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

Parameter		Description				
	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	Number of flight				
	classOfService	Information about selected tariff				
	fares	Tariff information object				
	fareBasis	Base fare name				
	isBreakpoint	Flag identifying whether tariff ends on present segment				
	paxType	Passenger type				
	passengerReference	Reference to the passenger whose ticket to be rebooked				

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": [  
  {  
    "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"
```

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

}

```

Объект fullReservationFlight

```

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

```

Response parameters

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

Parameter			Description
	partnerProducts		An array of objects with information about partner services.
	journey		Flight information for current booking
		outboundFlights	An array of the fullReservationFlight objects with details of outbound flights
		outboundFareFamily	Service field, contains the fareFamily code
		returnFlights	An array of fullReservationFlight objects with detailed information about the "return" flights
		returnFareFlights	
		outboundDuration	
		returnDuration	
		origin	Initial airport of departure
		destination	Initial airport of destination
		departureDate	Departure date
		returnDate	Return date
		outboundBaggageAllowances	
		returnBaggageAllowances	
		containsInternationalFlights	
	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"

Parameter			Description
		passengerType	Passenger type Possible values: see Attachment. Passenger types.
		referenceNumber	Reference number
		referenceQualifier	Link type, for example "TST"
		totalAmount	The total amount of the tariff
		currency	Reservation currency
fareBasisCodesFlightRefs			
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. 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

Parameter		Description
	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

Parameter	Description
aircraftIataCode	IATA aircraft code
referenceNumber	Reference to the current flight segment
origin	Departure airport for the current segment
destination	Destination airport for the current segment
departureDate	Departure date and time for the current segment
arrivalDate	Arrival date and time for the current segment
arrivalDateUtc	Date and time of arrival for the current segment in UTC
departureDateUtc	Departure date and time for the current segment in UTC
flightNumber	Flight number
classOfService	Information about the selected tariff

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

passengerData object

Parameter	Description
id	Passenger's serial number (numbered from 1)
hasInfant	An sign that this passenger has a baby
referenceNumber	Reference to this passenger in the booking
firstName	Passenger's name
middleName	Passenger's patronymic (may be absent)
surname	Passenger's surname
passengerType	Passenger type Possible values: see Attachment 2. Passenger types

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

Certificate management service (unavailable for agents)

API service	certificate
Short description	The service allows to receive information about certificates and delete them for the current booking
Supported HTTP(S)-methods	GET, DELETE, POST
Methods of related systems used	Sirena: emd_display, svc_add

GET api/certificate method

Description: Obtaining information about a certificate

Request format /api/apiVersion/certificate

Request parameters: X-Session is passed in the header

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

Response format

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

```

    }
  ],
  "success": boolean
}

```

Response parameters

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

DELETE api/certificate method

Description: allows to remove the applied certificate.

Request format /api/apiVersion/certificate

Request parameters

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

Response format

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

Response parameters

Parameter		Description
success		Sign of the success of the request. If the errors array is empty, then the field value will be true
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
	additionalInfo	Additional information
	ssrType	SSR type

Parameter	Description
info	Additional information text

GET api/certificate/CheckCertificate method

Description: Obtaining information about a certificate

Request format /api/apiVersion/certificate/CheckCertificate

Request parameters:

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

Response format

```
{
  "data": {
    "documents": [
      {
        "number": string,
        "surname": string,
        "status": string,
        "notValidAfter": "datetime,
        "amount": integer,
        "currency": string
      }
    ]
  },
  "errors": [
    {
      "ibeCode": integer,
      "description": string,
      "messages": [
        {
          "code": string,
          "text": string,
          "additionalInfo": {
            "ssrType": string,
            "info": string
          }
        }
      ]
    }
  ]
},
],
```

```

    "success": boolean
}

```

Response parameters

Parameter		Description
data		The data returned by the request
	documents	Certificate information
	number	Certificate number
	surname	Surname of the owner of the certificate
	status	Certificate status (open - open, close - closed)
	notValidAfter	Date until which the certificate is valid
	amount	Certificate amount
	currency	Currency (RUB)
errors		Information about errors that occurred during the operation of the service.
	ibeCode	Error code, see Attachment 1. IBE error codes
	description	Error description, see Attachment 1. IBE error codes
	messages	An array of objects containing errors from 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	Additional information
	ssrType	SSR type
	info	Additional information text
success		Sign of the success of the request. If the errors array is empty, then the field value will be true

POST api/certificate method

Description: Creating new certificate.

Request format /api/apiVersion/certificate

Request format

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

Request parameters:

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

Response format

```
{
  "data": {
    "pnr": string,
    "surname": string,
    "amount": decimal,
    "currency": string,
    "notValidAfter": datetime,
  }
}
```

```

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

```

Response parameters

Parameter		Description
data		The data returned by the request
	pnr	Unique booking number
	surname	Last name of the certificate holder.
	amount	Certificate amount.
	currency	Certificate currency.
	notValidAfter	Date until which the certificate is valid.
	isGiftCertificate	Indicates whether the certificate is a gift or not.
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	<p>An array of objects containing errors from Sirena answer.</p> <p>The array may be empty, it is recommended to analyze the error by the ibeCode field.</p>
		code	Sirena error code.
		text	Sirena error text.
		additionalInfo	Additional information
		ssrType	SSR type
		info	Additional information text
		success	<p>Sign of the success of the request.</p> <p>If the errors array is empty, then the field value will be true.</p>

POST api/certificate/apply method

Description: Gets a certificate data and adds it to context.

Request format /api/apiVersion/certificate/apply

Request parameters

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

Response format

```
{
  "data": {
    "number": string,
    "surname": string,
    "amount": decimal,
    "notValidAfter": datetime,
    "isGiftCertificate": boolean
  }
  "errors"
  {
    "ibeCode": integer,
    "description": string,
    "messages": [
```

```

{
    "code": integer,
    "text": string,
    "additionalInfo": {
        "ssrType": string,
        "info": string
    }
}

]
}
]
"success": boolean
}

```

Response parameters

Parameter			Description
data			The data returned by the request
	surname		Last name of the certificate holder.
	amount		Certificate amount.
	currency		Certificate currency.
	notValidAfter		Date until which the certificate is valid.
	isGiftCertificate		Indicates whether the certificate is a gift or not.
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.
		additionalInfo	Additional information
		ssrType	SSR type

			info	Additional information text
success				Sign of the success of the request. If the errors array is empty, then the field value will be true.

Service API methods description

Retrieve airports list service (unavailable for agents)

API service	airports (service)
Short description	Allows to retrieve list of airports to which U6 company performs flights
Supported HTTP(S)-methods	GET

GET airports methods

Description: returns list of airports and its iata codes.

Request format /api/apiVersion/airports

Request parameters

No parameters.

Response format

```
{
  "data": [{
    "country": string,
    "cityRu": string,
    "cityEn": string,
    "iataCity": string,
    "airportRu": string,
    "airportEn": string,
    "iataAirport": string
  }
],
  "errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }
  ]
},
  "success": boolean
}
```

Response parameters

Parameter		Description
success		Sign of query execution success. If errors array is empty, field value will be set to true
data		Returned data
	country	Country where city is located
	cityRu	City name (russian locale)
	cityEn	City name (english locale)
	iataCity	IATA-code of the city
	airportRu	Airport name (russian locale)
	airportEn	Airport name (english locale)
	iataAirport	IATA-code of the airport
errors		Information about errors appeared during query execution.
	ibeCode	Error code
	description	Code description
	messages	Array of objects, containing errors. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Error code
	text	Error text

Send error report service (unavailable for agents)

API service	problemNotification (service)
Short description	Allows users to send error reports
Supported HTTP(S)-methods	POST

POST problemNotification method

Description: allows users to report about problems. Report sending form is available in frontend when "general technical error" gets received with 1000 code.

Request format /api/apiVersion/problemNotification

Request body

```
{
  "from": "string",
  "description": "string",
  "details": "string"
}
```

Request parameters

Parameter	Description
from	Sender. Any string identifying user.
description	Description of the problem.
details	Technical details of the problem. For example, session id.

Response format

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


Response parameters

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

Request and response examples

Successful sending of information about error

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

Get exchange rates service (unavailable for agents)

API service	exchangerates (service)
Short description	Allows to receive information about currency exchange rates
Supported HTTP(S) methods	GET

GET exchangerates method

Description: retrieves information about currency exchange rates

Request format /api/apiVersion/exchangerates

Request parameters

No parameters

Response format

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

Response parameters

Parameter	Description
success	Sign of query execution success. If errors array is empty, field value will be set to true
data	Returned data
name	Currency name, for instance, «Azerbaijan Manat»

	code	Currency code, for instance, «AZN»
	value	current exchange rate to RUR
errors		Information about errors appeared during query execution.
	ibeCode	Error code.
	description	Code description.
	messages	Array of objects, containing errors. Array can be empty, it is recommended to analyse errors by ibeCode.
	code	Error code
	text	Error text

Response example

```
{
  "data": [{
    "name": "Azerbaijan Manat",
    "code": "AZN",
    "value": 39.857316
  }, {
    "name": "Armenia Dram",
    "code": "AMD",
    "value": 0.13939014
  }, {
    "name": "US Dollar",
    "code": "USD",
    "value": 67.617942
  }, {
    "name": "Euro",
    "code": "EUR",
    "value": 76.895148
  }, {
    "name": "Indian Rupee",
    "code": "INR",
    "value": 0.9637674
  }, {
    "name": "Kazakhstan Tenge",
    "code": "KZT",
    "value": 0.18109692
  }, {
    "name": "Kyrgyzstan Som",
    "code": "KGS",
    "value": 0.96873888
  }, {
    "name": "China Yuan",
    "code": "CNY",
    "value": 9.8162046
  }, {
```

```
    "name": "Tajikistan Ruble",
    "code": "TJS",
    "value": 7.1774442
  }, {
    "name": "Uzbekistan Sum",
    "code": "UZS",
    "value": 0.0081475458
  }, {
    "name": "Czech Koruna",
    "code": "CZK",
    "value": 2.9621310
  }, {
    "name": "Japanese Yen",
    "code": "JPY",
    "value": 0.59577894
  }
],
"errors": [],
"success": true
}
```

Get Cyrillic transliteration rules service (unavailable for agents)

API service	settings/transliteration (service)
Short description	Allows to receive Cyrillic transliteration rules
Supportes HTTP(S) methods	GET

GET settings/transliteration method

Description: requests Cyrillic transliteration rules

Request format /api/apiVersion/settings/transliteration

Request parameters

No parameters.

Response format

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

Response parameters

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

success	Sign of query execution success. If errors array is empty, field value will be set to true	
data	Returned data	
	nativeLetter	Cyrillic symbol
	latinLetter	Latin symbol
errors	Information about errors appeared during query execution.	
	ibeCode	Error code.
	description	Code description.
	messages	Array of objects, containing errors. Array can be empty, it is recommended to analyse errors by ibeCode.
		code
		text

Response example

```
{
  "data": [
    {
      "nativeLetter": "А",
      "latinLetter": "A"
    },
    {
      "nativeLetter": "а",
      "latinLetter": "a"
    },
    {
      "nativeLetter": "Б",
      "latinLetter": "B"
    },
    {
      "nativeLetter": "б",
      "latinLetter": "b"
    },
    {
      "nativeLetter": "Я",
      "latinLetter": "Ia"
    },
    {
      "nativeLetter": "я",
      "latinLetter": "ia"
    }
  ],
  "errors": []
}
```

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

Attachments

Attachment: Dictionaries

Passenger types

paxType field value	Decryption	Passenger type
0	ADT, adult	Conventional
1	SRC, pensioner	Subsidised
2	YTH, youth	Subsidised
3	CHD, child	Conventional
4	INF, infant	Conventional
5	CNN, child	Subsidised/conventional
6	UMNR, child	Unaccompanied child
7	STR, adult	Subsidised for Far East Region
8	JNN, child	Subsidised for Far East Region
13*	INS, infant	Infant with a seat (for now this type is cosidered as infant when adding a seat, in developing)
17*	MMI, adult	Military

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

Passenger sex

sex field value	Decryption
0	Male
1	Female

Document types

documentType field value	Decryption
---------------------------------	-------------------

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

Passenger assistance types

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

Wheelchair types

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

Auxiliary services types (a part of services are in development)

services.code field value	Category	Decryption
ABAG	luggage	First additional bag
BBAG	luggage	Second additional bag
CBAG	luggage	Third additional bag
DBAG	luggage	Additional bag 10 kg
HBAG	luggage	Extension of additional bag or default luggage (if allowed)
CHML	meal	Child meal
HTML	meal	Hot meal
MOML	meal	Muslim halal meal

SNML	meal	Sandwich meal
SFML	meal	Hot seafood
VGML	meal	Vegeterian meal
CAML	meal	Dessert
KSML	meal	Kosher
BGML	meal	Burger
AVML	meal	Vegetarian Hindu
HNML	meal	Indian Cuisine
VJML	meal	Vegetarian jain
TCML	meal	Hot breakfast
GFML	meal	Gluten free
BBML	meal	Baby meal
CPML	meal	Minced steak
MPML	meal	Salad "fitness"
GRML	meal	Greek salad
MTML	meal	Roll with chicken
UDML	meal	Poke with chicken
FTML	meal	Fish Steak Meal
ECML	meal	English Breakfast
FCML	meal	French Breakfast
ACML	meal	American Breakfast
RCML	meal	Russian Breakfast
CSML	meal	Caucasian Cuisine
RUML	meal	Russian Cuisine
ORML	meal	Asian Cuisine
MEML	meal	Italian Cuisine
GEML	meal	German Cuisine
ASML	meal	Central Asian Cuisine
PETC	pets	Pet transportation

URFL	insurance	Insurance, avia bundle
URTR	insurance	Insurance, travel bundle
SEAT	seat	Seat selection
BLOU	room	Business room
VLOU	room	VIP room
CVID	covidExpress	Rapid analysis for covid-19
XBLL	handLuggage	Hand luggage up to 10 kg in the cabin (luggage in the cabin)
XBLS	handLuggage	Additional piece of carry-on baggage up to 10 kg in the cabin
NYML	meal	New Year meal
CCML	meal	Chocolate Honey Cake
CFML	meal	Assorted macaroons
CTML	meal	Coffee and dessert
ROML	meal	HAM and CHEESE ROLL

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

Значение поля serviceCategories.code	Категория	Расшифровка
CBBG	luggage	Luggage in the aircraft cabin with a separate seat
BIKE	luggage	Bicycle in the luggage compartment
SKI	luggage	Ski equipment
BULK	luggage	Additional carry-on baggage
XBAG	luggage	Oversized baggage
HBGS	luggage	Excess baggage
HBGT	luggage	Excess baggage
DBGS	luggage	Increase in baggage weight up to 32 kg
DBGT	luggage	Increase in baggage weight up to 32 kg
FRAG	luggage	Fragile luggage

Значение поля serviceCategories.code	Категория	Расшифровка
WBAG	luggage	registration of payment for excess weight of checked baggage from 10 to 23 kg
SRXBLL	luggage	registration of payment for excess weight of hand luggage from 5 to 10 kg, not exceeding dimensions 55*40*20
SRXBLS	luggage	registration of payment for an additional piece of hand luggage weighing no more than 5 kg and not exceeding dimensions 55*40*20
MEQT	luggage	Medical equipment
WEAP	luggage	Weapon
GOLF	luggage	Golf equipment
SPEQ	luggage	Sport equipment
RVML	meal	Raw vegetarian meal
VLML	meal	Vegetarian lacto-ovo meal
VOML	meal	Vegetarian oriental meal
DBML	meal	Diabetic food
BLML	meal	Diet food
LCML	meal	Low-calorie food
FPML	meal	Fruit nutrition
LFML	meal	Low-fat meals
NLML	meal	Low lactose meals
LSML	meal	Low salt diet
NFML	meal	Fish-free meals
SPML	meal	Special meals
JPML	meal	Japanese food
AVIH	pets	Animal in the hold
SVAN	pets	Animal in the cabin
EXST	seat	Additional free space nearby
STRQ	seat	SEAT REQUEST, Seat

Значение поля serviceCategories.code	Категория	Расшифровка
GPST	seat	Group seat request, specify number and preferred location, Seat
RQST	seat	Seat request-include seat number or preference, Seat
CHNM	data	CHANGE NAME, Name change
NAMM	data	Payment of the fee for changing the name of the passenger on the ticket
NAME	data	Name - when airline holds reservation under a different name, Passanger's name change
PFEE	data	Payment of the fee for renewal at the call center
CKIN	data	Registration service
BLTW	data	Blocking the nearest seat
UMNR	other	Unaccompanied child
BUSS	other	Transfer by bus
TIME	other	Deferred payment service
WCHR	other	Wheelchair to/from the plane
WCHS	other	Wheelchair to climb/descend the airplane ladder
WCHC	other	Wheelchair for movement on the apron and the cabin of the aircraft
URFO	insurance	Avia
NSP1	insurance	Cancellation of a trip with insurance coverage of RUB 5,000
NSP2	insurance	Cancellation of a trip with insurance coverage of RUB 10000
NSP3	insurance	Cancellation of a trip with insurance coverage of RUB 20000
NSP4	insurance	Cancellation of a trip with insurance coverage of RUB 30000
NSP5	insurance	Cancellation of a trip with insurance coverage of RUB 40000

Types of animals

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

Most widely spread Seatmap seat characteristics (in development)

characteristics field value	Decryption
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/compartiment
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)

Services statuses

Статус	Описание
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	Decryption
0	New
1	Used
2	Sent
3	Saved
4	Received
5	Hold

Tariff options

Option code	Category	Decryption
bestSeats	availablePaid	first rows in cabin of economy class (up to 15th row) if available at the moment of registration
upTo5bonuses	bonus	Bonus Accrual up to 5%
upTo10bonuses	bonus	Bonus Accrual up to 10%
upTo15bonuses	bonus	Bonus Accrual up to 15%
flexibilityOnChangesAndCancellations	ffServices	Possibility of ticket change and cancellation
refundExchangeProhibited	ffServices	Ticket refund is prohibited
handLuggage5kg	handBag	Hand luggage 5 kg
handLuggage15kg	handBag	Hand luggage 15 kg
oneSuitcase10kg	luggage	One suitcase 10 kg
oneSuitcase23kg	luggage	One suitcase 23 kg
oneSuitcase32kg	luggage	One suitcase 32 kg
twoSuitcase32kg	luggage	Two suitcase 32 kg
mealsOnBoard	menu	Standard meal
boarding	priorityBoarding	Priority boarding
checkin	priorityCheckIn	Priority checkin
businessClassCatering	specialMenu	Business class catering
upgradeToBusinessClass	upgradeToBusiness	Upgrade to business class
vipLounges	vipLounge	Vip lounge
twoSeatsOnRow	vipSeats	Two seats on row
vipSeats	vipSeats	Vip seats
noMealsOnBoard	menu	No meals

Main fare family categories

Category	Decryption
----------	------------

PR	«Promo» tariff
EC	«Econom» tariff
FL	«Premium-Econom» tariff
BS	«Business» tariff
CF	«Comfort» tariff
PL	«Promo-light» tariff
LC1	«Bagless» tariff
LC2	«Luggage» tariff
SOC	«Subsidised» tariff

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
COK	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
CYM	Острова Кайман	Cayman Islands
CYP	Кипр	Cyprus
CZE	Чехия	Czech Republic
DEU	Германия	Germany
DJI	Джибути	Djibouti
DMA	Доминика	Dominica
DNK	Дания	Denmark
DOM	Доминиканская Республика	Dominican Republic
DZA	Алжир	Algeria
ECU	Эквадор	Ecuador
EGY	Египет	Egypt
ERI	Эритрея	Eritrea
ESH	САДР	Western Sahara
ESP	Испания	Spain
EST	Эстония	Estonia
ETH	Эфиопия	Ethiopia
FIN	Финляндия	Finland

FJI	Фиджи	Fiji
FLK	Фолклендские острова	Falkland Islands (Malvinas)
FRA	Франция	France
FRO	Фареры	Faroe Islands
FSM	Микронезия	Micronesia Federated States of
GAB	Габон	Gabon
GBR	Великобритания	United Kingdom
GEO	Грузия	Georgia
GGY	Гернси	Guernsey
GHA	Гана	Ghana
GIB	Гибралтар	Gibraltar
GIN	Гвинея	Guinea
GLP	Гваделупа	Guadeloupe
GMB	Гамбия	Gambia
GNB	Гвинея-Бисау	Guinea-Bissau
GNQ	Экваториальная Гвинея	Equatorial Guinea
GRC	Греция	Greece
GRD	Гренада	Grenada
GRL	Гренландия	Greenland
GTM	Гватемала	Guatemala
GUF	Гвиана	French Guiana
GUM	Гуам	Guam
GUY	Гайана	Guyana
HKG	Гонконг	Hong Kong
HMD	Херд и Макдональд	Heard Island and McDonald Islands
HND	Гондурас	Honduras
HRV	Хорватия	Croatia
HTI	Гаити	Haiti

HUN	Венгрия	Hungary
IDN	Индонезия	Indonesia
IMN	Остров Мэн	Isle of Man
IND	Индия	India
IOT	Британская территория в Индийском океане	British Indian Ocean Territory
IRL	Ирландия	Ireland
IRN	Иран	Iran Islamic Republic of
IRQ	Ирак	Iraq
ISL	Исландия	Iceland
ISR	Израиль	Israel
ITA	Италия	Italy
JAM	Ямайка	Jamaica
JEY	Джерси	Jersey
JOR	Иордания	Jordan
JPN	Япония	Japan
KAZ	Казахстан	Kazakhstan
KEN	Кения	Kenya
KGZ	Киргизия	Kyrgyzstan
KHM	Камбоджа	Cambodia
KIR	Кирибати	Kiribati
KNA	Сент-Китс и Невис	Saint Kitts and Nevis
KOR	Республика Корея	Korea Republic of
KWT	Кувейт	Kuwait
LAO	Лаос	Lao People's Democratic Republic
LBN	Ливан	Lebanon
LBR	Либерия	Liberia
LBY	Ливия	Libyan Arab Jamahiriya
LCA	Сент-Люсия	Saint Lucia

LIE	Лихтенштейн	Liechtenstein
LKA	Шри-Ланка	Sri Lanka
LSO	Лесото	Lesotho
LTU	Литва	Lithuania
LUX	Люксембург	Luxembourg
LVA	Латвия	Latvia
MAC	Макао	Macao
MAF	Сен-Мартен	Saint Martin (French part)
MAR	Марокко	Morocco
MCO	Монако	Monaco
MDA	Молдавия	Moldova Republic of
MDG	Мадагаскар	Madagascar
MDV	Мальдивы	Maldives
MEX	Мексика	Mexico
MHL	Маршалловы Острова	Marshall Islands
MKD	Македония	Macedonia the Former Yugoslav Republic of
MLI	Мали	Mali
MLT	Мальта	Malta
MMR	Мьянма	Myanmar
MNE	Черногория	Montenegro
MNG	Монголия	Mongolia
MNP	Северные Марианские Острова	Northern Mariana Islands
MOZ	Мозамбик	Mozambique
MRT	Мавритания	Mauritania
MSR	Монтсеррат	Montserrat
MTQ	Мартиника	Martinique
MUS	Маврикий	Mauritius
MWI	Малави	Malawi

MYS	Малайзия	Malaysia
MYT	Майотта	Mayotte
NAM	Намибия	Namibia
NCL	Новая Каледония	New Caledonia
NER	Нигер	Niger
NFK	Остров Норфолк	Norfolk Island
NGA	Нигерия	Nigeria
NIC	Никарагуа	Nicaragua
NIU	Ниуэ	Niue
NLD	Нидерланды	Netherlands
NOR	Норвегия	Norway
NPL	Непал	Nepal
NRU	Науру	Nauru
NZL	Новая Зеландия	New Zealand
OMN	Оман	Oman
PAK	Пакистан	Pakistan
PAN	Панама	Panama
PCN	Острова Питкэрн	Pitcairn
PER	Перу	Peru
PHL	Филиппины	Philippines
PLW	Палау	Palau
PNG	Папуа — Новая Гвинея	Papua New Guinea
POL	Польша	Poland
PRI	Пуэрто-Рико	Puerto Rico
PRK	КНДР (Корейская Народно-Демократическая Республика)	Korea Democratic People's Republic of
PRT	Португалия	Portugal
PRY	Парагвай	Paraguay
PSE	Государство Палестина	Palestinian Territory Occupied

PYF	Французская Полинезия	French Polynesia
QAT	Катар	Qatar
REU	Реюньон	Reunion
ROU	Румыния	Romania
RUS	Россия	Russian Federation
RWA	Руанда	Rwanda
SAU	Саудовская Аравия	Saudi Arabia
SDN	Судан	Sudan
SEN	Сенегал	Senegal
SGP	Сингапур	Singapore
SGS	Южная Георгия и Южные Сандвичевы Острова	South Georgia and the South Sandwich Islands
SHN	Острова Святой Елены Вознесения и Тристан-да-Кунья	Saint Helena
SJM	Шпицберген и Ян-Майен	Svalbard and Jan Mayen
SLB	Соломоновы Острова	Solomon Islands
SLE	Сьерра-Леоне	Sierra Leone
SLV	Сальвадор	El Salvador
SMR	Сан-Марино	San Marino
SOM	Сомали	Somalia
SPM	Сен-Пьер и Микелон	Saint Pierre and Miquelon
SRB	Сербия	Serbia
SSD	Южный Судан	South Sudan
STP	Сан-Томе и Принсипи	Sao Tome and Principe
SUR	Суринам	Suriname
SVK	Словакия	Slovakia
SVN	Словения	Slovenia
SWE	Швеция	Sweden
SWZ	Свазиленд	Swaziland
SXM	Синт-Мартен	Sint Maarten (Dutch part)

SYC	Сейшельские Острова	Seychelles
SYR	Сирия	Syrian Arab Republic
TCA	Теркс и Кайкос	Turks and Caicos Islands
TCD	Чад	Chad
TGO	Того	Togo
THA	Таиланд	Thailand
TJK	Таджикистан	Tajikistan
TKL	Токелау	Tokelau
TKM	Туркмения	Turkmenistan
TLS	Восточный Тимор	Timor-Leste
TON	Тонга	Tonga
TTO	Тринидад и Тобаго	Trinidad and Tobago
TUN	Тунис	Tunisia
TUR	Турция	Turkey
TUV	Тувалу	Tuvalu
TWN	Китайская Республика	Taiwan
TZA	Танзания	Tanzania United Republic of
UGA	Уганда	Uganda
UKR	Украина	Ukraine
UMI	Внешние малые острова (США)	United States Minor Outlying Islands
URY	Уругвай	Uruguay
USA	США	United States of America
UZB	Узбекистан	Uzbekistan
VAT	Ватикан	Holy See (Vatican City State)
VCT	Сент-Винсент и Гренадины	Saint Vincent and the Grenadines
VEN	Венесуэла	Venezuela
VGB	Виргинские Острова (Великобритания)	Virgin Islands British
VIR	Виргинские Острова (США)	Virgin Islands U.S.

VNM	Вьетнам	Viet Nam
VUT	Вануату	Vanuatu
WLF	Уоллис и Футуна	Wallis and Futuna
WSM	Самоа	Samoa
XKX	Косово	Kosovo
YEM	Йемен	Yemen
ZAF	ЮАР	South Africa
ZMB	Замбия	Zambia
ZWE	Зимбабве	Zimbabwe

Attachment: Main IBE error codes

Error code	Description returned by IBE	Error Description
IBE errors		
1000	Unknown IBE error	Unhandled exception, reason unknown.
1001	Operation can not be performed due to invalid reservation state	Error gets raised if certain API method was called at improper moment in booking flow. For instance if api/services method was called right after api/session (booking was not created yet)
1006	Failed to resolve timezone	Error might appear during failed attempt to refresh timezone dictionary
1007	Session pool is unable to acquire stateful session as it's expired	The engine tries to get information about the session in the Session Pool, but there is no such session.
1008	Data validation error	Validation error, might appear on all booking steps. For instance when invalid passenger data is being sent
1013	Unable to load IBE reservation context	It is not possible to get information on the open session ID of the booking engine.
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.
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.
1098	Airport code not found	The error appears if the airport code not found.

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.
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.
4100	OrderInfo service has failed to process request	The error occurs if the GOI-proxy was unable to fulfill the request.

Adding services to booking main error codes

Error code	IBE description	Error description
1000	Unknown error	Unknown error. Any error not handled by IBE.
1001	Invalid passenger or segment reference	Error appears when attempting to add service for passenger or segment not existing in booking.

Attachment: Partner's services.

Partner codes

Partner code	Title
1	Alfastrah
2	Sealine insurance
3	Service
4	Aeroexpress
7	Telemedicine
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

Aeroexpress service codes

№	Service code	Tariff decryption
1	STANDARTOW	Standard
2	STANDARTRT	Round trip

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

Alfastrah service codes

№	Service code	Tariff decryption
---	--------------	-------------------

1	AVIA	Avia
2	TRAVEL	Travel
3	LUGGAGE	Luggage
4	SPORT	Sport
5	NOFLIGHT	No flight
6	AVIADYNAMIC	Avia with dynamic cost calculation
7	AVIADYNAMICBUS	Avia for Business fares with dynamic cost calculation

Tariff description <https://www.uralairlines.ru/passengers-info/partner-services/insurance/>

Alfastrah service codes (Anticovid)

№	Service code	Tariff decryption
1	ANTICOVID19	Anticovid Insurance for disclosure of COVID-19

Sealine service codes

№	Service code	Tariff decryption	Availability
1	SLAVIA	Avia	Yes
2	SLAVIAPLUS	Avia+	Yes
3	SLSPORT	Sport	Yes
4	SLPROMO	Promo	Yes
5	SLTRAVEL	Travel	Yes

Sealine service codes (COVID insurance)

№	Service code	Tariff decryption
1	SLANTICOVID	Страхование АнтиВирус

Sberhealth service codes

№	Service code	Tariff decryption	Availability
1	SBERDOC	Therapist and Pediatrician Online	Yes

№	Service code	Tariff decryption	Availability
2	SBERVET	Veterinarian Online	Yes

Alfahealth service codes

№	Service code	Tariff decryption	Availability
1	AFLADOC	Therapist and Pediatrician Online	Yes

Business rooms service codes

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

Attachment: 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/SegmentRefernece/PassengerReference

Examples:

DISCOUNT/DS20/1/1-5KYLTVG/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
SegmentRefernece	Reference to segment	Discount applies to all segment or not applies at all. Separate SSR is added for each segment.
PassengerReference	Reference to passenger	-

SSR with information about 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/PassengerReferenc
e**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

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	-

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/500.00/1/0.00/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	<p>A link to a segment is not added if the partner service is added to all segments. Currently only used for baggage packing partner.</p> <p>Example: S12</p>
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".