

# Internet Booking Engine

## Документация API

<b>Версия документа</b>	v2.3.73 2.3 - рекомендуемая версия API 73 – номер спринта, в котором были выполнены последние изменения
<b>Поддерживаемые версии API</b>	v2.2, v2.3
<b>Устаревшие версии</b>	v1.1, v2.0, v2.1

# Оглавление

Internet Booking Engine .....	1
Документация API .....	1
История изменений.....	4
Глоссарий.....	21
Общие положения.....	22
Общая информация о методах API .....	23
Актуальная версия API.....	23
Общая информация о методах API v2.x .....	23
HTTP(S)-заголовки v2.x .....	23
Вызов сервисов IBE API.....	24
Ответ от сервисов IBE API.....	25
Описание процесса бронирования .....	26
Общий процесс бронирования.....	26
Описание процесса докупки дополнительных услуг после оплаты .....	27
Описание методов API .....	29
Сервис для получения сессии .....	29
Сервис получения минимальных цен .....	30
Сервис получения перелётов .....	40
Сервис управления бронированием .....	59
Сервис получения информации о партнёрских услугах .....	117
Сервис подтверждения (прайсинга) тарифов в PNR.....	123
Сервисы управления дополнительными услугами.....	128
Сервис управления партнёрскими услугами.....	152
Сервис получения информации о партнёрских услугах .....	162
Сервис получения схемы мест в самолёте .....	168
Сервис оплаты бронирования.....	172
Сервис авторизации в CRM .....	178
Сервис получения информации о связанных пользователях (CRM).....	185
Сервис управления промокодами (CRM).....	188
Сервис для получения скидок.....	195
Сервис получения и отправки на почту маршрутной квитанции .....	199

Сервис добровольного возврата билетов и услуг .....	201
Сервис вынужденного возврата билетов. Упрощенная схема. ....	216
Сервис изменения данных о пассажире.....	219
Сервис онлайн-регистрации .....	226
Сервис аннулирования билетов.....	232
Сервис получения схемы мест в самолете без PNR .....	234
Сервис получения каталога услуг без PNR .....	240
Сервис управления сертификатами.....	251
Описание служебных методов API .....	257
Сервис для продления сессии Amadeus .....	257
Сервис для получения списка аэропортов.....	258
Сервис для проверки работоспособности движка бронирования.....	260
Сервис отправки отчёта о проблеме .....	263
Сервис для получения курса валют .....	265
Сервис для получения правил транслитерации кириллицы .....	267
Сервис для отслеживания состояния резервного контура.....	270
Сервис для отслеживания состояния резервного контура.....	271
Приложения.....	273
Приложение 1. Коды ошибок IBE.....	273
Приложение 2. Справочники .....	285
Приложение 3. Ремарки/служебные SSR .....	305
Приложение 4. Партнёрские услуги. ....	312
Приложение X. Примеры ответов от сервисов .....	315

# История изменений

Дата изменения	Версия	Описание изменения	Автор
25.03.2021	2.3.73	<p>Добавлено:</p> <ul style="list-style-type: none"><li>• Приложение 2. Справочники: добавлены новые типы пассажиров: UNN, STR</li><li>• Приложение 2. Справочники: добавлены новые пакеты услуг: PAKR и PAKZ</li><li>• в ответ сервиса POST /Refund/Calculate параметр массив строк refundWithTicketOnlyServices подкатегории data;</li><li>• в запросы сервисов POST /Reservation и /Flights/Search целочисленные параметры StrAdults, ChrChildren и UnaccompaniedMinors ;</li><li>• в ответ запроса /Flights/Search строка ruleKey по пути data.recommendations.outboundFareAndAmountDetails</li></ul> <p>Изменено:</p> <ul style="list-style-type: none"><li>• в Технические задания: для тестового окружения изменён эндпоинт, из которого получаем информацию о питании в опциях тарифа с <a href="https://webticket.uralairlines.ru/test/foodinfo/getjson">https://webticket.uralairlines.ru/test/foodinfo/getjson</a> на <a href="https://loyalty-tst.u6.ru/WebTicket/foodinfo/getjson">https://loyalty-tst.u6.ru/WebTicket/foodinfo/getjson</a></li></ul>	Кулиш К.В.

25.02.2021	2.3.72	<p>Добавлено:</p> <ul style="list-style-type: none"> <li>• в ответ сервиса GET api/Catalogue параметр subCategory подкатегории доп. услуги;</li> <li>• в запрос сервиса POST api/Refund/Partial добавлен массив services с информацией о возвращаемых доп. услугах АК;</li> <li>• в общую техническую информацию по возврату новые статусы сессий возвратов AutoRefundCompleted и ServicesCanceled;</li> <li>• для сервиса добровольного возврата билетов и услуг добавлен метод регистрации заказа на автовозврат в BPC - bmmRefund;</li> <li>• в запрос сервиса GET api/Calendar для поиска через SIG параметр twp, являющийся индикатором будет ли запрошен календарь на +-7 дней от текущей даты;</li> <li>• в ответ GET api/v2.3/Reservation параметр bookingTimeLimit времени и даты, до которого будет существовать неоплаченное бронирование;</li> <li>• в ремарку по добавленной наценке параметр ChargeType, обозначающий тип добавленной наценки (наценка из DMS или динамическая наценка из getContactMarkup);</li> <li>• в Приложение 4. Партнёрские услуги в таблицу Коды услуг Альфастрахование новый тариф AVIADYNAMICBUS, возвращаемый для Бизнес тарифов;</li> <li>• в Приложение 2. Справочник типов дополнительных услуг услуга CVID.</li> </ul> <p>Изменено:</p> <ul style="list-style-type: none"> <li>• сценарий возврата, в связи с добавлением возможности автовозврата;</li> <li>• в Приложении 2. Справочники доп. услуги XBLL и XBLS перенесены из справочника Типов дополнительных услуг, поддерживаемые движком (не продаются в движке) в справочник Типов дополнительных услуг, изменены категория и описание.</li> </ul>	Гудумак С.Е.
------------	--------	--	--------------

01.02.2021	2.3.71	<ul style="list-style-type: none"> <li>• В ответ GET api/Flights добавлен блок alternativeFlights с альтернативными направлениями из CRM;</li> <li>• добавлен запрос POST api/refund/partial, в параметры запроса добавлен признак возвращается билет или только услуги isTicketRefund и блок возвращаемых партнерских услуг partnerProducts;</li> <li>• добавлены примеры запросов для полного возврата билета с услугами и возврата партнерских услуг;</li> <li>• в ответ api/reservation добавлены параметры доступности и причины недоступности аннулирования соответственно isAnnulAvailable и annulUnavailabilityReason, а также признак доступности возвратов isRefundAvailable;</li> <li>• добавлено описание двух методов по добавлению и удалению услуг из заказа: api/services/add, api/services/remove;</li> <li>• в Приложение 2. Справочник типов дополнительных услуг добавлены коды питания: ECML, FCML, ACML, RCML, CSML, RUML, ORML, MEML, GEML, ASML;</li> <li>• в Приложение 2. Справочник опций тарифов добавлена опция noMealsOnBoard "Без питания".</li> </ul>	Гудумак С.Е.
23.12.2020	2.3.70	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• в ответ GET api/Partners/{id} массив compatibleProductCodes кодов тарифов партнерской услуги, которые могут быть добавлены в бронирование совместно с текущей;</li> <li>• в ответ GET api/Reservation, в блок appliedDiscounts параметр serviceReference, указывающий идентификатор услуги, на которую применена скидка, внутри текущего бронирования, и параметр promotionType, обозначающий тип промокода (на перелет или услугу);</li> <li>• в Приложение 2. Справочник типов дополнительных услуг коды услуг BLOU - бизнес-зала и VLOU - Vip-зала;</li> <li>• в Приложение 3 информация по добавлению SSR по партнерским услугам.</li> </ul>	Гудумак С.Е.

30.11.2020	2.3.69	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• блок <code>excludedPeriods</code> с информацией о периоде исключения действия промокода и параметр <code>loyaltyCardIsRequired</code> - флаг, обозначающий промокод для ДКЛ, в ответе <code>POST api/Promotions</code>;</li> <li>• параметры <code>arrivalTerminal</code> и <code>departureTerminal</code> с информацией о номере терминала в пункте прибытия и отправления соответственно и параметр <code>hash</code> с уникальным хешем перелета в ответе <code>GET api/Reservation</code>;</li> <li>• параметры <code>loyaltyCardValidTo</code>, обозначающий срок действия карты лояльности, и <code>serviceClassUpOrdered</code>, обозначающий повышение класса обслуживания для владельца золотой карты "Крылья", запроса <code>POST api/Reservation</code>;</li> <li>• параметры запроса <code>GET api/Flights</code>, влияющие на поиск через SIG: <code>mode</code> - режим поиска, <code>ofn</code> и <code>rfn</code> - массивы номеров рейсов туда и обратно соответственно;</li> <li>• параметры ответа <code>GET api/Flights</code>: <code>outboundFlightGroupHash</code> и <code>returnFlightGroupHash</code> - хеши групп перелетов туда и обратно соответственно; <code>outboundFareFamilyCode</code> и <code>returnFareFamilyCode</code> - коды семейств тарифов для направлений туда и обратно соответственно.</li> </ul>	Гудумак С.Е.
03.11.2020	2.3.68	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• блок <code>preselectedOption</code> с информацией о предвыборе для партнерских услуг в ответ метода <code>GET api/Partners/{id}</code>;</li> <li>• пример ответа при запросе Телемедицины.</li> </ul>	Гудумак С.Е.
09.10.2020	2.3.67	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• параметр <code>passengerReference</code> для запроса <code>POST api/Partners</code> для добавления телемедицины</li> <li>• параметр номера заказа партнерской услуги <code>orderId</code> в ответе сервиса <code>POST api/Partners</code></li> </ul>	Гудумак С.Е.

15.09.2020	2.2.66	<p>Добавлено:</p> <ul style="list-style-type: none"> <li>• в ответе метода GET api/reservation параметр appliedCertificates, содержащий информацию о примененных сертификатах;</li> <li>• новый метод GET api/Checkin/GetPassengerChekinStatus, позволяющий получить информацию о доступности регистрации с учетом доступности регистрации в разрезе каждого сегмента.</li> </ul>	Кислова М.О.
27.07.2020	2.2.64	<p>Добавлено:</p> <ul style="list-style-type: none"> <li>• для запроса POST api/payment параметр адреса возврата returnUrl.</li> </ul> <p>Изменено:</p> <ul style="list-style-type: none"> <li>• для ответа метода GET api/catalogue v2.3 в массивах services, packs и seatService ссылка referenceNumber на сегмент перелета изменена на массив ссылок на сегменты перелета referenceNumbers.</li> </ul> <p>Удалено:</p> <ul style="list-style-type: none"> <li>• для запроса POST api/services v2.3 блоки insurance и partnerProducts;</li> <li>• в ответе GET api/catalogue v2.3 блок insuranceProducts.</li> </ul>	Бодрова С.Е.



26.06.2020	2.2.63	<p>Удалено:</p> <ul style="list-style-type: none"> <li>• информация о неактуальном запросе GET api/discounts;</li> <li>• в Приложении 4 удалены неактуальные коды услуг Аэроэкспресс: ADVSTANDART - Стандарт Предварительный, STANDARTRT - Туда-обратно, PAIROW - Парный, PAIRRT - Парный Туда-обратно, GROUPOW - Группа, GROUPRT - Группа Туда-обратно.</li> </ul> <p>Добавлены:</p> <ul style="list-style-type: none"> <li>• в ответе метода GET api/catalogue параметр crmEmdPromoCodes, содержащий информацию о промокодах на дополнительные услуги, получаемые от сервиса CoreApi;</li> <li>• в ответе метода GET api/flights параметр aircraftIataCode со значением IATA-кода воздушного судна;</li> <li>• в ответах методов GET, POST api/reservation параметр aircraftIataCode со значением IATA-кода воздушного судна;</li> <li>• новый сервис POST api/discounts, который позволяет получить применимую скидочную компанию по заданным параметрам поиска</li> <li>• новый сервис по управлению сертификатами GET, DELETE api/certificates, который позволяет получать информацию о сертификатах и удалять их для текущего бронирования</li> <li>• новый сервис POST api/CatalogueStandalone для получения каталога услуг без создания PNR.</li> </ul> <p>Изменено:</p> <ul style="list-style-type: none"> <li>• сервис api/SeatmapStateless переименован в api/SeatmapStandalone.</li> </ul>	<p>Кислова М.О.</p> <p>Бодрова С.Е.</p>
------------	--------	--	---

01.06.2020	2.2.62	<p>Удалена информация об устаревшей версии api 2.1</p> <p>Добавлены:</p> <ul style="list-style-type: none"> <li>• в ответ метода GET api/catalogue блок с пакетами услуг racks и массив характеристик мест seatCharacteristics;</li> <li>• в ответ метода GET api/seatmap параметр isSeatUnavailable и массив объектов с информацией о стоимости мест в разрезе пассажиров passengersTotalData;</li> <li>• добавлен новый сервис POST api/SeatmapStateless для получения схемы мест без создания PNR;</li> <li>• в ответ метода GET api/reservation добавлен параметр rackReferenceNumber ссылки на добавленный в бронирование пакет услуг;</li> <li>• в Приложение 2 добавлен справочник пакетов услуг;</li> <li>• в Приложение X добавлен пример успешного запроса и ответа для сервиса POST api/SeatmapStateless.</li> </ul>	Бодрова С.Е.
06.05.2020	2.2.61	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• методы GET и POST api/annul</li> <li>• параметры ответа метода GET api/reservation: <ul style="list-style-type: none"> <li>○ reservationRetrieveStatus - статус получения бронирования;</li> <li>○ engineType - система, в которой создано бронирование;</li> <li>○ isGroupReservation - флаг, определяющий групповое бронирование;</li> <li>○ outboundBaggageAllowances и returnBaggageAllowances - массивы, определяющие нормы багажа;</li> </ul> </li> <li>• в ответ GET api/flights v2.2 параметр, определяющий распродажный тариф isSale;</li> <li>• в Приложение 2 новые типы питания и услуг, поддерживаемых движком.</li> </ul>	Бодрова С.Е.
03.04.2020	2.2.60	Добавлен блок promotion с примененной промокампанией в ответ api/flights v2.2	Бодрова С.Е.

02.03.2020	2.2.59	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• параметр MoneyWallCoefficient в ответе запросов api/flights и api/catalogue, для расчета суммы ежемесячного платежа при оплате в рассрочку</li> <li>• параметр BaggageAllowance в блок recommendations ответа api/flights, для определения норм багажа</li> <li>• параметр NextNearestFlightDate в ответе запроса api/flights</li> <li>• добавлен метод GET /api/system/Availability для отслеживания состояния резервного контура движка в IT Grad</li> </ul>	Крутякова Е.С.
27.02.2020	2.2.58	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• описание параметра text в блоке services запроса api/services;</li> <li>• пример запроса POST services для добавления провоза животного;</li> <li>• справочник животных в Приложение 2.</li> </ul>	Маслянный И. В.
03.02.2020	2.2.58	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• параметры serviceCode, promoNumber в блок appliedDiscounts ответа api/reservation;</li> <li>• параметры text, petType в блок services ответа api/reservation;</li> <li>• параметр redirectLang в запросе POST api/payment.</li> </ul> <p>Изменено:</p> <ul style="list-style-type: none"> <li>• тело и описание параметров запроса POST api/payment.</li> </ul>	Бодрова С.Е.

25.12.2019	2.2.57	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• информация о методах partners, которые позволяют запрашивать информацию о партнёрских услугах отдельно от api/catalogue;</li> <li>• новые типы питания: CAML (десерт) и KSML (кошерное);</li> </ul> <p>Изменения:</p> <ul style="list-style-type: none"> <li>• услуги SLSPORT и SLTRAVEL для партнёра Sealine доступны для бронирования</li> <li>• изменена структура запроса api/flights переводы в блоке foodOnBoard перенесены в объект translation</li> </ul>	<p>Холманов А.А.</p> <p>Вершинин А. В.</p>
09.12.2019	2.2.56	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• параметры startFlightDate, endFlightDate, startDate, endDate в ответе на запрос POST api/promotions</li> </ul> <p>Удалены:</p> <ul style="list-style-type: none"> <li>• параметры serviceType для запросов GET api/calendar и GET api/flights</li> </ul>	<p>Бодрова С. Е.</p>

20.11.2019	2.2.55	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• поле showInBookingCart в опциях в ответе GET api/reservation;</li> <li>• поле originOfficeId с информацией об офисе, в котором сделано бронирование, в ответе GET api/reservation;</li> <li>• поле SearchPrice с информацией о цене, которая была получена в поисковом запросе, в запросе POST api/reservation;</li> <li>• поля adults, children, infants, pensioners, youths, subsidizedChildren - с информацией о количестве пассажиров каждого типа, в запросе POST api/reservation;</li> <li>• поле origin - Код IATA аэропорта/хаба отправления, в запросе POST api/reservation;</li> <li>• поле destination - Код IATA аэропорта/хаба назначения, в запросе POST api/reservation;</li> <li>• уточнение в приложении 2 для справочника типов офисов Amadeus;</li> </ul>	<p>Вершинин А. В.</p> <p>Крутякова Е.С</p>
08.11.2019	2.2.54	Добавлен блок partnerProducts в ответ api/refund/calculate	Бодрова С. Е.
14.10.2019	2.2.53	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• список партнёров в Приложение 4 и коды услуг партнёрских сервисов Sealine/Alfastrah;</li> <li>• пример с покупкой партнёрской страховки в движке в версии 2.2 сервиса управления дополнительными услугами api/services;</li> <li>• добавлены изменения в версии 2.3 сервиса управления дополнительными услугами /api/services;</li> <li>• пример получения информации о партнёрской страховке с помощью сервиса api/reservation/partners;</li> <li>• уточнение ответа api/calendar, добавилось поле fareFamilies в объекте prices → discount;</li> <li>• уточнение ответа api/refund/calculate - добавлен блок партнерских услуг</li> </ul>	<p>Вершинин А. В.</p> <p>Берёзка О. М.</p>

08.10.2019	2.2.52	<p>Приложение с примерами переименовано и помещено в конец документации. Актуализированы номера других приложений.</p> <p>Добавлены:</p> <ul style="list-style-type: none"> <li>• технические поля ticketFares и ticketFaresWithoutZapOff в ответ POST api/reservation. Используются для показа информационного сообщения о разнице в цене SIG/Amadeus;</li> <li>• в ответ api/flights добавился массив discountConstraints, который содержит настройки применения скидочных акций;</li> <li>• в ответ api/flights добавился массив excludedNationalities, который содержит коды национальностей, запрещенных к перевозке на данном рейсе;</li> <li>• в описание сервиса /api/promotions добавлены коды ошибок, возвращаемые сервисом и соответствующие им тексты.</li> </ul>	Вершинин А. В. Воронина Д. А.
25.09.2019	2.2.51	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• поле priceHidden в ответ GET api/reservation, которое позволяет управлять показом цены для IT-тарифов.</li> </ul>	Вершинин А. В.

11.09.2019	2.2.50	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• параметр serviceType в запросах api/calendar и api/flights, который позволяет управлять поиском через Amadeus/SIG;</li> <li>• поле airShopSessionId в ответ api/flights;</li> <li>• поля expectedPrice и airShopSessionId в запрос POST api/reservation. Используются при бронировании с поиском через SIG.</li> </ul> <p>Общие изменения в документации:</p> <ul style="list-style-type: none"> <li>• удалена информация об устаревших версиях 1.1 и 2.0;</li> <li>• версия 2.3 добавлена в список;</li> <li>• уточнено описание процесса бронирования;</li> <li>• добавлена информация о методах смежных систем, которые использует API.</li> </ul>	<p>Бодрова С. Е.</p> <p>Вершинин А. В.</p>
22.08.2019	2.2.49	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• описание сервиса вынужденного возврата билетов.</li> </ul>	<p>Кольцова О. М.</p>
07.08.2019	2.2.48	<p><b>Изменён подход к нумерации документации. Теперь номер содержит рекомендуемую версию API и номер спринта, по изменениям в котором документация составлена.</b></p> <p>Добавлены:</p> <ul style="list-style-type: none"> <li>• описание сервиса GET settings/transliteration, который позволяет получить правила транслитерации кириллицы;</li> <li>• массив foodOnBoards в ответ api/flights, который содержит информацию о доступном питании на рейсах;</li> <li>• массив foodOnBoards в ответ GET api/reservation, который содержит информацию о доступном питании на рейсах.</li> </ul>	<p>Смирнова С.А.</p> <p>Вершинин А. В.</p>

22.07.2019	1.4.47	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>описание сервиса <code>api/reservation/partners</code>, который позволяет получать информацию по партнёрским услугам в бронировании и выполнять переотправку билетов/полисов по выпущенным услугам.</li> </ul>	Вершинин А. В.
04.07.2019	1.4.46	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>поле <code>bookingStatus</code> в ответ <code>GET api/catalogue</code>;</li> <li>уточнение про поле <code>serviceOptions</code> в ответе метода <code>api/catalogue</code>;</li> <li>описание метода <code>api/refund</code> и <code>api/refund/calculate</code>, позволяющие выполнять добровольный возврат бронирований;</li> <li>описание метода <code>api/passenger</code>, который позволяет вносить изменения в пассажирские данные в бронировании;</li> <li>коды ошибок в приложение 1</li> </ul>	Вершинин А. В.
10.06.2019	1.4.45	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>описание поля <code>sessionMarker</code> в ответ <code>GET api/reservation</code>.</li> </ul>	Смирнова С.А.
29.05.2019	1.4.44	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>поле <code>serviceOptions</code> в ответ <code>GET api/catalogue</code>;</li> <li>изменения в методе <code>api/flights v2.2</code> (тарифы для субсидированных типов пассажиров);</li> <li>изменения в методе <code>api/reservation v2.2</code> (бронирование для субсидированных типов пассажиров);</li> <li>изменения в методе <code>api/services v2.2</code> (покупка дополнительных услуг после оплаты бронирования);</li> <li>описание процесса докупки дополнительных услуг для оплаченного бронирования;</li> <li>информация о возможности авторизации с токеном для метода <code>api/user</code>.</li> </ul>	Кольцова О. М.  Вершинин А. В.



20.05.2019	1.4.43	<p>Была изменена информация о рекомендуемой версии API: вместо версии 1.1 рекомендуется использовать версию 2.1.</p> <p>Версия 1.1 помечена, как устаревшая.</p> <p>Обновлена информация по ошибкам IBE в приложении 1.</p>	Вершинин А. В.
28.04.2019	1.4.42	<p>Ошибки при добавлении услуг. Добавлено:</p> <ul style="list-style-type: none"> <li>• коды ошибок при добавлении услуг в бронирование (приложение 1)</li> <li>• описание сервиса api/services v2.1, в ответ добавлена структура с ошибками</li> </ul> <p>Аэроэкспресс. Добавлено:</p> <ul style="list-style-type: none"> <li>• информация о партнёрских услугах partnerProducts в ответе api/catalogue</li> <li>• информация о партнёрских услугах в запросе api/services</li> <li>• информация о партнёрских услугах в ответе api/reservation</li> <li>• приложение 5 с кодами партнёров и партнёрских услуг</li> <li>• уточнение об отправке информации в BPC для партнёрских услуг, метод api/payment</li> </ul> <p>Другое:</p> <ul style="list-style-type: none"> <li>• добавлена информация о записываемых в PNR ремарках в приложении 4.</li> <li>• уточнено описание сервиса api/session</li> </ul>	Вершинин А. В.
12.04.2019	1.4.39	Добавлено поле marketingCarrier в ответы api/reservation и api/flights	Воронина Д. А.

11.03.2019	1.4.39	<p>Изменения, внесённые в 38 и 39 спринтах.</p> <p>Добавлены:</p> <ul style="list-style-type: none"> <li>• описание тарифов промо-лайт, багажный, безбагажный. Удалён бизнес-лайт из списка тарифов.</li> <li>• описание типа пассажира CNN, субсидированный ребёнок.</li> <li>• описание субсидированного тарифа в справочник тарифов.</li> <li>• описание параметра cnn в запросе api/calendar v1.1.</li> <li>• описание параметра cnn в запросе и признак isSubsidized в ответе api/flights v1.1.</li> <li>• описание правил валидации при создании бронирований с субсидированным тарифом, метод POST api/reservation v1.1.</li> </ul>	Вершинин А. В.
11.01.2019	1.4.37	<p>Добавился раздел с описанием API 2.0</p> <ul style="list-style-type: none"> <li>• появился метод POST api/session вместо api/token;</li> <li>• добавлены заголовки X-API-Key и X-Session;</li> <li>• убран заголовок Authorization;</li> </ul>	Смирнова С. А.
19.12.2018	1.4.35	<p>Добавлены:</p> <ol style="list-style-type: none"> <li>1. Изменения в ответе GET api/reservation <ol style="list-style-type: none"> <li>1. массив tickets с информацией о билетах;</li> <li>2. добавлены поля spentBonuses и spentBonusesByPassengers;</li> <li>3. добавлено поле originOfficeType;</li> <li>4. добавлено поле hasSpecialCondition</li> </ol> </li> <li>2. Добавлена информация о времени жизни токена.</li> <li>3. Добавлено уточнение о работе сервиса POST api/token.</li> <li>4. В ответ api/token добавилось поле ID</li> <li>5. Добавился заголовок X-Stamp для метода api/flights</li> </ol>	Вершинин А. В.

01.12.2018	1.4	<p>Добавлены:</p> <ul style="list-style-type: none"> <li>• описание процесса бронирования;</li> <li>• новый тип токена, type=Skyscanner;</li> <li>• описание метода POST api/reservation версии 1.1, добавлены примеры создания бронирований;</li> <li>• описание zapOff (информация о скидках) в ответе api/pricing;</li> <li>• уточнение для формата ответа метода api/catalogue;</li> <li>• поле totalPrice в ответе api/flights -&gt; recommendations, которое обозначает полную стоимость билетов за вычетом применённых скидок. Приведены примеры;</li> <li>• поле isEmpty в ответе api/seatmap -&gt; data. Признак доступности выбора мест на сегменте;</li> <li>• описание сервиса api/discounts;</li> <li>• поля в ответ api/user, обозначающие страну выпуска документа;</li> <li>• входной параметр ticketNumber в метод GET api/reservation, позволяющий запрашивать информацию о бронировании по номеру билета и фамилии;</li> <li>• входной параметр disc (служебный) для сервиса api/calendar. Позволяет контролировать запрос скидок;</li> <li>• описание сервиса api/ problemNotification;</li> <li>• описание сервиса api/receipt;</li> <li>• из ответа метода POST api/promotions удалено поле isValid;</li> <li>• обновление описания параметра cache для сервиса api/calendar;</li> <li>• описание метода api/exchanges.</li> </ul>	Вершинин А. В.
29.08.2018	1.3	<p>Изменена версия API на 1.1. Добавлены:</p> <ul style="list-style-type: none"> <li>• описание сервиса api/airports</li> <li>• описание сервиса api/payment версии 1.1</li> <li>• описание параметра cache для сервиса api/calendar</li> <li>• описание сервиса api/health</li> </ul>	Вершинин А. В.

19.06.2018	1.2	<ol style="list-style-type: none"> <li>1. Уточнено описание входных параметров метода POST api/payment.</li> <li>2. Добавлено описание тарифных опций и категорий тарифов, приложение 2.</li> <li>3. Сервис api/services разделён на два: GET api/catalogue для запроса дополнительных услуг, POST api/services для добавления услуг в бронирование.</li> <li>4. Метод для сервиса api/seatmap изменён с POST на GET. Добавлено описание структуры passengerRelatedPrice.</li> <li>5. Добавлены примеры использования поля flightDurationUntilNextStop из ответа api/flights.</li> </ol>	Вершинин А. В.
04.06.2018	1.1	Добавлено описание основных сервисов API, необходимых для создания бронирования.	Вершинин А. В.
23.05.2018	1.0	Начальная версия документа	Вершинин А. В.

# Глоссарий

U6	Компания Ural Airlines
IBE, движок бронирования	Internet Booking Engine, новая система бронирования билетов компании U6, API которой описывается в данном документе
Amadeus	Глобальная система бронирования авиабилетов (GDS)
SIG	Интегратор поисковых решений из различных GDS
CRM	Система управления клиентами, интеграция с которой выстроена у IBE
BPC	Система оплаты бронирований, интеграция с которой выстроена у IBE
WSAP	Web Service Access Point, точка доступа к веб-сервисам Amadeus
PNR	Passenger name record, запись о бронировании на стороне Amadeus, которая содержит информацию о перелёте, данных пассажира, дополнительных услугах, скидках и др.
SSR	Special Service Request. В текущем документе SSR почти всегда используется в значении «отдельная запись в PNR». Запись может содержать информацию о пассажире, услуге и др.
TST	Transitional Stored Ticket, маска документа Amadeus с информацией о тарифных опциях для PNR.
TSM	Transitional Stored Miscellaneous, маска документа Amadeus с информацией о дополнительных услугах, добавленных в PNR.

## **Общие положения**

Назначение данного документа – предоставить описание методов IBE API и примеров их использования для сторонних разработчиков.

IBE API представляет собой REST API, который позволяет искать билеты и цены, бронировать билеты на различные направления, добавлять дополнительные услуги и производить оплату.

IBE API является прослойкой между системами Amadeus/SIG/CRM/BPC, другими партнёрскими сервисами, и конечным клиентским приложением.

# **Общая информация о методах API**

## **Актуальная версия API**

**Текущая версия API: v2.3**

Рекомендуется для всех запросов использовать эту версию.

Вызов сервисов без указания версии API невозможен (за исключением сервиса api/health).

**Поддерживаемые версии: v2.2, v2.3**

**Устаревшие версии: v1.1, v2.0**

## **Общая информация о методах API v2.x**

Основные принципиальные отличия версии 2.x от версии 1.x:

Добавлена возможность вместо токена использовать уникальный API-ключ. Он позволяет запрашивать информацию для календаря и перелётов без создания сессии.

Если необходимо сохранить информацию в контекст бронирования, то через метод api/v2.x/session необходимо создать сессию бронирования и передать ID сессии в заголовке X-Session, при этом заголовок X-API-Key также должен передаваться.

**На данный момент ключи заведены для:**

Arcadia (веб-сайт движка бронирования),

Axmor (мобильное приложение),

CRM (получение цен для рассылок),

ВРС (создание тестовых бронирований),

Skyscanner (получение цен для мета-поисковика),

Aviasales (получение цен для мета-поисковика)

Ключ API предоставляется по запросу.

## **HTTP(S)-заголовки v2.x**

Для вызова сервисов IBE API v2.x должны использоваться следующие HTTP(S)-заголовки:

Название заголовка	Значение	Назначение	Исключения
X-API-Key	xApiKey — это значение ключа, зависимое от типа приложения (arcadia, ahmor, skyscanner)	API-ключ для авторизации	Метод GET Airports Метод GET ExchangeRates Метод POST ProblemNotification Метод GET Settings/Transliteration
X-Session	sessionKey — это значение, полученное при вызове сервиса API Session	Управление сессиями через заголовки HTTP	Метод POST session Метод GET flights Метод GET calendar Метод POST promotions Метод GET reservation Служебные методы, кроме renew и problemNotification.
Content-Type	application/json	Общее	Нет

### Вызов сервисов IBE API

Строка вызова сервиса IBE API в общем случае выглядит следующим образом: <http://host/api/apiVersion/serviceName>

где host – сервер, где развёрнут движок бронирования,

api – константа,

apiVersion – нужная версия IBE API в формате «v2.x»



serviceName – имя сервиса IBE API, например, «session» или «flights»

## **Ответ от сервисов IBE API**

IBE API обрабатывает исключительные ситуации, которые возникают при вызове сервисов, поэтому в общем случае ответы от сервисов могут быть следующие:

200 OK – успешный и неуспешный. В случае неуспешного ответа поле success в ответе будет иметь значение false, а массив ошибок errors будет содержать код и текст ошибки.

400 Bad Request – в случае, если запрос составлен неверно. Модель ответа может отличаться от той, что используется для корректного ответа сервиса.

401 Unauthorized – в случае, если не был передан API-ключ.

Также могут возвращаться коды, которые специфичны для конкретного сервиса.

# Описание процесса бронирования

## Общий процесс бронирования

Процесс бронирования состоит из следующих вызовов API v2.x:

№	Метод API	Описание
1	POST api/session	Получение уникальной сессии бронирования. В рамках сессии хранится контекст бронирования.
2	GET api/flights	Получение информации о перелётах, тарифах и ценах.
3	POST api/reservation	Создание бронирования: <ol style="list-style-type: none"><li>1. Добавление информации о перелётах;</li><li>2. Добавление информации о пассажирах;</li><li>3. Добавление информации о тарифах.</li></ol> Может быть выполнено 2 способами: <ol style="list-style-type: none"><li>1. Одним запросом с отправкой всех данных.</li><li>2. Двумя запросами с отправкой сначала данных о перелётах, а затем данных о пассажирах и тарифах.</li></ol> На фронтенде IBE используется 2ой вариант.
4	GET api/catalogue	Опционально. Получение списка и цен дополнительных услуг авиакомпании и партнёрских услуг.
5	GET api/seatmap	Опционально. Получение схемы рассадки самолёта и цен на места.
6	POST api/services	Опционально. Добавление в бронирование дополнительных услуг авиакомпании и/или партнёрских услуг. Услуги могут быть получены на шагах 4 и 5.
7	GET api/partners/types	Опционально. Получение списка доступных партнёров.

8	GET api/partners/[id]	Опционально. Получение списка услуг для конкретного партнёра. Где id - номер этого партнёра (представлены в приложении).
9	POST api/partners	Опционально. Добавление партнёрской услуги в бронирование.
10	POST api/payment	Получение ссылки на страницу оплаты.
11	GET api/reservation	Опционально. Получение информации об оплаченном бронировании (выпущенные билеты и др.).

### Описание процесса докупки дополнительных услуг после оплаты

Начиная с версии API 2.2 в движке реализована возможность докупки дополнительных услуг в рамках новой сессии.

№	Метод API	Описание
1	-	Создание и оплата бронирования по процессу, описанному выше.
2	POST api/session	Получение новой сессии, в которой планируется докупать услуги.
3	GET api/reservation	Обязательно наличие заголовка X-Session. Загрузка контекста готового бронирования в сессию.
4	GET api/catalogue	Получение списка и цен дополнительных услуг авиакомпании и партнёрских услуг.
5	GET api/seatmap	Опционально. Получение схемы рассадки самолёта и цен на места.
6	POST api/services	Добавление в бронирование дополнительных услуг
7	POST api/payment	Получение ссылки на страницу оплаты.

#### Примечания:

1. Если бронирование не было оплачено, то будет сформирована ссылка на оплату полного заказа (все неоплаченные части бронирования).
2. Временно докупка страховок и партнёрских услуг недоступна (метод api/catalogue не вернёт информацию, если бронирование оплачено).



# Описание методов API

## Сервис для получения сессии

<b>Сервис API</b>	session
<b>Краткое описание</b>	Позволяет получить сессию для сохранения контекста бронирования.
<b>Поддерживаемые HTTP(S)-методы</b>	POST
<b>Используемые методы смежных систем</b>	нет

### Метод POST session

**Описание:** создание новой сессии

На данный момент сессия истекает за 60 минут, каждый запрос автоматически продлевает её.

**Формат запроса** /api/apiVersion/session

### Заголовки запроса

X-API-Key - уникальный GUID для систем, использующих IBE API.

### Параметры запроса

Нет параметров

### Формат ответа

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

### Параметры ответа

Параметр	Описание
----------	----------

sessionKey	Ключ сессии Используется для хранения контекста бронирования
sessionId	Id сессии Служебное поле

### Пример ответа

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

### Сервис получения минимальных цен

<b>Сервис API</b>	calendar
<b>Краткое описание</b>	Получение минимальных цен на авиабилеты по датам. В основном используется для отображения цен в календаре.
<b>Поддерживаемые HTTP(S)-методы</b>	GET
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> Fare_InstantFlexPricer и Fare_FlexPricerCalendar <b>SIG:</b> SIG_AirShop, SIG_AirMinPrice <b>CRM:</b> CalculateDiscountList2

### Метод GET calendar

**Формат запроса:** /api/apiVersion/calendar

### Примеры запросов

Перелёт туда, поиск через Amadeus:

/api/apiVersion/calendar

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

Перелёт туда и обратно, поиск через SIG:

/api/apiVersion/calendar

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

## Параметры запроса

**ВАЖНО:** для получения корректного ответа необходимо запрашивать либо субсидированные типы пассажиров SRC / YTH / CNN, либо обычные ADT / CHD / INF.

Параметр	Тип	Описание	Примеры значений
orig	string	Original airport, IATA-код аэропорта вылета	LED
dest	string	Destination airport, IATA-код аэропорта вылета	DME
dep	date	Departure date, дата вылета туда	Формат ууууММдд Пример: 20180801
ret	date	Return date, дата вылета обратно	Формат ууууММдд Пример: 20180905
adt	integer	Adults, количество взрослых пассажиров	Целое число в интервале [0; 9]. Общее количество пассажиров не должно превышать 9
chd	integer	Children, количество детей	
inf	integer	Infants, количество младенцев	
src	integer	Senior citizen, количество пенсионеров	
yth	integer	Youth, количество пассажиров типа «молодёжь»	

cnn	integer	Количество пассажиров типа «субсидированный ребёнок»	
mth	boolean	Запрашивать значение по месяцам или нет. false – стандартный запрос, true – запрос значений по месяцам	false
twp	boolean	Запрашивать значение за период +-7 дней от текущей даты или нет (работает только для поиска через SIG) false – стандартный запрос, true – запрос значений +-7 дней от текущей даты	false



cache	string	<p>Default (0) – сначала запрашивать метод Амадеуса Fare_InstantFlexPricer для получения цен, если произошла ошибка – запрашивать Fare_FlexPricerCalendar.</p> <p>OnlyCache (1) – запрашивать только метод Амадеуса Fare_InstantFlexPricer для получения цен</p> <p>SkipCache (2) – запрашивать только метод Амадеуса Fare_FlexPricerCalendar для получения цен</p> <p>Значение по умолчанию – 0.</p> <p>Нельзя использовать этот параметр в случае, если параметр mth имеет значение true.</p> <p>Изменена логика вызова сервисов Amadeus:</p> <ol style="list-style-type: none"> <li>1. Если запрашиваемая дата перелёта <math>\leq 2</math> недели от текущей даты, то сразу запрашивается Fare_FlexPricerCalendar.</li> <li>2. Также FlexPricerCalendar запрашивается в периоды 15.12 - 15.01 и 22.04 - 15.05 (все даты включительно).</li> </ol>	0
-------	--------	---	---

disc	boolean	<p>Служебный параметр, позволяет контролировать запрос скидок.</p> <p>Значение по умолчанию – true.</p> <p>disc = false используется для запроса информации для свайперов.</p> <p>Вызывается только метод Amadeus Fare_FlexPricerCalendar.</p> <p>disc = true используется для запроса информации для календаря со скидками по месяцам.</p> <p>Вызывается метод Amadeus Fare_FlexPricerCalendar и метод CRM CalculateDiscountList2.</p>	-
------	---------	---	---

### Формат ответа

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

```

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса.  Если массив ошибок errors пуст, то значение поля будет true
data	Данные, которые вернул запрос
range	Интервал в днях, за который были возвращены данные
prices	Массив объектов с информацией о ценах

		price	Минимальная цена для перелёта туда в дату, которая указана в поле prices.date
		date	Дата перелёта туда, для которой сервис вернул минимальную цену  Формат: уууу-ММ-ддТНН:мм:ss  Пример: 2018-06-03T00:00:00
		flightsNumbers	Массив строк с информацией о номерах рейсов для конкретного перелёта. Содержит несколько значений, если рейс с пересадками.
		fareAmounts	Массив объектов, содержащий информацию о цене билета для конкретного типа пассажира.
		passType	Тип пассажира.  Возможные значения: [link] справочник типов пассажиров
		passCount	Количество пассажиров определённого типа
		taxAmount	Тарифы и сборы
		totalAmount	Полная стоимость билета, включая тарифы и сборы

		<b>discount</b>	<p>Объект с информацией о скидке, доступной для данного перелёта.</p> <p>Скидка может отсутствовать для одного или нескольких перелётов, если рейсы находятся в исключении. Исключения хранятся на стороне CRM и IBE.</p>
		id	ID кампании скидок в CRM
		name	Название кампании скидок
		amount	Размер скидки
		discountType	<p>Тип скидки, возможные значения:</p> <p>0 – Percent, проценты</p> <p>1 – Roubles, рубли</p> <p>Во избежание проблем рекомендуется проверять тип скидки в связи с её значением. Например, для типа 0 размер должен быть в интервале [0;100].</p>
		usageType	<p>Тип применения скидки, возможные значения:</p> <p>0 – Fare, тариф</p> <p>1 – Order, весь заказ</p>
		startDate	Дата начала действия скидки

		<b>endDate</b>	Дата окончания действия скидки
		<b>fareFamilies</b>	Массив строк с информацией о семействах тарифов, для которых может применяться скидка
		<b>returnPrices</b>	Массив объектов, который содержит минимальные цены для обратных дат перелёта, которые сочетаются с датой, указанной в <code>prices.date</code> . Данные берутся из соответствующих рекомендаций Amadeus.
		<b>price</b>	Минимальная цена для перелёта обратно в дату <code>returnPrices.date</code>
		<b>date</b>	Дата обратного перелёта, для которой сервис вернул минимальную цену
		<b>flightsNumbers</b>	Массив строк с информацией о номерах рейсов для конкретного перелёта. Содержит несколько значений, если рейс с пересадками.

			<b>fareAmounts</b>	<p>Массив объектов, содержащий информацию о цене билета для конкретного типа пассажира.</p> <p>Аналогичен массиву data.prices.fareAmounts.</p> <p>Для перелёта в одну сторону массив пуст.</p>
			passType	<p>Тип пассажира.</p> <p>Возможные значения: [link] справочник типов пассажиров</p>
			passCount	Количество пассажиров определённого типа
			taxAmount	Тарифы и сборы
			totalAmount	Полная стоимость билета, включая тарифы и сборы
<b>errors</b>				Информация об ошибках, возникших во время работы сервиса.
			ibeCode	Код ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>
			description	Описание ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>

	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю <code>ibeCode</code> .
	<code>code</code>	Код ошибки Amadeus
	<code>text</code>	Текст ошибки Amadeus

### Примеры ошибочных запросов и ответов

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

Параметр `dep` содержит прошедшую дату.

#### Ответ:

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

### Сервис получения перелётов

#### Сервис получения перелётов v2.2

<b>Сервис API</b>	flights
<b>Краткое описание</b>	Получение списка доступных перелётов на выбранные даты и рекомендаций с ценами билетов.



<b>Поддерживаемые HTTP(S)-методы</b>	GET
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> Fare_FlexPricerUpsell <b>SIG:</b> SIG_AirShop <b>CRM:</b> CalculateDiscount

Метод GET flights

**Описание:** получение списка перелётов и цен для них.

**Формат запроса:** /api/apiVersion/flights

### Примеры запросов

Перелёт туда:

/api/apiVersion/flights

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

Перелёт туда и обратно:

/api/apiVersion/flights

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

### Параметры запроса

**ВАЖНО:** для получения корректного ответа необходимо запрашивать либо субсидированные типы пассажиров SRC / YTH / CNN, либо обычные ADT / CHD / INF.

Параметр	Тип	Описание	Примеры значений
orig	string	Original airport, IATA-код аэропорта вылета	LED
dest	string	Destination airport, IATA-код аэропорта вылета	DME

dep	date	Departure date, дата вылета туда	Формат ууууММdd Пример: 20180801
ret	date	Return date, дата вылета обратно	Формат ууууММdd Пример: 20180905
adt	integer	Adults, количество взрослых пассажиров	Целое число в интервале [0; 9]. Общее количество пассажиров не должно превышать 9
chd	integer	Children, количество детей	
inf	integer	Infants, количество младенцев	
src	integer	Senior citizen, количество пенсионеров	
yth	integer	Youth, количество пассажиров типа «молодёжь»	
cnn	integer	Количество пассажиров типа «субсидированный ребёнок»	
mode	string	Режим поиска (необязательный параметр, влияет на поиск через SIG)	mono - запрашиваются только монорекомендации, например, PR туда и PR обратно;  full - запрашиваются все рекомендации.

ofn	string	Массив номеров рейсов для перелета туда (необязательный параметр, влияет на поиск через SIG)	60
rfn	string	Массив номеров рейсов для перелета обратно (необязательный параметр, влияет на поиск через SIG)	371,392

### Формат ответа

```
{
  "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
},

```

```

"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
}

```

**Формат ответа, если на запрашиваемые даты нет рейсов (рейс туда-обратно)**

```

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

```

## Параметры ответа

Параметр	Описание			
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true			
<b>data</b>	Данные, которые вернул запрос			
airShopSessionId	ID сессии поиска SIG			
currency	Валюта, в которой представлены цены			
moneyWallCoefficient	Коэффициент, для расчета ежемесячного платежа при оплате в рассрочку			
<b>outboundFlights</b>	Массив объектов flightGroup с информацией о перелётах туда			
<b>returnFlights</b>	Массив объектов flightGroup с информацией о перелётах обратно. В случае с перелётом в одну сторону массив значений будет пуст.			
<b>fareFamilies</b>	Массив с информацией о семействах тарифов			
fareFamilyId	Номер семейства тарифов			

code	Код семейства, например: ECOLIGHTRU/ECOFLEXRU/ECOPREMRU и др.			
shortCode	Сокращённый код для обозначения семейства тарифов. Например, для семейств, приведённых в пример выше: PR/EC/FL и др.			
category	Категория семейства, например: ECOLIGHT/ECOFLEX/ECOPREM Возможные значения: см. справочник основных категорий тарифов			
order	Порядок отображения семейства тарифов			
commercialFamily	Тип семейства тарифов, например: U6ECONOMY или U6BUSINESS			
isSubsidized	Признак субсидированного тарифа			
canRefund	Признак является ли тариф возвратным			
hasUpsale	Признак показывает можно ли данный тариф "повысить" до следующего в ценовой категории. Позволяет управлять показом блока Upsale на фронтенде.			
options	Массив объектов fareFamiliesOptions с информацией о тарифных опциях. Возможные значения: см. справочник опций тарифов			
outboundNextNearestFlightDate	Ближайшая дата вылета туда, если на запрашиваемую дату нет рейсов			
returnNextNearestFlightDate	Ближайшая дата вылета обратно, если перелет RT и на запрашиваемые даты нет рейсов			



<b>foodOnBoards</b>		Массив с информацией о питании, доступном на рейсах			
	departureAirport	Аэропорт вылета			
	arrivalAirport	Аэропорт прилёта			
	flightNumber	Номер рейса			
	commercialFamily	Тип семейства тарифов (на одном и том же рейсе может быть доступно разное питание для эконома и бизнеса).			
	translation	объект с опциями переводов			
	ru	Перевод опции на русский язык			
	en	Перевод опции на английский язык			
	zh	Перевод опции на китайский язык			
<b>alternativeFlights</b>		Альтернативные направления (возвращаются из CRM, в случае, если на выбранную дату перелеты не найдены)			
	departure	Аэропорт вылета альтернативного направления			
	arrival	Аэропорт прилета альтернативного направления			
<b>discountConstraints</b>		Массив объектов с информацией об исключениях для скидок			
	fareFamilies	Массив семейств тарифов, на который скидка должна действовать			
	flights	Массив номеров перелётов, на который скидка должна действовать.  Если в этом массиве отсутствует хотя бы 1 номер рейса из пересадок, то скидка не должна применяться.			
	excludedFlights	Массив объектов с информацией о рейсах-исключениях			
	flightNumber	Номер рейса			
	departureAirport	Аэропорт вылета			
	arrivalAirport	Аэропорт прилёта			

<b>recommendations</b>	Массив рекомендаций для перелётов			
outboundFlightGroupId	Номер перелёта туда, соответствует одному из номеров outboundFlights.flightGroupId			
outboundFareFamilyId	Номер семейства тарифов для перелёта туда. Соответствует одному из номеров fareFamilies.fareFamilyId			
returnFlightGroupId	Номер перелёта обратно, соответствует одному из номеров returnFlights.flightGroupId			
returnFareFamilyId	Номер семейства тарифов для перелёта обратно. Соответствует одному из номеров fareFamilies.fareFamilyId			
outboundFlightGroupHash	Хеш группы перелета туда			
returnFlightGroupHash	Хеш группы перелета обратно			
outboundFareFamilyCode	Код семейства тарифа для направления туда Соответствует fareFamilies.code			
returnFareFamilyCode	Код семейства тарифа для направления обратно  Соответствует fareFamilies.code			
outboundPrice	Стоимость билета для направления «туда»			
outboundTaxes	Тарифы и сборы билета для направления «туда»			
returnPrice	Стоимость билета для направления «обратно»			
returnTaxes	Тарифы и сборы билета для направления «обратно»			
outboundSeatsLeft	Количество оставшихся мест для данной рекомендации для направления «туда»			
returnSeatsLeft	Количество оставшихся мест для данной рекомендации для направления «обратно»			
totalPrice	Итоговая стоимость билета с учётом применённых скидков.			
<b>outboundFareAndAmountDetails</b>	Массив объектов fareDetails, содержащий подробную информацию о тарифах для направления «туда»			

	<b>returnFareAndAmountDetails</b>	Массив объектов fareDetails, содержащий подробную информацию о тарифах для направления «обратно»			
	<b>outboundBaggageAllowances</b>	Массив объектов, содержащий подробную информацию о нормах багажа для направления «туда»			
	<b>returnBaggageAllowances</b>	Массив объектов, содержащий подробную информацию о нормах багажа для направления «обратно»			
	<b>discount</b>	Информация о скидке, доступной для данного перелёта.  Скидка может отсутствовать для одного или нескольких перелётов, если рейсы находятся в исключении. Исключения хранятся на стороне CRM и IBE.			
	id	ID кампании скидок в CRM			
	name	Название кампании скидок			
	amount	Размер скидки			
	discountType	Тип скидки, возможные значения:  0 – Percent, проценты  1 – Roubles, рубли  Во избежание проблем рекомендуется проверять тип скидки в связи с её значением. Например, для типа 0 размер должен быть в интервале [0;100].			
	usageType	Тип применения скидки, возможные значения:  0 – Fare, тариф  1 – Order, весь заказ			
	startDate	Дата начала действия скидки			
	endDate	Дата окончания действия скидки			
	<b>promotion</b>	Информация о примененном промокоде для данного перелета.			

		number	Промокод			
		amount	Размер скидки по промокоду			
		promotionStatus	Статус промокода Возможные значение: справочник статусов промокодов			
		promotionType	Тип промокода 0 – промокод на перелёт 1 – промокод на дополнительную услугу			
		description	Описание кампании промокодов			
		startFlightDate	Дата начала периода полетов по промокоду			
		endFlightDate	Дата окончания периода полетов по промокоду			
		startDate	Дата начала периода действия промокода			
		endDate	Дата окончания периода действия промокода			
		isUnique	Признак уникальности промокода			
		campaignId	ID кампании промокодов.			
		<b>directions</b>	Направления исключений/включений			
		origin	Аэропорт вылета			
		destination	Аэропорт прилёта			
		<b>services</b>	Массив объектов с информацией об услугах, на которые действует промокод.			
		code	Код услуги Возможные значения: справочник доп. услуг (кроме страховки)			
		<b>excludedFlights</b>	Общие исключения – рейсы, на которые не действуют скидки			
		flightNumber	Номер рейса			
		departureAirport	Аэропорт вылета			
		Аэропорт вылета	Аэропорт прилёта			

		hasConflict	Признак успешности добавления промокода в бронирование.  False, если при добавлении промокода возникли ошибки.			
<b>errors</b>						
		ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE			
		description	Описание ошибки, см. Приложение 1. Коды ошибок IBE			
		<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.			
		code	Код ошибки Amadeus			
		text	Текст ошибки Amadeus			

outboundFareAndAmountDetails

## Объект flightGroup

Содержит информацию о перелётах

<b>data.outboundFlights</b> или <b>data.returnFlights</b>	
flightGroupId	Порядковый номер перелёта, используется в рекомендациях
flightGroupDuration	Общее время перелёта: складывается из значений длительности перелёта каждого из сегментов и длительности остановок.
hash	Служебное поле
<b>flights</b>	Массив с информацией о сегментах перелётов
flightId	Номер сегмента
departureAirport	Аэропорт отправления
departureTerminal	Терминал в аэропорте отправления
arrivalAirport	Аэропорт прибытия

arrivalTerminal	Терминал в аэропорте прибытия
departureDateTime	Дата и время отправления Формат: уууу-ММ-ддТНН:мм:сс Пример: 2018-05-27T17:25:00
excludedNationalities	Массив с информацией о национальностях, запрещенных к перевозке на данном рейсе Пример: ISR (для рейсов 3837 и 3838)
arrivalDateTime	Дата и время прибытия Формат: уууу-ММ-ддТНН:мм:сс Пример: 2018-05-27T20:50:00
aircraft	Полное название самолёта Например: Airbus A320
aircraftIataCode	IATA код воздушного судна
flightDuration	Длительность перелёта для текущего сегмента
flightDurationUntilNextStop	Время полёта до первой технической остановки. См. примеры в Приложение 3. Примеры ответов от сервисов
stopDuration	Время на посадку
flightNumber	Номер рейса
marketingCarrier	Маркетинговая авиакомпания Например: U6, S7
operatingCarrier	Авиакомпания, исполняющая перелёт Например: U6, ОК

	codeshareName	<p>Полное название авиакомпании, которая выполняет кодшер рейс.</p> <p>Например: Czech Airlines для operatingCarrier = ОК</p> <p>Для operatingCarrier = U6 данное поле не заполняется</p>
	hash	Служебное поле
	<b>techStops</b>	Массив с информацией о технических остановках
	arrivalDateTime	<p>Дата и время прибытия в аэропорт, где выполняется техническая остановка</p> <p>Формат: уууу-ММ-ддТНН:мм:сс</p> <p>Пример: 2018-05-26T05:30:00</p>
	departureDateTime	<p>Дата и время отправления из аэропорта, где выполняется техническая остановка</p> <p>Формат: уууу-ММ-ддТНН:мм:сс</p> <p>Пример: 2018-05-26T08:00:00</p>
	airportCode	Код аэропорта остановки
	stopDuration	Длительность остановки
	flightDurationUntilNextStop	<p>Длительность перелёта до следующей остановки</p> <p>См. примеры в Приложение 3. Примеры ответов от сервисов</p>

### Объект fareFamiliesOptions

Содержит информацию об опциях тарифа. Используется для отображения опций тарифов на сайте движка бронирования.

<b>data.fareFamilies.options</b>
----------------------------------

code	Код опции тарифа Например: upTo5bonuses, refundExchangeProhibited, mealsOnBoard и др.
category	Код категории для опции Например: bonus, ffServices и др.
order	Порядок для вывода опции
showInPopup	Показывать ли опции во всплывающем окне «ваши возможности».
showInSummary	Показывать ли опции на странице Summary.
showInUpsell	Показывать ли опции в информации о повышении тарифа.
showInFareIncluded	Показывать ли опции при выборе фильтра.
showForAdult	Показывать ли опции для взрослого пассажира.
showForInfant	Показывать ли опции для младенца.

## Объект fareDetails

Содержит информацию о деталях тарифа

<b>data.recommendations.outboundFareAndAmountDetails</b> или <b>data.recommendations.returnFareAndAmountDetails</b>	
bookingCode	В общем случае это первый символ fareBasis
fareBasis	Название базового тарифа Например: WPROW или WECOW
isBreakpoint	Признак того, заканчивает ли тариф своё действие на текущем сегменте.
passType	Тип пассажира. Возможные значения: справочник типов пассажиров
passCount	Количество пассажиров определённого типа
taxAmount	Цена тарифов и сборов
totalAmount	Полная стоимость билета, включая тарифы и сборы



isSale	Параметр, определяющий распродажный тариф.
--------	--

## Объект BaggageAllowances

Содержит информацию о нормах багажа

<b>data.recommendations.outboundBaggageAllowances или data.recommendations.returnBaggageAllowances</b>	
quantity	количество багажа
weight	вес багажа
passengerType	тип пассажира

## Примеры значений поля totalPrice

*Перелёт в одну сторону, скидок нет:*

Значение totalPrice равно значению outboundPrice.

```

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

```

*Перелёт в одну сторону, есть скидка:*

Значение totalPrice считается следующим образом:

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

[totalAmount – taxAmount] – нужно выполнить для всех подтарифов и типов пассажиров.

Floor – округление вниз.

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

```

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

```

*Перелёт в обе стороны, нет скидок:*

$totalPrice = outboundPrice + returnPrice$

$totalPrice = 6727 + 8051 = 14778$

```

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

```

*Перелёт в обе стороны, есть скидка:*

Значение totalPrice считается следующим образом:

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

[totalAmount – taxAmount] – нужно выполнить для всех подтарифов и типов пассажиров для направлений «туда» и «обратно».

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

```

▼47: {outboundFlightGroupId: 1, outboundFareFamilyId: 2, returnFlightGroupId: 1, returnFareFamilyId: 1,-}
  ▶ discount: {id: "1-1PSJTS1", amount: 4, discountType: 0, usageType: 0}
  ▼ outboundFareDetails: [{bookingCode: "I", fareBasis: "IBLOW", isBreakpoint: true,-}]
    ▼ 0: {bookingCode: "I", fareBasis: "IBLOW", isBreakpoint: true,-}
      bookingCode: "I"
      ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2423, totalAmount: 15308}]
        fareBasis: "IBLOW"
        isBreakpoint: true
      ▼ 1: {bookingCode: "Q", fareBasis: "QFLOW", isBreakpoint: true,-}
        bookingCode: "Q"
        ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2404, totalAmount: 19164}]
          fareBasis: "QFLOW"
          isBreakpoint: true
      outboundFareFamilyId: 2
      outboundFlightGroupId: 1
      outboundPrice: 34472
      outboundSeatsLeft: 2
      outboundTaxes: 4827
    ▼ returnFareDetails: [{bookingCode: "A", fareBasis: "AECRT", isBreakpoint: true,-}]
      ▼ 0: {bookingCode: "A", fareBasis: "AECRT", isBreakpoint: true,-}
        bookingCode: "A"
        ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2404, totalAmount: 4084}]
          fareBasis: "AECRT"
          isBreakpoint: true
      ▼ 1: {bookingCode: "W", fareBasis: "WECRT", isBreakpoint: true,-}
        bookingCode: "W"
        ▶ fareAmounts: [{paxType: 0, paxCount: 1, taxAmount: 2423, totalAmount: 3308}]
          fareBasis: "WECRT"
          isBreakpoint: true
      returnFareFamilyId: 1
      returnFlightGroupId: 1
      returnPrice: 7392
      returnSeatsLeft: 9
      returnTaxes: 4827
      totalPrice: 40576

```

## Сервис управления бронированием

### Сервис управления бронированием v2.2

<b>Объект</b> <b>fullReservationFlightСервис</b> <b>API</b>	reservation
<b>Краткое описание</b>	Сервис позволяет управлять бронированием: - добавлять информацию в бронирование (данные о перелётах, пассажира) - получать информацию о бронировании; - удалять бронирование.
<b>Поддерживаемые</b> <b>HTTP(S)-методы</b>	GET, POST, DELETE

<b>Используемые методы смежных систем</b>	<p><b>Amadeus:</b></p> <p>Создание/обновление бронирования - PNR_AddMultiElements, Fare_PricePNRWithBookingClass, Ticket_CreateTS</p> <p>Получение информации по бронированию - PNR_Retrieve, Ticket_Retrieve</p> <p>Удаление бронирования - PNR_Retrieve, PNR_Cancel, PNR_AddMultiElem</p>
---	---

#### Метод POST reservation

**Описание:** создание бронирования, добавление информации о перелётах, пассажирах и тарифах.

Возможно 2 варианта создания бронирования:

1. Предварительное бронирование, когда тариф с выбранной ценой закрепляется за сессией пользователя (сессия Amadeus переходит в состояние Stateful, её не могут получить другие пользователи системы).
2. Обычное бронирование, когда тариф с выбранной ценой не закрепляется за пользователем, так как количество доступных Stateful-сессий превысило некоторое предустановленное значение. В данном случае возможен конфликт при покупке клиентами последнего билета по определённому тарифу: билет будет отображаться у всех пользователей, но непосредственное бронирование будет выполнено только для того клиента, который раньше других отправит данные пассажира. Другие клиенты могут получить ошибку о том, что выбранный ими тариф больше недоступен.

На сайте движка бронирования на данный момент используется следующий алгоритм создания бронирования:

1. Отправить запрос на создание предварительного бронирования (только данные о перелёте и количество пассажиров).
2. Если ответ успешный, значит создалось предварительное бронирование и на следующем шаге нужно будет добавить данные о пассажирах и тарифах к бронированию.
3. Если в ответ пришла ошибка с кодом 1003, значит предварительное бронирование не создалось и на следующем шаге бронирования необходимо будет добавить полную информацию о перелёте и пассажирах.

**Формат запроса** /api/apiVersion/reservation

**Тело запроса**

```
{
  "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": {
    "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
},
"expectedPrice": integer,
"searchPrice": integer,
"airShopSessionId": string
}

```

## Объект reservationFlight

```

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

```

## Объект passengerData

```
{
  "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
  }
}
```

## Параметры запроса

Параметр	Описание
<b>flights</b>	Объект с информацией о перелётах
outboundFlights	Массив объектов reservationFlight с информацией о перелётах в направлении «туда»
returnFlights	Массив объектов reservationFlight с информацией о перелётах в направлении «обратно»
adults, children, infants, pensioners, youths, subsidizedChildren	Количество пассажиров каждого типа
origin	Аэропорта отправления
destination	Аэропорта назначения
passengersCount	Общее количество пассажиров

<b>passengers</b>	Объект с информацией о пассажирах
passengers	Массив объектов passengerData с информацией о пассажирах
userAgreement	Признак отправки согласия на информационную рассылку. Согласие отправляется в CRM для всех добавленных пассажиров.
<b>fares</b>	Объект с информацией о тарифах
<b>outboundFareFamily</b>	Объект с информацией о тарифах для направления "туда"
code	Код семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.code
category	Категория семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.category
isSubsidized	Признак субсидированного тарифа.
<b>returnFareFamily</b>	Объект с информацией о тарифах для направления "обратно"
code	Код семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.code
category	Категория семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.category
isSubsidized	Признак субсидированного тарифа.
<b>fares</b>	Массив с информацией о тарифах для направлений "туда" и "обратно"
fareBasis	Название базового тарифа. Данные берутся из ответа api/flights: data.recommendations.outboundFareDetails.fareBasis и data.recommendations.returnFareDetails.fareBasis

	isBreakpoint	<p>Признак того, заканчивает ли тариф своё действие на текущем сегменте.</p> <p>Данные берутся из ответа api/flights: data.recommendations.outboundFareDetails.isBreakpoint и data.recommendations.returnFareDetails.isBreakpoint</p>
	passengerType	<p>Тип пассажира.</p> <p>Возможные типы приведены в справочнике "Типы пассажиров" в Приложении 2.</p>
	flightDirection	<p>Направление перелёта. 0 - туда, 1 - обратно</p>
	flightIndex	Номер сегмента для конкретного перелёта.
	sessionMarker	Маркер сессии Aviasales
	expectedPrice	Ожидаемая цена бронирования, берётся из ответа api/flights
	searchPrice	Цена, которая была получена в поисковом запросе, берётся из api/flights или api/flights/search
	airShopSessionId	ID сессии поиска в системе SIG, берётся из ответа api/flights

### reservationFlight

Параметр	Описание
origin	Аэропорт отправления
destination	Аэропорт прибытия



departureDate	<p>Дата и время отправления.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.outboundFlights.flights.departureDateTime</p> <p>или</p> <p>data.returnFlights.flights.departureDateTime</p> <p>Формат: уууу-ММ-ддТНН:мм:ss</p> <p>Пример: 2018-06-20T01:00:00</p>
flightNumber	<p>Номер рейса.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.outboundFlights.flights.flightNumber</p> <p>или</p> <p>data.returnFlights.flights.flightNumber</p>
classOfService	<p>Информация о выбранном тарифе.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.recommendations.outboundFareDetails.bookingCode</p> <p>или</p> <p>data.recommendations.returnFareDetails.bookingCode</p>

### Объект passengerData Request

Параметр	Описание	Формат
id	Порядковый номер пассажира (нумерация с 1)	
firstName	Имя пассажира	[A-Z], 2-20 chars
middleName	Отчество пассажира (может отсутствовать)	[A-Z], 2-20 chars

surname	Фамилия пассажира	[A-Z], 2-20 chars
passengerType	Тип пассажира. Возможные значения: справочник типов пассажиров	
birthDate	Дата рождения	
sex	Пол пассажира Возможные значения: справочник "пол пассажиров"	
nationality	Национальность (страна) Возможные значения: справочник стран	
documentNumber	Номер документа	
documentType	Тип документа Возможные значения: справочник типов документов	
documentExpiryDate	Дата, до которой действует документ Обязательно только для заграничного паспорта. Для остальных документов автоматически устанавливается +10 лет от текущей даты.	
loyaltyCard	Номер карты лояльности «Крылья»	
loyaltyCardValidTo	Срок действия карты лояльности "Крылья" (передается только, если serviceClassUpOrdered=true)	
serviceClassUpOrdered	Флаг, обозначающий, что владелец золотой карты "Крылья" хочет повысить класс обслуживания	
email	Электронная почта	
phone	Телефон	
<b>disability</b>	Сведения о людях с ограниченными возможностями	

	isDeaf	Нарушение слуха	
	isBlind	Нарушение зрения	
	hasGuideDog	Необходимо сопровождение собаки-поводыря, должно отправляться только вместе с признаком isBlind	
	assistanceType	Тип сопровождения пассажира Возможные значения: справочник типов сопровождения	
	wheelchairType	Тип инвалидного кресла Возможные значения: справочник типов инвалидных кресел	
	commentary	Комментарий (не обязателен)	

## Формат ответа

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

```

        "documentNumber": string,
        "documentExpiryDate": datetime,
        "email": string,
        "phone": string,
        "documentType": integer,
        "isAdult": boolean,
        "isChanged": boolean
    }
],
"services": [],
"seats": [],
"insurance": {},
"partnerProducts": [],
    "ticketFares": [{
        "baseAmount": integer,
        "passengerReferences": [{
            "referenceNumber": string,
            "referenceQualifier": string
        }],
        "passengerType": integer,
        "referenceNumber": string
        "referenceQualifier": string
        "totalAmount": 10964
    }],
    "ticketFaresWithoutZapOff": [{
        "baseAmount": integer,
        "passengerReferences": [{
            "referenceNumber": string,
            "referenceQualifier": string
        }],
        "passengerType": integer,
        "referenceNumber": string
        "referenceQualifier": string
        "totalAmount": 10964
    }],
"hasSpecialConditions": boolean,
"isRefundAvailable": boolean,
"annulUnavailabilityReason": integer,
"isAnnulAvailable": boolean
},
"errors": [{
    "ibeCode": string,
    "description": string,
    "messages": [{
        "code": string,
        "text": 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
}

```

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса.  Если массив ошибок errors пуст, то значение поля будет true.
<b>data</b>	Данные, которые вернул запрос.
number	Уникальный номер бронирования (PNR).

<b>reservationDate</b>	<p>Дата и время создания бронирования</p> <p>Формат: уууу-ММ-ддТНН:мм:ss</p> <p>Пример: 2018-06-20T15:44:00</p>
<b>passengers</b>	<p>Массив объектов passengerData с информацией о пассажирах</p>
<b>services</b>	<p>Массив объектов с информацией о дополнительных сервисах.</p> <p>Заполняется при использовании сервиса api/services и будет описан ниже</p>
<b>seats</b>	<p>Массив объектов с информацией о выбранных местах.</p> <p>Заполняется при использовании сервиса api/services и будет описан ниже</p>
<b>insurance</b>	<p>Объект с информацией о выбранной страховке.</p> <p>Заполняется при использовании сервиса api/services и будет описан ниже</p>
<b>partnerProducts</b>	<p>Массив объектов с информацией о партнёрских услугах.</p> <p>Заполняется при использовании сервиса api/services и будет описан ниже</p>

<b>journey</b>	Информация о перелётах для данного бронирования
outboundFlights	Массив объектов fullReservationFlight с подробной информацией о перелётах «туда»
outboundFareFamily	Служебное поле, содержит код fareFamily
returnFlights	Массив объектов fullReservationFlight с подробной информацией о перелётах «обратно»
origin	Изначальный аэропорт вылета
destination	Изначальный аэропорт прилёта
departureDate	Дата вылета
returnDate	Дата возвращения
<b>ticketFares</b>	Массив объектов с информацией о тарифах после применения скидки
baseAmount	Размер базового тарифа
<b>passengerReferences</b>	Массив объектов со ссылками на пассажиров
referenceNumber	Номер ссылки
referenceQualifier	Тип ссылки, например "РА"
passengerType	Тип пассажира. См. справочник "типы пассажиров".

	referenceNumber	Номер ссылки
	referenceQualifier	Тип ссылки, например, "TST"
	totalAmount	Итоговая сумма тарифа
	<b>ticketFaresWithoutZapOff</b>	Массив объектов с информацией о тарифах без применения скидки
	baseAmount	Размер базового тарифа
	<b>passengerReferences</b>	Массив объектов со ссылками на пассажиров
	referenceNumber	Номер ссылки
	referenceQualifier	Тип ссылки, например "РА"
	passengerType	Тип пассажира. См. справочник "типы пассажиров".
	referenceNumber	Номер ссылки
	referenceQualifier	Тип ссылки, например, "TST"
	totalAmount	Итоговая сумма тарифа
	hasSpecialConditions	Признак наличия дополнительных условий
	isRefundAvailable	Признак доступности осуществления возврата
	annulUnavailabilityReason	Причина недоступности осуществления аннулирования бронирования



	isAnnulAvailable	Признак доступности осуществления аннулирования бронирования
	<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

### fullReservationFlight

Параметр	Описание
aircraftIataCode	IATA код воздушного судна
referenceNumber	Ссылка на текущий сегмент перелёта
origin	Аэропорт вылета для текущего сегмента
destination	Аэропорт прилёта для текущего сегмента
departureDate	Дата и время отправления для текущего сегмента
arrivalDate	Дата и время прибытия для текущего сегмента

arrivalDateUtc	Дата и время прибытия для текущего сегмента в UTC
departureDateUtc	Дата и время отправления для текущего сегмента в UTC
flightNumber	Номер рейса
classOfService	Информация о выбранном тарифе
marketingCarrier	Маркетинговая авиакомпания Например: U6, S7
operatingCarrier	Авиакомпания, по факту исполняющая перелёт Например: U6, ОК
<b>techStops</b>	Массив с информацией о технических посадках.
arrivalDateTime	Дата и время прибытия в пункт посадки
departureDateTime	Дата и время отправления из пункта посадки
airportCode	Код аэропорта остановки
stopDuration	Длительность остановки
flightDurationUntilNextStop	Длительность перелёта до следующей остановки
flightDurationUntilNextStop	Длительность перелёта до следующей остановки
arrivalTerminal	Номер терминала в пункте прибытия
departureTerminal	Номер терминала в пункте отправления
hash	Уникальный хеш перелета

### Объект passengerData Response

Параметр	Описание
id	Порядковый номер пассажира (нумерация с 1)
hasInfant	Признак наличия у данного пассажира младенца
referenceNumber	Ссылка на данного пассажира в бронировании
firstName	Имя пассажира
middleName	Отчество пассажира (может отсутствовать)
surname	Фамилия пассажира

passengerType	Тип пассажира. Возможные значения: справочник типов пассажиров
isAdult	Признак является ли пассажир взрослым true для пассажиров типа ADT/SRC/YTH false для пассажиров типа CHD/INF
birthDate	Дата рождения
sex	Пол пассажира Возможные значения: справочник "пол пассажиров"
nationality	Национальность (страна) Возможные значения: справочник стран
documentNumber	Номер документа
documentType	Тип документа Возможные значения: справочник типов документов
documentExpiryDate	Дата, до которой действует документ Обязательно только для заграничного паспорта. Для остальных документов автоматически устанавливается +10 лет от текущей даты.
email	Электронная почта
phone	Телефон

### Примеры успешных запросов и ответов

См. в разделе Приложение X. Примеры ответов от сервисов.

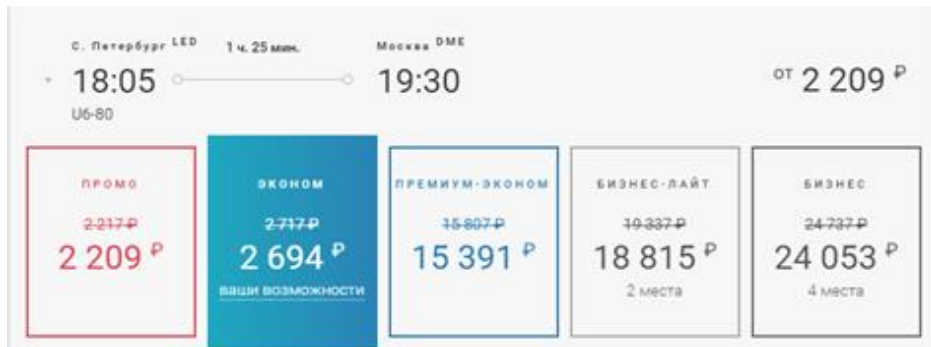
### Правила валидации при создании бронирования для субсидированных тарифов

№	Правило валидации	Код ошибки	Описание ошибки бронирования
---	-------------------	------------	------------------------------

1	Субсидированный тариф доступен только для бронирования типам пассажиров SRC / YTH / CNN	1008	Only subsidized types of passengers (SRC/YTH/CNN) could be used
2	Нельзя добавить в бронирование одновременно субсидированный и обычный тип пассажира	1008	Subsidized passengers found. All passengers in request must have subsidies.
3	Субсидированный тариф можно забронировать только гражданам России	1008	Only passenger with Russian citizenship can use subsidized tariffs
4	Возрастные ограничения: от 12 до 23 лет для типа YTH, от 55 лет женщины и от 60 лет мужчины для типа SRC.	1008	Passenger age is incorrect

## Примеры бронирований

1. Создание бронирования в одну сторону, без пересадок.



Берём из ответа api/flights конкретную рекомендацию: data -> recommendations:

<p>Рекомендация:</p> <pre>{   "outboundFlightGroupId": 0,   "outboundFareFamilyId": 1,   "outboundPrice": 2717.0,   "outboundTaxes": 1922.0,   "outboundSeatsLeft": 9,   "outboundFareDetails": [{     "bookingCode": "W",     "fareBasis": "WEWOW",     "isBreakpoint": true,     "fareAmounts": [{       "paxType": 0,       "paxCount": 1,       "taxAmount": 1922.00,       "totalAmount": 2717.00     }   ] }</pre>	<p>По полю outboundFlightGroupId определяем номер перелёта, который нужно будет добавить и берём информацию по нему из ответа api/flights, data -&gt; outboundFlights:</p> <pre>{   "flightGroupId": 0,   "flights": [{     "flightId": 0,     "departureAirport": "LED",     "departureTerminal": "1",     "arrivalAirport": "DME",     "departureDateTime": "2018-12-08T18:05:00",     "arrivalDateTime": "2018-12-08T19:30:00",     "aircraft": "Airbus A319",     "flightDuration": "1:25",     "flightNumber": "80",     "marketingCarrier": "U6",     "operatingCarrier": "U6",     "hash": "-1547914441"   } ],   "flightGroupDuration": "1:25",   "hash": "-1643139679" }</pre>
--	---

<pre>"id": "1-1PSJTS5", "amount": 3, "discountType": 0, "usageType": 0 }, "totalPrice": 2694.0 }</pre>	<p><b>Подставляем выделенные значения, для массива flights получаем:</b></p> <pre>"flights": { "outboundFlights": [{ "origin": "LED", "destination": "DME", "departureDate": "2018-12-08T18:05:00.000Z", "flightNumber": "80", "classOfService": "W" } ], "returnFlights": null, "passengersCount": 1 }</pre> <p>Значение classOfService совпадает с bookingCode из рекомендации.</p> <p>Значение passengersCount зависит от количества добавляемых в бронирование пассажиров.</p> <p>По полю outboundFareFamilyId определяем номер тарифа, который нужно будет добавить и берём информацию по нему из ответа api/flights, data -&gt; fareFamilies:</p> <pre>{ "fareFamilyId": 1,</pre>
--	---

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

**Подставляем выделенные значения, для массива fares получаем:**

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

2. Создание бронирования в обе стороны, одна пересадка на сегменте «обратно».

Направление туда

Екатеринбург SVX 2 ч. 35 мин. Москва DME

18:40 19:15 от 3 636 Р

U6-266

Категория	Цена (Р)	Возможности
ПРОМО	3 636 Р	ВАШИ ВОЗМОЖНОСТИ
ЭКОНОМ	3 736 Р	
ПРЕМИУМ-ЭКОНОМ	26 321 Р	
БИЗНЕС	25 881 Р	8 мест

Направление обратно

Москва DME 8 ч. 20 мин. Екатеринбург SVX

10:10 20:30 от 9 803 Р

U6-79 1 пересадка, U6-624

Категория	Цена (Р)	Возможности
ПРОМО	9 803 Р	7 мест
ЭКОНОМ	12 603 Р	ВАШИ ВОЗМОЖНОСТИ
ПРЕМИУМ-ЭКОНОМ	38 203 Р	
БИЗНЕС	51 448 Р	н/доступно



<p>Рекомендация:</p> <pre>{   "outboundFlightGroupId": 0,   "outboundFareFamilyId": 0,   "returnFlightGroupId": 6,   "returnFareFamilyId": 1,   "outboundPrice": 3636.0,   "outboundTaxes": 2321.0,   "returnPrice": 12603.0,   "returnTaxes": 4318.0,   "outboundSeatsLeft": 9,   "returnSeatsLeft": 7,   "outboundFareDetails": [{     "bookingCode": "O",     "fareBasis": "OPROW",     "isBreakpoint": true,     "fareAmounts": [{       "paxType": 0,       "paxCount": 1,       "taxAmount": 2321.00,       "totalAmount": 3636.00</pre>	<p>По полям outboundFlightGroupId и returnFlightGroupId определяем номера перелётов и берём информацию по ним из ответа api/flights, data -&gt; outboundFlights и data -&gt; returnFlights соответственно.</p> <p>Перелёт туда:</p> <pre>{   "flightGroupId": 0,   "flights": [{     "flightId": 0,     "departureAirport": "SVX",     "arrivalAirport": "DME",     "departureDateTime": "2018-12-01T18:40:00",     "arrivalDateTime": "2018-12-01T19:15:00",     "aircraft": "Airbus A320",     "flightDuration": "2:35",     "flightNumber": "266",     "marketingCarrier": "U6",     "operatingCarrier": "U6",     "hash": "1861162940"   } ],   "flightGroupDuration": "2:35",   "hash": "-1144973042"</pre>
--	--

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

<pre> "taxAmount": 2393.00, "totalAmount": 8343.00 } ] } ], "totalPrice": 16239.0 } </pre>	<pre> "departureTerminal": "1", "arrivalAirport": "SVX", "departureDateTime": "2019-01-10T15:40:00", "arrivalDateTime": "2019-01-10T20:30:00", "aircraft": "Airbus A319", "flightDuration": "2:50", "flightNumber": "624", "marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "-375095535" } ], "flightGroupDuration": "8:20", "hash": "327592200" } </pre> <p><b>Подставляем выделенные значения, для массива flights получаем:</b></p> <pre> { "flights": { "outboundFlights": [{ "origin": "SVX", "destination": "DME", "departureDate": "2018-12-01T18:40:00.000Z", </pre>
--	---

	<pre>"flightNumber": "266",  "classOfService": "O"  }  ],  "returnFlights": [{    "origin": "DME",    "destination": "LED",    "departureDate": "2019-01-10T10:10:00.000Z",    "flightNumber": "79",    "classOfService": "N"  }, {    "origin": "LED",    "destination": "SVX",    "departureDate": "2019-01-10T15:40:00.000Z",    "flightNumber": "624",    "classOfService": "P"  }  ],  "passengersCount": 1  }  }</pre> <p>Значение classOfService совпадает с bookingCode из рекомендации.</p>
--	--

Значение passengersCount зависит от количества добавляемых в бронирование пассажиров.

По полям outboundFareFamilyId и returnFareFamilyId определяем номера тарифов, которые нужно будет добавить и берём информацию по ним из ответа api/flights, data -> fareFamilies:

**Подставляем выделенные значения, для массива fares получаем:**

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

	<pre>"isBreakpoint": true } ] }</pre> <p>Значения fareBasis берём из рекомендации.</p>
--	--

### Пример создания бронирования для субсидированных пассажиров

3. Создание бронирования для состава пассажиров: 1 SRC + 1 YTH + 1 CNN (типы пассажиров описаны в приложении 2).

Перелёт туда

Перелёт обратно

Найдём рекомендацию для выбранного перелёта (в ответе api/flights)

```

▼ 54: {outboundFlightGroupId: 6, outboundFareFamilyId: 1, returnFlightGroupId: 2, returnFareFamilyId: 0,...}
  ▼ outboundFareAndAmountDetails: [...]
    ▼ 0: {bookingCode: "R", fareBasis: "RBDRTCD", isBreakpoint: true, paxType: 1, paxCount: 1, taxAmount: 0,...}
      bookingCode: "R"
      fareBasis: "RBDRTCD"
      flightIndex: 0
      isBreakpoint: true
      paxCount: 1
      paxType: 1
      taxAmount: 0
      totalAmount: 6000
    ▶ 1: {bookingCode: "R", fareBasis: "RBDRTZZ", isBreakpoint: true, paxType: 2, paxCount: 1, taxAmount: 0,...}
    ▶ 2: {bookingCode: "R", fareBasis: "RBDRTCH", isBreakpoint: true, paxType: 5, paxCount: 1, taxAmount: 0,...}
    outboundFareFamilyId: 1
    outboundFlightGroupId: 6
    outboundPrice: 16500
    outboundSeatsLeft: 9
    outboundTaxes: 0
  ▼ returnFareAndAmountDetails: [...]
    ▼ 0: {bookingCode: "E", fareBasis: "EPROW", isBreakpoint: true, paxType: 1, paxCount: 1, taxAmount: 2764,...}
      bookingCode: "E"
      fareBasis: "EPROW"
      flightIndex: 0
      isBreakpoint: true
      paxCount: 1
      paxType: 1
      taxAmount: 2764
      totalAmount: 4374
    ▶ 1: {bookingCode: "E", fareBasis: "EPRRT", isBreakpoint: true, paxType: 1, paxCount: 1, taxAmount: 2666,...}
    ▶ 2: {bookingCode: "E", fareBasis: "EPROW", isBreakpoint: true, paxType: 2, paxCount: 1, taxAmount: 2764,...}
    ▶ 3: {bookingCode: "E", fareBasis: "EPRRT", isBreakpoint: true, paxType: 2, paxCount: 1, taxAmount: 2666,...}
    ▶ 4: {bookingCode: "E", fareBasis: "EPROW", isBreakpoint: true, paxType: 5, paxCount: 1, taxAmount: 2632,...}
    ▶ 5: {bookingCode: "E", fareBasis: "EPRRT", isBreakpoint: true, paxType: 5, paxCount: 1, taxAmount: 2583,...}
    returnFareFamilyId: 0
    returnFlightGroupId: 2
    returnPrice: 29855
    returnSeatsLeft: 9
    returnTaxes: 16075
    totalPrice: 46355

```

Видим, что в массиве тарифов для направлений туда и обратно присутствует информация о типе пассажира, для которого должен использоваться тариф, и о номере сегмента перелёта.

*Примечание:* используется именно тип пассажира, а не ссылка на него, так как на данном этапе пассажиры ещё не добавлены в бронирование.

Информация о тарифах должна отправляться следующим образом:

```

"fares": {
  "outboundFareFamily": {
    "category": "SUBSIDIZED",
    "code": "SUBSIDIZED",
    "isSubsidized": true
  },
  "returnFareFamily": {
    "category": "ECOLIGHT",
    "code": "ECOLIGHTRU",
    "isSubsidized": true
  },
  "fares": [
    {
      "fareBasis": "RBDRTCD",

```

```

    "isBreakpoint": true,
    "passengerType": 1,
    "flightDirection": 0,
    "flightIndex": 0
  },
  {
    "fareBasis": "RBDRTZZ",
    "isBreakpoint": true,
    "passengerType": 2,
    "flightDirection": 0,
    "flightIndex": 0
  },
  {
    "fareBasis": "RBDRTCH",
    "isBreakpoint": true,
    "passengerType": 5,
    "flightDirection": 0,
    "flightIndex": 0
  },
  {
    "fareBasis": "EPROW",
    "isBreakpoint": true,
    "passengerType": 1,
    "flightDirection": 1,
    "flightIndex": 0
  },
  {
    "fareBasis": "EPRRT",
    "isBreakpoint": true,
    "passengerType": 1,
    "flightDirection": 1,
    "flightIndex": 1
  },
  {
    "fareBasis": "EPROW",
    "isBreakpoint": true,
    "passengerType": 2,
    "flightDirection": 1,
    "flightIndex": 0
  },
  {
    "fareBasis": "EPRRT",
    "isBreakpoint": true,
    "passengerType": 2,
    "flightDirection": 1,
    "flightIndex": 1
  },
  {
    "fareBasis": "EPROW",
    "isBreakpoint": true,

```



```

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

```

#### 4. Создание бронирования в одну сторону с одной посадкой.

Владивосток VVO 9 ч. 25 мин. Екатеринбург SVX

09:50 14:15

U6-9170 1 посадка

от 16 880 P

Категория	Цена	Дополнительно
ПРОМО	16 880 P	
ЭКОНОМ	17 880 P	
ПРЕМИУМ-ЭКОНОМ	53 445 P	ВАШИ ВОЗМОЖНОСТИ
БИЗНЕС	56 945 P	2 места

<p>Рекомендация:</p> <pre> {   "outboundFlightGroupId": 0,   "outboundFareFamilyId": 2,   "outboundPrice": 53445.0,   "outboundTaxes": 3445.0,   "outboundSeatsLeft": 5,   "outboundFareDetails": [{     "bookingCode": "Q",     "fareBasis": "QFLOW",     "isBreakpoint": true,     "fareAmounts": [{       "paxType": 0,       "paxCount": 1,       "taxAmount": 3445.00,       "totalAmount": 53445.00     }   ] } ], "totalPrice": 53445.0 } </pre>	<p>По полю outboundFlightGroupId определяем номер перелёта, который нужно будет добавить и берём информацию по нему из ответа api/flights, data -&gt; outboundFlights:</p> <pre> {   "flightGroupId": 0,   "flights": [{     "flightId": 0,     "departureAirport": "VVO",     "arrivalAirport": "SVX",     "departureDateTime": "2019-02-05T09:50:00",     "arrivalDateTime": "2019-02-05T14:15:00",     "aircraft": "Airbus A320",     "techStops": [{       "arrivalDateTime": "2019-02-05T13:05:00",       "departureDateTime": "2019-02-05T13:45:00",       "airportCode": "OVV",       "stopDuration": "0:40",       "flightDurationUntilNextStop": "2:30"     }   ] },   "flightDuration": "9:25",   "flightDurationUntilNextStop": "6:15",   "flightNumber": "9170", </pre>
---	---

	<pre>"marketingCarrier": "U6", "operatingCarrier": "U6", "hash": "1138974543" } ], "flightGroupDuration": "9:25", "hash": "-1673368260" } <b>Подставляем выделенные значения, для массива flights получаем:</b> { "flights": { "outboundFlights": [{ "origin": "VVO", "destination": "SVX", "departureDate": "2019-02-05T09:50:00.000Z", "flightNumber": "9170", "classOfService": "Q" }] }, "returnFlights": null, "passengersCount": 1 }</pre>
--	--

	<pre> }  Значение classOfService совпадает с bookingCode из рекомендации.  Значение passengersCount зависит от количества добавляемых в бронирование пассажиров.  По полю outboundFareFamilyId определяем номер тарифа, который нужно будет добавить и берём информацию по нему из ответа api/flights, data -&gt; fareFamilies:  {  "fareFamilyId": 2,  "code": "ECOPREMRU",  "shortCode": "FL",  "category": "ECOPREM",  "options": [ ... ],  "order": 3,  "commercialFamily": "U6ECONOMY"  }  <b>Подставляем выделенные значения, для массива fares получаем:</b>  "fares": {  "outboundFareFamily": {  "category": "ECOPREM",  "code": "ECOPREMRU"  },  "returnFareFamily": null, </pre>
--	---

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

#### Метод GET reservation

**Описание:** позволяет получить информацию об уже созданном бронировании.

**Формат запроса** /api/apiVersion/reservation

#### Примеры запросов

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

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

#### Параметры запроса

Параметр	Описание
pnrNumber	Номер PNR, для которого нужно получить информацию
ticketNumber	Номер выпущенного билета. Номер билета должен содержать 13 цифр, например: 2622438214047 или 2622438214046.
lastName	Фамилия одного из пассажиров, добавленных в бронирование. Параметр передаётся латиницей. ИБЕ вернёт ошибку, если фамилии нет в PNR.

*Примечание: при запросе информации о заказе по номеру билета, поля ticketNumber и lastName должны относиться к одному и тому же пассажиру.*

## Формат ответа

```
{
  "data": {
    "tickets": [{
      "number": string,
      "passengerReference": string,
      "isInfantTicket": boolean
    }
  ],
  "outboundFareFamily": fareFamilyObject,
  "returnFareFamily": fareFamilyObject,
  "tsmData": [{
    "passengerReference": string,
    "price": number,
    "referenceType": string,
    "referenceNumber": string
  }
  ],
  "appliedDiscounts": [{
    "flightReference": string,
    "amount": integer,
    "discountAmount": decimal,
    "id": string,
    "type": string,
    "passengerReference": string,
    "serviceCode": string,
    "promoNumber": string,
    "serviceReference": string,
    "usageType": integer,
    "promotionType": integer
  }
  ],
  "baseFarePrices": [{
    "currency": string,
    "passengerReference": string,
    "isInfantFare": boolean,
    "totalPrice": number,
    "basePrice": number,
    "priceHidden": boolean
  }
  ],
  "foodOnBoards": [{
    "departureAirport": string,
    "arrivalAirport": string,
    "flightNumber": string
    "commercialFamily": string,
    "ru": string,
    "en": string,
    "zh": string
  }
  ],
  "reservationRetrieveStatus": integer,
  "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
    }
],
"seats": [{
    "number": integer,
    "title": string,
    "code": string,
    "status": string,
    "passengerReference": string,
    "flightReference": string,
    "referenceNumber": string,
    "packReferenceNumber": string
}
],
"insurance": {
    "number": string,
    "code": string,
    "status": string,
    "referenceNumber": string,
    "packReferenceNumber": string
},
"partnerProducts": [{
    "partnerId": integer,
    "partnerType": integer,
    "code": string,
    "amount": number,
    "currency": string,
    "passengerReference": string,
    "isInfantProduct": boolean
}
],
"hasSpecialConditions": boolean
"errors": [{
    "ibeCode": integer,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
}
],
}
]
"success": boolean
}
}

```

## Объект fareFamily

```

{
    "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
}

```

### Объект fullReservationFlight

```

{
    "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
}

```

### Объект passenger

```

{
    "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
}
```

## Параметры ответа

Параметр		Описание
success		Признак успешности выполнения запроса.  Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>		Данные, которые вернул запрос
	<b>tickets</b>	Информация о выписанных билетах
	number	Номер билета.  <b>Пример:</b> 262 2439494143
	passengerReference	Ссылка на пассажира, к которому относится данный билет.
	isInfantTicket	Признак является ли билет билетом для младенца.
<b>outboundFareFamily</b>		Массив объектов farFamily с информацией о выбранных тарифах для перелёта «туда»
<b>returnFareFamily</b>		Массив объектов farFamily с информацией о выбранных тарифах для перелёта «обратно»

	<b>tsmData</b>	Массив объектов с информацией о дополнительных услугах, добавленных в PNR
	passengerReference	Ссылка на пассажира, к которому относится данная услуга
	price	Стоимость услуги
	referenceType	Тип ссылки на запись в PNR, обычно имеет значение «SSR».
	referenceNumber	Номер SSR. Каждой услуге соответствует уникальный номер SSR в PNR.
	<b>appliedDiscounts</b>	Применённые скидки в данном PNR.
	flightReference	Ссылка на сегмент, на который была применена скидка
	amount	Размер скидки в процентах
	discountAmount	Абсолютное значение скидки в рублях
	id	ID кампании в CRM, по которой была применена скидка

	type	<p>Тип скидки. Возможные значения:</p> <p>DISCOUNT – обычная скидка,</p> <p>PROMO – скидка по промокоду на перелет.</p> <p>PROMOCODE – скидка по промокоду на дополнительные услуги.</p>
	passengerReference	Ссылка на пассажира, для которого была применена скидка.
	serviceCode	Код добавленной услуги, на которую была применена скидка.
	promoNumber	Номер примененного промокода из CRM.
	serviceReference	Идентификатор услуги внутри текущего бронирования.
	usageType	Тип скидки: на тариф или на заказ.
	promotionType	Тип промокода: на перелет или на услугу.
	<b>baseFarePrices</b>	Массив с информацией о базовых стоимостях для тарифов.
	currency	<p>Валюта, в которой представлена стоимость.</p> <p>Пример значения: RUB</p>

	passengerReference	Ссылка на пассажира, для которого определяется стоимость.
	isInfantFare	Признак является ли пассажир младенцем
	totalPrice	Общая цена
	basePrice	Базовая цена (без тарифов и сборов)
	priceHidden	Признак, должна ли сумма тарифа скрываться для клиента. Будет в значении "true" для IT-тарифов.
	<b>foodOnBoards</b>	Массив с информацией о питании, доступном на рейсах
	departureAirport	Аэропорт вылета
	arrivalAirport	Аэропорт прилёта
	flightNumber	Номер рейса
	commercialFamily	Тип семейства тарифов (на одном и том же рейсе может быть доступно разное питание для эконома и бизнеса).
	ru	Перевод опции на русский язык
	en	Перевод опции на английский язык
	zh	Перевод опции на китайский язык

	<b>reservationRetrieveStatus</b>	<p>Статус получения бронирования.</p> <p>Может принимать значения:</p> <p>0 - Ok (успех)</p> <p>1 - TicketRequired (групповое бронирование, пассажир не указал номер билета)</p> <p>2 - InvalidTicketNumber (групповое бронирование, пассажир указал некорректный номер билета)</p>
	<b>engineType</b>	<p>Система, в которой создано бронирование.</p> <p>Может принимать значения:</p> <p>0 - Unknown,</p> <p>1 - Amadeus,</p> <p>2 - Tais</p>
	<b>isGroupReservation</b>	Флаг группового бронирования.
	<b>spentBonuses</b>	Объект с информацией об общем количестве потраченных бонусов
	amount	Количество бонусов
	passengerReference	Ссылка на первого пассажира

	<b>spentBonusesByPassengers</b>	Массив с информацией о количестве бонусов, потраченных конкретным пассажиром
	amount	Количество бонусов
	passengerReference	Ссылка на пассажира
	<b>appliedCertificates</b>	Массив с информацией о примененных сертификатах
	amount	Номинал сертификата
	number	Номер сертификата
	originOfficeId	Офис, под которым было создано бронирование.  Например, SVXU608AZ
	originOfficeType	Тип офиса, под которым было создано бронирование.  Возможные значения: справочник офисов
	<b>journey</b>	Информация о перелётах для данного бронирования
	<b>outboundFlights</b>	Массив объектов fullReservationFlight с подробной информацией о перелётах «туда»
	outboundFareFamily	Служебное поле с информацией о выбранном тарифе для перелёта «туда». Например, ECOFLEXRU

	returnFareFamily	Служебное поле с информацией о выбранном тарифе для перелёта «обратно»
	outboundDuration	Общая продолжительность перелёта «туда»
	<b>returnFlights</b>	Массив объектов fullReservationFlight с подробной информацией о перелётах «обратно»
	returnDuration	Общая продолжительность перелёта «обратно»
	origin	Изначальный аэропорт вылета
	destination	Изначальный аэропорт прилёта
	departureDate	Дата вылета
	returnDate	Дата возвращения
	<b>outboundBaggageAllowances</b>	Массив объектов, содержащий подробную информацию о нормах багажа для направления «туда»
	<b>returnBaggageAllowances</b>	Массив объектов, содержащий подробную информацию о нормах багажа для направления «обратно»
	number	Номер PNR
	reservationDate	Дата и время создания бронирования



	<b>passengers</b>	Массив объектов passengerData с информацией о пассажирах
	<b>services</b>	Массив объектов с информацией о добавленных в бронирование дополнительных услугах
	<b>code</b>	Код дополнительной услуги  Возможные значения: справочник кодов доп. услуг
	<b>status</b>	Статус добавления услуги.  Пример: НК – услуга успешно добавлена
	<b>text</b>	Текст в формате ssg для отправки информации по питомцу в Amadeus.  Пример: DOG 1PC 8KG 45X35X25
	<b>petType</b>	Тип питомца.  Возможные значения: DOG, CAT, PARROT, CANARY
	<b>category</b>	Категория дополнительной услуги  Возможные значение: справочник кодов доп. услуг
	<b>passengerReference</b>	Ссылка на пассажира, для которого добавлена дополнительная услуга

	packReferenceNumber	Ссылка на пакет услуг, в котором была добавлена услуга
	flightReference	Ссылка на сегмент, на котором действует услуга
	referenceNumber	Номер SSR. Каждой услуге соответствует уникальный номер SSR в PNR.
	price	Стоимость дополнительной услуги
	<b>seats</b>	Массив объектов с информацией о добавленных в бронирование местах
	number	Ряд, на котором выбрано место
	title	Место
	code	Код услуги «выбор мест». Значение: RQST
	status	Статус добавления услуги.  Пример: НК – услуга успешно добавлена
	passengerReference	Ссылка на пассажира, для которого добавлена дополнительная услуга
	flightReference	Ссылка на сегмент, на котором действует услуга
	referenceNumber	Номер SSR. Каждой услуге соответствует уникальный номер SSR в PNR.
	packReferenceNumber	Ссылка на пакет услуг, в котором было добавлено место

	<b>insurance</b>	Объект с информацией о добавленной в бронирование страховке
	number	Номер документа со страховкой
	code	Код страховки Возможные значения: справочник доп. услуг (категория insurance)
	status	Статус добавления услуги. Пример: НК – услуга успешно добавлена
	referenceNumber	Номер SSR. Каждой услуге соответствует уникальный номер SSR в PNR.
	packReferenceNumber	Ссылка на пакет услуг, в котором была добавлена услуга
	<b>partnerProducts</b>	Массив объектов с информацией о добавленных в бронирование партнёрских услугах
	partnerId	Уникальный ID партнёра в движке IBE. Список партнёров доступен в приложении 5
	partnerType	Уникальный код типа партнёра Список типов партнёров доступен в приложении 5

	code	<p>Код партнёрской услуги.</p> <p>Для каждого партнёра заведены уникальные коды услуг.</p> <p>См. коды партнёрских услуг в приложении 5.</p>
	amount	Стоимость партнёрской услуги.
	currency	Валюта тарификации
	passengerReference	Ссылка на пассажира, для которого добавлена услуга
	isInfantProduct	Признак услуги для младенца
	<b>hasSpecialConditions</b>	<p>Признак, что в бронирование добавлена одна из следующих услуг:</p> <p>PETC, CBVG, WEAP, UMR, MAAS, MEDA, EXST, STCR, WCHR, WCHC, WCHS, WCBG, WCBW, WCMR, WCOB, DEAF, BLND, DEPA, DEPU</p>
	<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>
	description	Описание ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>

<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

### fareFamily

Параметр	Описание
fareFamilyId	Порядковый номер семейства тарифов, используется в рекомендациях.
code	Код семейства, например:  ECOLIGHT/ECOFLEX/ECOPREM и др.  Возможные значения: см. справочник основных категорий тарифов
order	Порядок отображения семейства тарифов
commercialFamily	Тип семейства тарифов, например:  U6ECONOMY или U6BUSINESS
canRefund	Признак является ли тариф возвратным
hasUpsale	Признак показывает можно ли данный тариф "повысить" до следующего в ценовой категории.
<b>options</b>	Массив объектов с информацией об опциях тарифа.  Используется для сайта движка бронирования.

code	<p>Код опции тарифа</p> <p>Например:</p> <p>upTo5bonuses</p> <p>refundExchangeProhibited</p> <p>mealsOnBoard</p> <p>и др.</p> <p>Используется для сайта движка бронирования.</p>
category	<p>Код категории для опции</p> <p>Например:</p> <p>bonus, ffServices и др.</p> <p>Используется для сайта движка бронирования.</p>
order	Порядок для вывода опции
showInPopup	Показывать ли опцию во всплывающем окне «ваши возможности».
showInSummary	Показывать ли опцию на странице Summary.
showInUpsell	Показывать ли опцию в информации о повышении тарифа.
showInFareIncluded	Показывать ли опцию при выборе фильтра.
showForAdult	Показывать ли опцию для взрослого пассажира.
showForInfant	Показывать ли опцию для младенца.
showInBookingCart	Показывать ли опцию в корзине.

### fullReservationFlight

Параметр	Описание
aircraft	Номер борта
aircraftIataCode	IATA код воздушного судна
referenceNumber	Ссылка на текущий сегмент перелёта

origin	Аэропорт вылета для текущего сегмента
destination	Аэропорт прилёта для текущего сегмента
departureDate	Дата и время отправления для текущего сегмента
arrivalDate	Дата и время прибытия для текущего сегмента
flightNumber	Номер рейса
classOfService	Информация о выбранном тарифе
marketingCarrier	Маркетинговая авиакомпания Например: U6, S7
operatingCarrier	Авиакомпания, исполняющая перелёт Например: U6, ОК
flightDuration	Длительность перелёта для текущего сегмента

#### **outboundBaggageAllowances и returnBaggageAllowances**

<b>Параметр</b>	<b>Описание</b>
quantity	количество багажа
weight	вес багажа
passengerType	тип пассажира

#### **passengerData**

<b>Параметр</b>	<b>Описание</b>
id	Порядковый номер пассажира (нумерация с 1)
hasInfant	Признак наличия у данного пассажира младенца
referenceNumber	Ссылка на данного пассажира в бронировании
firstName	Имя пассажира
middleName	Отчество пассажира (может отсутствовать)
surname	Фамилия пассажира

passengerType	Тип пассажира. Возможные значения: справочник типов пассажиров
isAdult	Признак является ли пассажир взрослым true для пассажиров типа ADT/SRC/YTH false для пассажиров типа CHD/INF
birthDate	Дата рождения
sex	Пол пассажира Возможные значения: справочник пол пассажиров
nationality	Национальность (страна) Возможные значения: справочник стран
documentNumber	Номер документа
documentType	Тип документа Возможные значения: справочник типов документов
documentExpiryDate	Дата, до которой действует документ Обязательно только для заграничного паспорта. Для остальных документов автоматически устанавливается +10 лет от текущей даты.
loyaltyCard	Номер карты лояльности «Крылья»
email	Электронная почта
phone	Телефон

### **Примеры успешных запросов и ответов**

Приложение X. Примеры ответов от сервисов

### **Примеры неуспешных запросов и ответов**

В запросе использован несуществующий номер PNR



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

В запросе использована фамилия, которой нет в PNR

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

В запросе использована фамилия, которая не соответствует фамилии в запрашиваемом билете

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

В запросе передан номер билета, который не соответствует требованиям

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

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

#### Метод DELETE reservation

**Описание:** метод позволяет удалять созданное бронирование.

**Формат запроса** /api/apiVersion/reservation

#### Параметры запроса

Отсутствуют. Бронирование для отмены определяется по ID сессии, переданной в заголовке X-Session.

#### Формат ответа

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

#### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus

	text	Текст ошибки Amadeus
--	------	----------------------

## Примеры неуспешных запросов и ответов

Попытка удаления, если информация о бронировании не загружена в контекст.

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

## Сервис управления бронированием v2.3

### Изменения в версии 2.3:

В ответе сервиса GET reservation объект insurance был заменён на массив insurances.

### Формат ответа

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

```

## Параметры ответа (в кратком формате)

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
number	Номер PNR
reservationDate	Дата и время создания бронирования
bookingTimeLimit	Дата и время, до которого будет существовать неоплаченное бронирование (правила задаются настройками в ИВЕ)
<b>passengers</b>	Массив объектов passengerData с информацией о пассажирах
<b>services</b>	Массив объектов с информацией о дополнительных сервисах.
<b>seats</b>	Массив объектов с информацией о выбранных местах.
<b>insurances</b>	Массив объектов с информацией о выбранных страховках.
<b>partnerProducts</b>	Массив объектов с информацией о партнёрских услугах.
<b>journey</b>	Информация о перелётах для данного бронирования
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.

## Описание структуры insurances

Параметр	Описание
number	Номер страхового документа
passengerType	Тип пассажира
code	Код услуги. Например, "URFL"
status	Статус добавления услуги. Например, "НК"
passengerReference	Ссылка на пассажира, для которого добавлена услуга
price	Стоимость услуги

## Сервис получения информации о партнёрских услугах

Сервис API	reservation/partners
------------	----------------------

<b>Краткое описание</b>	Позволяет получить информацию по партнёрским услугам в бронировании и выполнить переотправку оплаченных услуг. Нужен в основном для получения ссылок на билеты/полисы партнёров.
<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> PNR_Retrieve, PNR_AddMultiElements <b>Аэроэкспресс:</b> getOrderTickets

### Метод GET `api/reservation/partners`

**Описание:** позволяет получить информацию о партнёрской услуге в бронировании

**Формат запроса** `/api/apiVersion/reservation/partners/{ { bookingId } }/details`

где `bookingId` - уникальный ID заказа партнёрской услуги в данном бронировании.

### Пример запроса

После добавления партнёрской услуги, например, Аэроэкспресса в бронирование в ответе `api/services` присутствует следующая структура:

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

Для того, чтобы получить информацию об этой услуге, необходимо отправить запрос:

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

### Формат ответа

```
{
```

```

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

```

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса.  Если массив ошибок errors пуст, то значение поля будет true.
<b>data</b>	Данные, которые вернул запрос.
bookingId	Уникальный ID партнёрской услуги в ИВЕ. Совпадает со значением из запроса.

	<b>description</b>	<p>Описание партнёрской услуги.</p> <p>Например, "СТАНДАРТ" для тарифа STANDARTOW Аэроэкспресса. Или "ТРЭВЕЛ" для партнёрской страховки АльфаСтрахование</p>
	<b>document</b>	ID заказа в партнёрской системе.
	<b>documentUrl</b>	Ссылка на билет/полис партнёра.
	<b>passengerReferences</b>	<p>Массив со ссылками на пассажиров, для которых добавлена услуга.</p> <p>Необязательное поле.</p>
	<b>flightReferences</b>	<p>Массив со ссылками на перелёты, для которых добавлена услуга.</p> <p>Необязательное поле.</p>
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	<b>ibeCode</b>	Код ошибки IBE.
	<b>description</b>	Описание ошибки IBE.
	<b>messages</b>	<p>Массив объектов, содержащий ошибки смежных систем.</p> <p>Массив может быть пуст, рекомендуется анализировать ошибку по полю <b>ibeCode</b>.</p>
	<b>code</b>	Код внутренней ошибки.



	text	Описание внутренней ошибки.
--	------	-----------------------------

### Пример ответа для партнера Аэроэкспресс

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

### Пример ответа для партнера АльфаСтрахование

```
{
  "data": {
    "bookingId": "b1bfb8b3-ebfe-4434-b97b-1e90298a360a",
    "description": "ТРЭВЕЛ",
    "document": "27514142",
    "documentUrl": "http://uat-tes.alfastrah.ru/travel-ext-services/reports?parameters=7E964978DD9AC0ED5D3165945FBED849D2C8D60734B455FA193B482A91E92FEC4E302791F557381FF437709023D1F30E"
  },
  "errors": [],
  "success": true
}
```

### Пример ответа с ошибкой

Отправлен неправильный bookingID

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

### Метод POST api/reservation/partners/send

**Описание:** позволяет переотправить выпущенные билеты/полисы по партнёрской услуге.

**Формат запроса** /api/apiVersion/reservation/partners/{{bookingId}}/send

где bookingId - уникальный ID заказа партнёрской услуги в данном бронировании.

### Параметры запроса

нет

### Пример запроса

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

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true.
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки IBE.
description	Описание ошибки IBE.
messages	Массив объектов, содержащий ошибки смежных систем. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.

	code	Код внутренней ошибки.
	text	Описание внутренней ошибки.

### Пример ответа

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

### Пример ответа с ошибкой

Отправлен неправильный bookingID

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

### Пример ответа с ошибкой

Попытка переотправить билет, который не был выпущен

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

### Сервис подтверждения (прайсинга) тарифов в PNR

<b>Сервис API</b>	pricing
<b>Краткое описание</b>	Сервис позволяет подтвердить тарифы, которые были использованы при создании б

<b>Поддерживаемые HTTP(S)-методы</b>	POST
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> PNR_AddMultiElements, Fare_PricePNRWithBookingClass, Ticket_CreateTS

**Описание:** позволяет подтвердить тарифы, которые были использованы при создании бронирования.

Сервис неявно вызывается при добавлении всех необходимых данных в PNR. Явно необходимо сделать вызов метода, если информация о ценах поменялась после создания бронирования.

Например, если был добавлен промокод на перелёт.

**Формат запроса** /api/apiVersion/pricing

**Тело запроса**

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

**Параметры запроса**

Параметр	Описание
<b>outboundFareFamily</b>	Объект с информацией о семействе тарифов для перелёта в направлении «туда»

	category	Категория семейства тарифов, например: ECOLIGHT/ECOFLEX/ECOPREM Возможные значения: см. [link] справочник основных категорий тарифов
	code	Код семейства тарифов, например: ECOLIGHTRU/ECOFLEXRU/ECOPREMРУ и др.
	isSubsidized	Признак субсидированного тарифа
<b>returnFareFamily</b>		Объект с информацией о семействе тарифов для перелёта в направлении «обратно»
	category	Категория семейства тарифов, например: ECOLIGHT/ECOFLEX/ECOPREM
	code	Код семейства тарифов, например: ECOLIGHTRU/ECOFLEXRU/ECOPREMРУ и др.
	isSubsidized	Признак субсидированного тарифа
<b>fares</b>		Массив объектов с информацией о выбранных тарифах
	fareBasis	Название базового тарифа. Например: WPROW или WECOW
	isBreakpoint	Признак того, заканчивает ли тариф своё действие на текущем сегменте.
sessionMarker		Маркер сессии Aviasales

## Формат ответа

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

```

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

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
<b>ticketFares</b>	Информация о подтверждённых ценах в разрезе пассажиров
referenceNumber	Номер TST
referenceQualifier	Тип ссылки Возможное значение: «TST»
baseAmount	Базовая цена билета (без тарифов и сборов) для данного сегмента и пассажира
totalAmount	Полная цена для данного сегмента и пассажира
<b>passengerReferences</b>	Массив с информацией о пассажирах, для которых справедлива указанная стоимость билетов.
referenceNumber	Ссылка на пассажира
referenceQualifier	Тип ссылки. РА – для взрослых, PI – для младенцев
<b>fareZapOff</b>	Объект с информацией о скидке, присутствующей в бронировании
amount	Размер скидки в процентах

	discountSsr	SSR с информацией о скидке Например, "DISCOUNT/DS03/1-1PSJTS5/" где DS03 – специальный формат процента скидки 1-1PSJTS5 – ID кампании скидок в CRM
	discountCardSsr	SSR с информацией о карте «Крылья», по которой предоставлена скидка. Например, "DISCOUNTCARD/1000332007"
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

## Примеры неуспешных запросов и ответов

Попытка подтвердить PNR с тарифом, который уже закончился:

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

## Сервисы управления дополнительными услугами

### Сервисы управления дополнительными услугами v2.2

<b>bookingStatusСервис API</b>	catalogue / services
<b>Краткое описание</b>	Сервисы позволяют получать информацию о дополнительных услугах и добавлять их в PNR
<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b>  Получение контекста PNR - PNR_Retrieve  Получение списка доступных услуг - Service_IntegratedCatalogue, Insurance_Pricing  Добавление услуг - PNR_AddMultiElements  Выпуск услуг - Ticket_RetrieveListOfTSM, Ticket_CreateTSMFromPricing  <b>Партнёрские сервисы:</b> Аэроэкспресс

#### Метод GET catalogue

**Описание:** получение информации о дополнительных услугах, которые доступны для добавления к бронированию.

**Формат запроса** /api/apiVersion/catalogue

#### Параметры запроса

Отсутствуют. Необходимые данные находятся в контексте бронирования.

#### Формат ответа

```
{
  "data": {
    "services": [{
      "code": string,
      "flightReferences": [{
        "referenceNumber": string,
        "passengerPrices": [{
          "referenceNumber": string,
          "amount": number
        }
      ]
    }
  ]
}
```



```

    ]
  }
]
},
"packs": [{
  "includedSeats": [string],
  "excludedSeats": [string],
  "includedServiceCode": string,
  "code": string,
  "flightReferences": [{
    "referenceNumber": string,
    "passengerPrices": [{
      "referenceNumber": string,
      "amount": decimal
    }]
  }]
}],
"seatServices": [{
"seatCharacteristics": [string],
  "code": string,
  "flightReferences": [{
    "referenceNumber": string,
    "passengerPrices": [{
      "referenceNumber": string,
      "amount": number
    }]
  }]
}],
}],
},
],
"petsServiceConfiguration": [{
  "petsList": [string],
  "flightReferences": [string]
}],
},
],
"insuranceProducts": [{
  "code": string,
  "amount": number
}],
},
],
"partnerProducts": [{
  "partnerId": 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
}
],
"serviceOptions": [{
    "code": string,
    "preselect": boolean,
    "category": string,
    "priority": integer,
    "flightReferences": [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
}

```

## Параметры ответа

Параметр	Описание
----------	----------

success		Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>		Данные, которые вернул запрос
<b>services</b>		Массив объектов с информацией о доступных услугах авиакомпании
	code	Код услуги Возможные значения: справочник доп. услуг
<b>flightReferences</b>		Массив с информацией о сегментах и пассажирах, для которых доступны услуги, а также о стоимости этих услуг.
	referenceNumber	Ссылка на сегмент перелёта
<b>passengerPrices</b>		Массив объектов с информацией о пассажире и стоимости услуги
	referenceNumber	Ссылка на конкретного пассажира
	amount	Стоимость услуги для конкретного пассажира
<b>packs</b>		Массив с доступными пакетами услуг
	includedSeats	Массив характеристик доступных мест в рамках пакета услуг
	excludedSeats	Массив характеристик недоступных мест в рамках пакета услуг
	includedServiceCode	Код включенной в пакет услуги
	code	Код пакета услуг. Возможные значения: Приложение 2. Справочник пакетов услуг
<b>flightReferences</b>		Массив с информацией о сегментах и пассажирах, для которых доступны услуги, а также о стоимости этих услуг.
	referenceNumber	Ссылка на сегмент перелета
<b>passengerPrices</b>		Массив объектов с информацией о пассажире и стоимости услуги
	referenceNumber	Ссылка на конкретного пассажира
	amount	Стоимость услуги для конкретного пассажира

	<b>serviceOptions</b>	Массив с информацией о том, какие услуги нужно автоматически предвыбирать на фронтенде движка бронирования.
	category	Категория услуги
	code	Код услуги Возможные значения: справочник доп. услуг
	flightReferences	Массив с информацией о сегментах, для которых доступны услуги
	preselect	Служебное поле. Если true, то услугу необходимо автоматически предвыбирать в UI. <i>Примечание: логика изначально добавлялась для использования в Web-клиенте.</i>
	priority	Служебное поле. Используется вместе с полем preselect. Означает приоритет автоматического добавления услуги в UI, 0 - наивысший. Например, если услуга с приоритетом 0 недоступна для предвыбора, то добавится услуга с приоритетом 1 и т.д.
	<b>seatServices</b>	Массив объектов с информацией о доступных местах для выбора на рейсе. <b>ВАЖНО:</b> Сервис api/seatmap возвращает более полную информацию о доступных местах и должен использоваться для бронирования услуги «выбор места».
	seatCharacteristics	Массив характеристик места. Возможные значения: Приложение 2. Справочник характеристик мест
	code	Код услуги, SEAT
	flightReferences	Массив с информацией о сегментах и пассажирах, для которых доступны услуги, а также о стоимости этих услуг.
	referenceNumber	Ссылка на сегмент перелёта
	<b>passengerPrices</b>	Массив объектов с информацией о пассажире и стоимости услуги

			referenceNumber	Ссылка на конкретного пассажира
			amount	Стоимость услуги для конкретного пассажира
			<b>petsServiceConfiguration</b>	Массив объектов, которые содержат информацию о доступности сервиса «провоз животных» на разных сегментах. Необходим для контроля провоза животных на тарифах семейства «бизнес».
			petsList	Массив строк, содержит список доступных к провозу животных
			flightReferences	Массив строк, содержит список сегментов, на которых доступна услуга
			<b>insuranceProducts</b>	Массив объектов с информацией о доступных страховках для выбора
			code	Код страховки Возможные значения: справочник доп. услуг (категория insurance)
			amount	Стоимость страховки
			<b>serviceCategories</b>	Массив объектов, содержит информацию о сервисах и порядке их отображения на сайте.
			code	Код услуги Возможные значения: справочник доп. услуг
			category	Категория услуги
			subCategory	Подкатегория услуги На текущий момент возвращается только для питания, возможные значения: vegetarian, special, snacks, hotmeal, childmeal, breakfast, dessert
			order	Порядок расположения на странице дополнительных услуг. 1 – верх страницы
			<b>partnerProducts</b>	Массив объектов, содержит информацию о партнёрских сервисах.

	partnerId	Уникальный ID партнёра в движке IBE. Альфастрахование – 1, Sealine страхование – 2, Аэроэкспресс – 4
	partnerType	Тип партнёра. Страхование – 1, Аэроэкспресс – 2
	code	Код партнёрской услуги. Для каждого партнёра заведены уникальные коды услуг. См. коды партнёрских услуг в приложении 5.
	amount	Стоимость партнёрской услуги.
	currency	Валюта тарификации.
	bookingStatus	Статус партнёрской услуги. Возможные значения: 0 - неизвестно, 1 - услуга не выпущена, 2 - услуга выпущена, 3 - выпуск завершился с ошибкой, 4 - выполнен возврат, 5 - услуга не может быть выпущена
	<b>portions</b>	Служебное поле
	type	Служебное поле
	number	Служебное поле
	flightReferenceNumbers	Служебное поле
	<b>crmEmdPromoCodes</b>	Массив объектов, содержащий информацию о промокодах на дополнительные услуги, получаемые от сервиса CoreApi
	serviceType	Код услуги
	promo	Промокод
	promoDescription	Описание промокода

	discountLevel	Размер скидки по промокоду
	expireDate	Дата завершения срока действия промокода
	moneyWallCoefficient	Коэффициент, для расчета ежемесячного платежа при оплате в рассрочку
	errors	Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus
	additionalInfo	Массив с дополнительной информацией об ошибке
	ssrType	Тип ssr
	info	Дополнительная информация

## Примеры успешных запросов и ответов

### Приложение X. Примеры ответов от сервисов

#### Метод POST services

**Описание:** позволяет добавлять услуги к бронированию.

**Формат запроса** /api/apiVersion/services

#### Тело запроса

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

```

    }
  ],
  "partnerProducts": [{
    "partnerId": integer,
    "code": string,
    "amount": decimal
  }
],
"services": [{
  "code": string,
  "text": string,
  "passengerReference": string,
  "flightReference": string,
  "referenceNumber": string
}
]
}

```

## Параметры запроса

Параметр	Описание
insurance	Код страховки, которую необходимо добавить. Возможные значения: справочник доп. услуг (категория insurance)
services	Массив объектов с информацией о добавляемых услугах
code	Код дополнительной услуги Возможные значения: справочник доп. услуг (категории, отличные от insurance, seat)
text	Тип животного Возможные значения: справочник животных
passengerReference	Ссылка на пассажира, для которого добавляется услуга
flightReference	Ссылка на сегмент, для которого добавляется услуга
referenceNumber	Ссылка на оплаченные услуги. Берется из ответа GET api/reservation.
seats	Массив объектов с информацией о добавляемых местах
number	Номер ряда для выбранного места
title	Место
code	Код места (Приложение 2. Справочники типов доп. услуг)



text	Текст
referenceNumber	Ссылка на оплаченные места. Берется из ответа GET api/reservation.
passengerReference	Ссылка на пассажира, для которого добавляется услуга
flightReference	Ссылка на сегмент, для которого добавляется услуга
<b>partnerProducts</b>	Массив объектов с информацией о добавляемых партнёрских услугах
partnerId	Уникальный ID партнёра
code	Уникальный код партнёрской услуги
amount	Стоимость партнёрской услуги

### Примеры запросов с услугами Авиакомпаний

Добавить авиа-страховку

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

Добавить один дополнительный багаж и трэвел-страховку

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

Добавить место

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

```

    }
  ],
  "insurance": null,
  "partnerProducts": [],
  "services": []
}

```

#### Добавить провоз животного

```

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

```

#### Примеры запросов с услугами партнёров

##### Добавить тариф "Стандартный" для Аэроэкспресс

```

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

```

#### Логика добавления/удаления услуг в бронирование

IBE всегда обновляет бронирование в части дополнительных услуг, основываясь на параметрах в запросе.

#### Например:

##### Добавить авиа-страховку и багаж

```

{
  "seats": [],

```

```

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

```

Чтобы удалить страховку из бронирования, нужно отправить следующий запрос

```

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

```

Данная логика реализована для всех типов услуг, в том числе партнёрских.

## Формат ответа

Формат ответа в части добавления услуг совпадает с ответом метода GET `api/reservation`.

Описание массивов `services`, `seats`, `insurance`, `partnerProducts` можно найти выше в данном разделе.

## Пример с докупкой дополнительных услуг

1. Создано и оплачено бронирование без дополнительных услуг
2. Добавляем багаж в бронирование. `referenceNumber` не указан, так как услуги ещё нет в бронировании.

REQUEST `api/services`

```

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

```

```
        "code": "ABAG"
      }
    ]
  }
}
```

### 3. Оплачиваем добавленную услугу

### 4. Пытаемся докупить сумку HBAG без указания услуги ABAG, которая уже оплачена:

REQUEST api/services

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

Получаем ошибку, так как нельзя удалить оплаченную услугу

RESPONSE api/services

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

### 5. Добавляем услугу HBAG, указывая оплаченную услугу ABAG (referenceNumber указан)

REQUEST api/services

```
{
  "insurance": null,
  "partnerProducts": [],
  "seats": [],
  "services": [
```

```

        {
            "passengerReference": "2",
            "flightReference": "1",
            "code": "ABAG",
            "referenceNumber": "17"
        },
        {
            "passengerReference": "2",
            "flightReference": "1",
            "code": "HBAG"
        }
    ]
}

```

6. В ответе GET api/reservation указаны 2 услуги: оплаченная и только что добавленная

RESPONSE PART api/reservation

```

"services": [
    {
        "code": "ABAG",
        "status": "HK",
        "passengerReference": "2",
        "flightReference": "1",
        "referenceNumber": "17",
        "ticketNumber": "262 4550915479",
        "price": 2000
    },
    {
        "code": "HBAG",
        "status": "HK",
        "passengerReference": "2",
        "flightReference": "1",
        "referenceNumber": "29"
    }
]

```

7. При вызове POST api/payment будет сформирована ссылка на оплату только услуги HBAG.

### Пример с покупкой партнёрской страховки в движке

*Ограничения:*

1. Страховка Sealine доступна только для мобильного API-ключа движка бронирования и мобильного офиса Amadeus соответственно.
2. Страховка Alfastrah доступна только для веб API-ключа движка бронирования и веб-офиса Amadeus соответственно.
3. Нельзя одновременно добавить страховку Amadeus (блок insurance) и партнерскую страховку (блок partnerProducts).

Коды партнёров и услуг доступны в приложении 4.

## Добавление страховки Alfastrah

1. Создать бронирование, запросить GET api/catalogue с API-ключом для веб-офиса, проверить, что в ответе есть партнерские страховки. Добавить одну из них в бронирование методом POST services.

REQUEST api/services

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

2. В ответе POST services есть блок с партнерской страховкой:

RESPONSE api/services

```
"partnerProducts": [
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 847.00,
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "3146ad12-bb61-445f-922f-10f36d0fc188"
  }
]
```

3. Пример для брони 1 взрослый + 1 ребенок + 1 младенец:

RESPONSE api/services

```
"partnerProducts": [
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 1694.00,
    "currency": "RUB",
    "passengerReference": "3",
    "isInfantProduct": true,
    "bookingId": "27713956-2687-49b8-a96f-25e61d252824"
  },
  {
```

```

    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 847.00,
    "currency": "RUB",
    "passengerReference": "3",
    "isInfantProduct": false,
    "bookingId": "1ad570e6-48da-4615-baf5-53769925a253"
  },
  {
    "partnerId": 1,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "TRAVEL",
    "amount": 847.00,
    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "9068c420-514e-4ea7-8790-46afb31dd7b0"
  }
]

```

## Добавление страховки Sealine

1. Создать бронирование, запросить GET api/catalogue с API-ключом для мобильного офиса, проверить, что в ответе есть партнерские страховки. Добавить одну из них в бронирование методом POST services.

REQUEST api/services

```

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

```

2. В ответе POST services есть блок с партнерской страховкой:

RESPONSE api/services

```

"partnerProducts": [
  {
    "partnerId": 2,
    "partnerType": 1,
    "bookingStatus": 1,
    "code": "SLAVIA",
    "amount": 260.00,

```

```

    "currency": "RUB",
    "passengerReference": "2",
    "isInfantProduct": false,
    "bookingId": "0e91e664-e4fa-4177-b2bc-e675ca197fa8"
  }
],

```

## Примеры неуспешных запросов при добавлении партнёрских услуг

Отправка неверного кода партнёра

Нет партнёра с кодом 111

```

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

```

Ответ

```

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

```

## Сервис получения информации о дополнительных услугах v2.3

Изменения в ответе метода GET api/catalogue версии 2.3:

Версия 2.2	Версия 2.3
В массиве services.flightReferences.referenceNumber типа string изменен на services.flightReferences.referenceNumbers массив типа 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>
---	--

В массиве `packs.flightReferences.referenceNumber` типа `string` изменен на `packs.flightReferences.referenceNumbers` массив типа `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>
---	--

В массиве `seatServices.flightReferences.referenceNumber` типа `string` изменен на `seatServices.flightReferences.referenceNumbers` массив типа `string`.

<pre> "seatServices": [{ "seatCharacteristics": [string], "code": string, "flightReferences": [{ "referenceNumber": string, "passengerPrices": [{ "referenceNumber": string, "amount": number } ] } ] } ] </pre>	<pre> "seatServices": [{ "seatCharacteristics": [string], "code": string, "flightReferences": [{ "referenceNumbers": [string], "passengerPrices": [{ "referenceNumber": string, "amount": number } ] } ] } ] </pre>
<p>В версии 2.3 удалены массивы insuranceProducts и partnerProducts.</p>	

### Сервис управления дополнительными услугами v2.3

**Изменения в версии 2.3:** в ответе сервиса блок insurance (страховки Амадеуса) не возвращается. Доступны только партнерские страховки из блока partnerProducts. В запросе удалены блоки insurance и partnerProducts.

Структура метода описана в api/reservation версии 2.3.

### Примеры запросов и ответов

<p>Ошибка при добавлении места (место недоступно)</p>
---

<pre>{   "seats": [     {       "number": 44,       "title": "A",       "passengerReference": "2",       "flightReference": "1"     }   ],   "insurance": null,   "partnerProducts": [],   "services": [] }</pre>	<pre>"errors": {   "seats": [{     "number": 44,     "title": "A",     "internalError": 2002,     "internalErrorMessage": "Seat is unavailable."   } ] }</pre>
<p><b>Ошибка при добавлении места (место занято)</b></p>	
<pre>{   "seats": [     {       "number": 5,       "title": "A",       "passengerReference": "2",       "flightReference": "1"     }   ],   "insurance": null,   "partnerProducts": [],   "services": [] }</pre>	<pre>"errors": {   "seats": [{     "number": 5,     "title": "A",     "internalError": 2002,     "internalErrorMessage": "Seat is unavailable."   } ] }</pre>
<p><b>Ошибка при добавлении места (попытка забронировать для взрослого с младенцем место не в среднем ряду)</b></p>	

<pre>{   "seats": [     {       "number": 6,       "title": "C",       "passengerReference": "2",       "flightReference": "1"     }   ],   "insurance": null,   "partnerProducts": [],   "services": [] }</pre>	<pre>"errors": {   "seats": [     {       "number": 6,       "title": "C",       "passengerReference": "2",       "flightReference": "1",       "internalError": 2001,       "internalErrorMessage": "Seat is unavailable for this passenger type."     }   ] }</pre>
<b>Ошибка при добавлении услуги авиакомпании (неверный код услуги)</b>	
<pre>{   "seats": [],   "insurance": null,   "partnerProducts": [],   "services": [     {       "code": "ABBB",       "passengerReference": "2",       "flightReference": "1"     }   ] }</pre>	<pre>"errors": {   "services": [{     "code": "ABBB",     "passengerReference": "2",     "flightReference": "1",     "internalError": 4002,     "internalErrorMessage": "Invalid service code."   } ] }</pre>
<b>Ошибка при добавлении услуги авиакомпании (неверная ссылка на пассажира)</b>	
<pre>{   "seats": [],   "insurance": null,   "partnerProducts": [],   "services": [     {       "code": "ABAG",       "passengerReference": "5",       "flightReference": "1"     }   ] }</pre>	<pre>"errors": {   "services": [{     "code": "ABAG",     "passengerReference": "5",     "flightReference": "1",     "internalError": 1001,     "internalErrorMessage": "Invalid passenger or segment reference."   } ] }</pre>

### Ошибка при добавлении партнёрской услуги (неверный код)

<pre>{   "seats": [],   "insurance": null,   "partnerProducts": [     {       "partnerId": 4,       "code": "STAND"     }   ],   "services": [] }</pre>	<pre>"errors": {   "partnerProducts": [{     "internalError": 1000,     "internalErrorMessage": "Unknown error.",     "partnerId": 4,     "partnerType": 2,     "code": "STAND",     "amount": 0,     "currency": "RUB"   } ]</pre>
---	---

### Метод добавления услуг в бронирование: `api/services/add`

**Описание:** метод добавления новых услуг, позволяющий не передавать в параметрах уже существующие в заказе услуги

### Формат запроса `api/apiVersion/services/add`

#### Тело запроса

```
{
  "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
  }
]
```

### Параметры запроса:

Параметр	Описание
<b>services</b>	Массив объектов с информацией о добавляемых услугах
code	Код дополнительной услуги  Возможные значения: справочник доп. услуг (категории, отличные от insurance, seat)
text	Тип животного  Возможные значения: справочник животных
passengerReference	Ссылка на пассажира, для которого добавляется услуга
flightReference	Ссылка на сегмент, для которого добавляется услуга
referenceNumber	Ссылка на оплаченные услуги. Берется из ответа GET api/reservation.
<b>seats</b>	
number	Номер ряда для выбранного места
title	Место
code	Код места (Приложение 2. Справочники типов доп. услуг)
text	Текст
passengerReference	Ссылка на пассажира, для которого добавляется услуга

Параметр	Описание
flightReference	Ссылка на сегмент, для которого добавляется услуга
referenceNumber	Ссылка на оплаченные места. Берется из ответа GET api/reservation.

### Метод удаления услуг из бронирования api/services/remove

**Описание:** метод удаления услуг из заказа, при условии, что на услугу не выпущено билета и сегмент перелета, на которую услуга была добавлена, еще не вылетел

**Формат запроса** api/apiVersion/services/remove

**Тело запроса** аналогично телу запроса api/services/add

### Сервис управления партнёрскими услугами

<b>Сервис API</b>	partners/types, partners/[id]
<b>Краткое описание</b>	Сервисы позволяют получать информацию о партнёрских услугах и добавлять их в бронирование
<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST
<b>Используемые методы смежных систем</b>	Партнёрские API

### Метод GET partners/types

**Описание:** сервис позволяет получать список партнёрских сервисов, которые доступны для бронирования для данного типа приложения.

Контроль осуществляется на основе переданного в метод API-key.

**Формат запроса** /api/apiVersion/partners/types

**Заголовки запроса** для вызова не нужно передавать X-session, никакие данные не берутся из контекста бронирования



**Параметры запроса** отсутствуют

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
partnerId	Уникальный код партнёра См. "Приложение 4. Партнёрские услуги", справочник "Коды партнёров"
name	Название партнёра
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

### Пример ответа (приложение типа Website)

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

### Пример ответа (приложение типа Mobile)

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

### Метод GET partners/id

**Описание:** сервис позволяет получать список услуг для конкретного партнёра.

**Формат запроса** /api/apiVersion/partners/id

**Пример запроса** /api/apiVersion/partners/1 - для получения услуг Альфастрахования

**Заголовки запроса** для вызова **нужно** передавать X-session, так как информация о партнёрских услугах зависит от состава пассажиров, направлений перелёта и др.

Вся информация будет браться из контекста бронирования.

## Параметры запроса

id - код партнёра

## Формат ответа

```
{
  "data": [{
    "partnerId": integer,
    "partnerType": integer,
    "bookingStatus": integer,
    "code": string,
    "amount": number,
    "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
}
```

## Параметры ответа

Параметр	Описание
----------	----------

success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
partnerId	Уникальный код партнёра См. "Приложение 4. Партнёрские услуги", справочник "Коды партнёров"
partnerType	Тип партнёра См. "Приложение 4. Партнёрские услуги", справочник "Типы партнёров"
bookingStatus	Статус бронирования услуги Поле оставлено для совместимости контракта. В ответе данного метода значение будет всегда 0.
code	Код партнёрской услуги
amount	Стоимость партнёрской услуги
currency	Валюта
<b>preselectedOption</b>	Массив настроек для предвыбора партнерской услуги (в корзине на третьем шаге)
preselect	Признак предвыбора партнерской услуги
<b>passengers</b>	Массив пассажиров, для которых предвыбирается партнерская услуга
referenceNumber	Ссылка на пассажира
passengerType	Тип пассажира, см. Приложение 2. Справочник типов пассажиров
<b>compatibleProductCodes</b>	Массив кодов партнерских услуг, которые могут быть добавлены в бронь с текущим тарифом
internalError	Код внутренней ошибки движка
internalErrorMessage	Описание внутренней ошибки движка
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE

	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus
	<b>additionalInfo</b>	дополнительная информация об ошибке
	ssrType	Тип ssr
	info	Информация

### Пример ответа, Аэроэкспресс

**Запрос** /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
}
```

### Пример ответа, АльфаСтрахование

**Запрос** /api/apiVersion/partners/1

API-ключ - мобильное приложение

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

```
}
```

## Пример ответа, Телемедицина

### Запрос /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
}
```

## Метод POST partners

**Описание:** сервис позволяет добавлять партнёрские услуги в бронирование.

### Формат запроса /api/apiVersion/partners

**Заголовки запроса** для вызова **нужно** передавать X-session, так как информация услуги добавляются для конкретного бронирования.

### Формат запроса

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

```
]
}
```

## Параметры запроса

Параметр	Описание
partnerId	Уникальный код партнёра См. "Приложение 4. Партнёрские услуги", справочник "Коды партнёров"
code	Код партнёрской услуги
amount	Стоимость добавляемой услуги
passengerReference	Ссылка на пассажира, которому добавляется услуга (обязательный параметр для телемедицины)

**Формат ответа** совпадает с ответом метода GET api/reservation.

## Пример ответа

```
{
  "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
}

```

### Примеры некорректных запросов и ответов

Запрос	Ответ
--------	-------



<p><b>Неправильный код партнёра для услуги TRAVEL</b></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><b>Неправильный код услуги для партнёра Sealine</b></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>
--	---

### Сервис получения информации о партнёрских услугах

<b>Сервис API</b>	reservation/partners
<b>Краткое описание</b>	<p>Позволяет получить информацию по партнёрским услугам в бронировании и выполнить переотправку оплаченных услуг.</p> <p>Нужен в основном для получения ссылок на билеты/полисы партнёров.</p>
<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST

<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> PNR_Retrieve, PNR_AddMultiElements <b>Аэроэкспресс:</b> getOrderTickets
---	--

## Метод GET `api/reservation/partners`

**Описание:** позволяет получить информацию о партнёрской услуге в бронировании

**Формат запроса** `/api/apiVersion/reservation/partners/{bookingId}/details`

где `bookingId` - уникальный ID заказа партнёрской услуги в данном бронировании.

## Пример запроса

После добавления партнёрской услуги, например, Аэроэкспресса в бронирование в ответе `api/services` присутствует следующая структура:

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

Для того, чтобы получить информацию об этой услуге, необходимо отправить запрос:

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

## Формат ответа

```
{
  "data": {
    "bookingId": string,
    "description": string,
    "document": string,
    "documentUrl": string,
    "passengerReferences": [
      string
    ],
    "flightReferences": [
      string
    ]
  }
}
```

```

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

```

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса.  Если массив ошибок errors пуст, то значение поля будет true.
<b>data</b>	Данные, которые вернул запрос.
bookingId	Уникальный ID партнёрской услуги в ИВЕ. Совпадает со значением из запроса.
description	Описание партнёрской услуги.  Например, "СТАНДАРТ" для тарифа STANDARTOW Аэроэкспресса. Или "ТРЭВЕЛ" для партнерской страховки Альфастрахование
document	ID заказа в партнёрской системе.

	<b>documentUrl</b>	Ссылка на билет/полис партнёра.
	<b>passengerReferences</b>	Массив со ссылками на пассажиров, для которых добавлена услуга.  Необязательное поле.
	<b>flightReferences</b>	Массив со ссылками на перелёты, для которых добавлена услуга.  Необязательное поле.
	<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
	<b>ibeCode</b>	Код ошибки IBE.
	<b>description</b>	Описание ошибки IBE.
	<b>messages</b>	Массив объектов, содержащий ошибки смежных систем.  Массив может быть пуст, рекомендуется анализировать ошибку по полю <b>ibeCode</b> .
	<b>code</b>	Код внутренней ошибки.
	<b>text</b>	Описание внутренней ошибки.

### Пример ответа для партнера Аэроэкспресс

```
{
  "data": {
    "bookingId": "8d7d39d9-2d87-49ff-970c-de94f4b0cb43",
    "description": "СТАНДАРТ",
    "document": "990801",
    "documentUrl":
    "https://api.aeroexpress.ru:48000/RenderTicket?oid=551913&guid=017669A2606DEA
    57D5EEDEE2F1DC86C9&token=74277705"
  },
  "errors": [],
}
```

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

### Пример ответа для партнера АльфаСтрахование

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

### Пример ответа с ошибкой

Отправлен неправильный bookingID

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

### Метод POST `api/reservation/partners/send`

**Описание:** позволяет переотправить выпущенные билеты/полисы по партнёрской услуге.

**Формат запроса** `/api/apiVersion/reservation/partners/{{bookingId}}/send`

где `bookingId` - уникальный ID заказа партнёрской услуги в данном бронировании.

### Параметры запроса

нет

### Пример запроса

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

### Формат ответа

```

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

```

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true.
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки ИВЕ.
description	Описание ошибки ИВЕ.
messages	Массив объектов, содержащий ошибки смежных систем. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код внутренней ошибки.
text	Описание внутренней ошибки.

### Пример ответа

```

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

```

### Пример ответа с ошибкой

Отправлен неправильный bookingID

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

### Пример ответа с ошибкой

Попытка переотправить билет, который не был выпущен

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

### Сервис получения схемы мест в самолёте

<b>Сервис API</b>	seatmap
<b>Краткое описание</b>	Сервис позволяет получать информацию о схеме мест в самолёте.
<b>Поддерживаемые HTTP(S)-методы</b>	GET
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> Air_RetrieveSeatMap

### Метод GET seatmap

**Описание:** получает информацию о схеме мест в самолёте.

**Формат запроса:** api/apiVersion/seatmap

### Пример запроса



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

## Параметры запроса

Информация о рейсе, для которого необходимо получить рассадку мест

Параметр	Описание
referenceNumber	Ссылка на сегмент перелёта
origin	Аэропорт вылета
destination	Аэропорт прилёта
departureDate	Дата и время отправления
flightNumber	Номер рейса
classOfService	Информация о выбранном тарифе
commercialFamily	Тип семейства тарифов, например: U6ECONOMY или U6BUSINESS

## Формат ответа

```
{
  "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
}

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
<b>isEmpty</b>	Признак вернулись ли данные из Amadeus. По этому полю можно определить доступность выбора мест на указанном сегменте.
<b>compartmentDetails</b>	Массив объектов с информацией о местах по рядам
number	Номер ряда
<b>seats</b>	Массив объектов, который содержит подробную информацию по местам
seat	Место. Возможные значения: A-F
number	Номер ряда для места

			description	<p>Описание места</p> <p>9 –центральное место</p> <p>A – aisle seat, место у прохода</p> <p>W – window seat, место у окна</p>
			characteristics	<p>Характеристики места</p> <p>Возможные значения: <a href="#">[link]</a> справочник характеристик</p>
			occupation	<p>Признак занятости места</p> <p>Возможные значения: O – occupied, занято</p>
			<b>passengerRelatedPrice</b>	<p>Массив объектов с информацией о цене мест для конкретных клиентов. Цены присутствуют только для мест с характеристикой CH = Chargeable seat, платное место.</p>
			name	Имя пассажира
			type	<p>Тип пассажира</p> <p>Возможные значения: <a href="#">[link]</a> справочник типов пассажиров</p>
			uniqueCustomerIdentifier	Уникальный номер пассажира
			seatPrice	Цена места
			isSeatUnavailable	Флаг доступности места
			hasInfant	Признак того, летит ли пассажир с младенцем
			classDesignator	<p>Информация о классе, к которому принадлежит место</p> <p>Возможные значения: ECONOMY, BUSINESS</p>
			<b>passengersTotalData</b>	Массив объектов с информацией о стоимости мест в разрезе пассажиров
			passengerReference	Ссылка на пассажира
			<b>passengerPriceLevels</b>	Массив объектов стоимостей мест, доступных для пассажира
			price	Стоимость места
			priceLevel	Категория стоимостей мест

<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>
description	Описание ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

## Примеры успешных запросов и ответов

Приложение X. Примеры ответов от сервисов

## Примеры неуспешных запросов и ответов

```
{
  "errors": [{
    "ibeCode": 1113,
    "description": "Amadeus Air_RetrieveSeatMap function returned error.",
    "messages": [{
      "code": "99",
      "text": null
    }]
  }]
},
"success": false
}
```

Из документации Amadeus: код 99 – это ошибка «Seat map not available for requested zone, seat may be requested», «Схема мест недоступна для запрошенной зоны, можно запросить место».

## Сервис оплаты бронирования

<b>Сервис API</b>	payment
<b>Краткое описание</b>	Сервис позволяет получить ссылку на оплату от системы ВРС

<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST
<b>Используемые методы смежных систем</b>	<b>ВРС:</b> Получение ссылки на оплату - registerOrder Получение информации о заказе - getPaymentResult <b>CRM:</b> Холдирование уникального промокода - UALPromotionCodesholdPromoCode

### Метод POST payment

**Описание:** формирует ссылку для перехода на страницу оплаты для данного бронирования.

**Формат запроса** /api/apiVersion/payment

#### Тело запроса

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

#### Параметры запроса

Параметр	Описание
lang	Язык интерфейса для формы оплаты ВРС. Возможные значения: <ul style="list-style-type: none"> <li>• ru - русский;</li> <li>• en - английский;</li> <li>• cn - китайский.</li> </ul>

redirectLang	<p>Язык редиректа в МУБ после оплаты.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> <li>• ru - русский;</li> <li>• en - английский;</li> <li>• zh - китайский;</li> <li>• uz - узбекский.</li> </ul>
returnUrl	<p>Адрес для возврата после оплаты.</p> <p>Например: <a href="https://service.uralairlines.ru/">https://service.uralairlines.ru/</a></p>

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	<p>Признак успешности выполнения запроса.</p> <p>Если массив ошибок errors пуст, то значение поля будет true</p>
<b>data</b>	Данные, которые вернул запрос
orderId	ID заказа для оплаты

	paymentFormURL	Сформированная ссылка для перехода на оплату. Формат: <a href="https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=номер_заказа&amp;hash=хэш&amp;systemId=идентификатор_системы&amp;language=язык">https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=номер_заказа &amp;hash=хэш &amp;systemId=идентификатор_системы &amp;language=язык</a>
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

### Примеры успешных запросов и ответов

```
{
  "data": {
    "orderId": "a2c0fa9a-9712-498e-b7f4-d10e21e43e68",
    "paymentFormURL":
    "https://web.rbsuat.com/amadeuspsp/static/payment/payment.html?orderId=a2c0fa9a-9712-498e-b7f4-d10e21e43e68&hash=7d0cb651e8b9bf767c8d1291d5c25b0fb1f9c58ac83b61ab1b4bb9853505eeb0&systemId=mainweb&language=ch"
  },
  "errors": [],
  "success": true
}
```

### Логика, добавленная после реализации покупки партнёрских услуг

1. Если в бронирование добавлена хотя бы одна партнёрская услуга, то для получения ссылки на оплату вызывается метод POST сервиса registerOrder (регистрации заказа в платёжной системе ВРС).

**ВАЖНО:** все SSR с партнёрскими услугами должны быть переданы в ВРС в виде:

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

Поле	Описание
ssrCodes	Массив объектов, который содержит блоки с SSR на дополнительные услуги
id	Уникальный идентификатор заказа
code	Уникальный код дополнительной услуги
gds	Для партнерских неоплетных услуг в качестве источника информации по бронированию используется GetOrderInfo. Поэтому в качестве кода GDS необходимо передавать значение = GOI

**Пример (Аэроэкспресс):**

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

2. Если в бронировании нет партнёрских услуг, то вызывается метод GET сервиса registerOrder (регистрации заказа в платёжной системе).

Структура, описанная выше, не передаётся.

## Метод GET payment

**Описание:** позволяет проверить статус оплаты бронирования

**Формат запроса** /api/apiVersion/payment



## Примеры запросов

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

## Параметры запроса

Параметр	Описание
orderId	Номер заказа для проверки

## Формат ответа

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

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
data	Информация о статусе оплаты. Если возвращается пустая строка, значит заказ не оплачен.
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE

<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю <code>ibeCode</code> .
<code>code</code>	Код ошибки Amadeus
<code>text</code>	Текст ошибки Amadeus

### Примеры успешных запросов и ответов

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

### Примеры неуспешных запросов и ответов

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

### Сервис авторизации в CRM

<b>Сервис API</b>	user
<b>Краткое описание</b>	Позволяет авторизоваться под существующим пользователем программы "Крылья" в CRM
<b>Поддерживаемые HTTP(S)-методы</b>	POST

<b>Используемые методы смежных систем</b>	<b>CRM:</b> AuthorizeMember, GetEntity, GetClientHousehold
---	--

## Метод POST user

**Описание:** авторизация в CRM

**Формат запроса** /api/apiVersion/user

### Параметры запроса

Параметр	Описание
login	Логин пользователя
password	Пароль пользователя
token	Токен авторизации

Метод позволяет авторизоваться как с логином и паролем напрямую, так и с токеном, полученным от метода getSession CRM.

### Примеры запросов

Авторизация с логином и паролем

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

Авторизация с токеном

1. Вызвать метод getSession с параметрами авторизации. Получить токен авторизации:

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

2. Авторизоваться с логином и токеном

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

## Формат ответа

```
{
  "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
}

```

## Параметры ответа

**Важно!** Все данные получаются из CRM и никак не валидируются на стороне ИВЕ.

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
userId	ID пользователя в CRM
firstName	Имя (кириллица)
firstNameLat	Имя (латиница)
middleName	Отчество (кириллица)
middleNameLat	Отчество (латиница)
lastName	Фамилия (кириллица)
lastNameLat	Фамилия (латиница)
birthDate	Дата рождения Формат: уууу-ММ-ddТНН:mm:ss
email	Электронная почта
cellPhone	Мобильный телефон
homePhone	Домашний телефон
<b>address</b>	Объект, содержащий информацию об адресе клиента
city	Город
country	Страна
region	Область, край, др.
street	Улица
zip	Индекс
<b>documents</b>	Массив объектов с информацией о документах пользователя
number	Номер документа
title	Название документа

	lastUpdateDate	Дата последнего использования документа
	documentType	Тип документа Возможные значения: справочник типов документов
	countryCode	Код страны, где был выпущен документ Возможные значения: справочник стран Например, «RUS»
	countryName	Название страны, где был выпущен документ Например, «Российская Федерация». Задаётся на стороне CRM, может не совпадать со значением из справочника стран.
	docId	Уникальный ID документа в CRM
	<b>logins</b>	Массив объектов с информацией о доступных логинах для пользователя
	loginName	Логин
	loginType	Тип логина Возможные значения: электронная почта, телефон, др.
	<b>loyalties</b>	Массив объектов с информацией о программах лояльности, в которых участвует пользователь
	memberId	ID программы
	level	Уровень программы
	program	Программа
	programArt	Артикул программы
	joinDate	Дата присоединения к программе
	currentPoints	Текущее количество баллов
	levelPoints	Служебное поле
	levelBonus	Служебное поле
	levelStartDate	Дата перехода на текущий уровень Формат: dd.MM.yyyy

	lastFlightDate	Дата последнего перелёта
	<b>cards</b>	Массив объектов Card с информацией о картах пользователя
	activeCard	Объект Card с информацией об активной карте пользователя
	sex	Пол пользователя Возможные значения: справочник "пол пассажиров".
	<b>wingsLoyalty</b>	Объект с информацией о программе лояльности «Крылья»
	memberId	ID программы
	level	Уровень программы
	program	Программа
	programArt	Артикул программы
	joinDate	Дата присоединения к программе
	currentPoints	Текущее количество баллов
	levelPoints	Служебное поле
	levelBonus	Служебное поле
	levelStartDate	Дата перехода на текущий уровень
	lastFlightDate	Дата последнего перелёта
	<b>cards</b>	Массив объектов Card с информацией о картах пользователя
	activeCard	Объект Card с информацией об активной карте пользователя
	<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

## Card



Параметр	Описание
active	Признак активности карты
ffn	Номер карты
name	Имя участника программы лояльности
cardLevel	Уровень карты
cardType	Тип карты
issueDate	Дата и время выпуска карты Формат: уууу-ММ-ддТНН:мм:сс
validFromDate	Дата и время начала действия карты Формат: уууу-ММ-ддТНН:мм:сс
validToDate	Дата и время окончания действия карты Формат: уууу-ММ-ддТНН:мм:сс

### Примеры успешных запросов и ответов

Приложение X. Примеры ответов от сервисов

### Сервис получения информации о связанных пользователях (CRM)

<b>Сервис API</b>	households
<b>Краткое описание</b>	Позволяет получать информацию о связанных пользователях (семья в небе).
<b>Поддерживаемые HTTP(S)-методы</b>	GET
<b>Используемые методы смежных систем</b>	<b>CRM:</b> GetClientHousehold

**ВАЖНО:** функционал является устаревшим на стороне CRM и не используется.

### Метод GET households

**Описание:** позволяет получить список связанных пользователей для авторизованного пользователя.

## Формат запроса /api/apiVersion/households

### Параметры запроса

Дополнительные параметры не требуются, информация берётся из контекста бронирования по ID сессии.

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
data	Массив объектов с информацией о связанных пассажирах
userId	ID пользователя в CRM
firstName	Имя (кириллица)
firstNameLat	Имя (латиница)

	middleName	Отчество (кириллица)
	middleNameLat	Отчество (латиница)
	lastName	Фамилия (кириллица)
	lastNameLat	Фамилия (латиница)
	birthDate	Дата рождения Формат: уууу-ММ-ддТНН:мм:сс
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок ИВЕ
	description	Описание ошибки, см. Приложение 1. Коды ошибок ИВЕ
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

## Примеры успешных запросов и ответов

Приложение X. Примеры ответов от сервисов

## Примеры неуспешных запросов и ответов

Метод вернёт ошибку 401 Unauthorized, если отсутствует заголовок X-Session или невозможно найти такую сессию или если пользователь в контексте текущего бронирования не авторизовался в CRM.

### Метод GET households/id

**Описание:** позволяет получить информацию по определённому связанному пользователю.

**Формат запроса** /api/apiVersion/households/id

### Примеры запросов

api/apiVersion/households/1-1PIR4T7

ID пользователя берётся из ответа на запрос api/households, поле data.userId.

## Формат ответа

Формат ответа совпадает с форматом ответа метода api/user.

## Сервис управления промокодами (CRM)

<b>Сервис API</b>	promotions
<b>Краткое описание</b>	Позволяет добавлять и удалять промокоды для текущего бронирования
<b>Поддерживаемые HTTP(S)-методы</b>	POST, DELETE
<b>Используемые методы смежных систем</b>	<b>CRM:</b> UALPromotionCodescheckPromoCode

## Метод POST promotions

**Описание:** позволяет добавлять промокод к бронированию. Одновременно могут действовать один промокод на перелёт и один на услугу.

**Формат запроса** /api/apiVersion/promotions

## Тело запроса

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

## Параметры запроса

Параметр	Описание
number	Промокод
origin	Аэропорт вылета
destination	Аэропорт прилёта

departureDate	Дата отправления Формат: уууу-ММ-дд
returnDate	Дата прибытия Формат: уууу-ММ-дд
passengersCount	Количество пассажиров

### Формат ответа

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

```

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

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
number	Промокод
amount	Размер скидки по промокоду
promotionStatus	Статус промокода Возможные значение: справочник статусов промокодов
promotionType	Тип промокода 0 – промокод на перелёт 1 – промокод на дополнительную услугу
description	Описание кампании промокодов
startFlightDate	Дата начала периода полетов по промокоду
endFlightDate	Дата окончания периода полетов по промокоду
startDate	Дата начала периода действия промокода
endDate	Дата окончания периода действия промокода
isUnique	Признак уникальности промокода
campaignId	ID кампании промокодов.
<b>directions</b>	Направления исключений/включений

		origin	Аэропорт вылета
		destination	Аэропорт прилёта
	<b>services</b>		Массив объектов с информацией об услугах, на которые действует промокод.
		code	Код услуги Возможные значения: справочник доп. услуг (кроме страховки)
	<b>families</b>		Массив объектов с информацией о тарифах, на которые действует промокод.
		code	Код тарифа
	<b>excludedFlights</b>		Общие исключения – рейсы, на которые не действуют скидки
		flightNumber	Номер рейса
		departureAirport	Аэропорт вылета
		arrivalAirport	Аэропорт прилёта
	<b>excludedPeriods</b>		Период исключения действия промокода
		startDate	Дата начала периода исключения действия промокода
		endDate	Дата окончания периода исключения действия промокода
	loyaltyCardIsRequired		Флаг, обозначающий промокод для ДКЛ
	hasConflict		Признак успешности добавления промокода в бронирование. False, если при добавлении промокода возникли ошибки.
	<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
		ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
		description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>		Массив объектов, содержащий ошибки из ответа CRM.
		code	Код ошибки CRM
		text	Текст ошибки CRM

## Примеры успешных запросов и ответов

### Приложение X. Примеры ответов от сервисов

## Примеры неуспешных запросов и ответов

```
{
  "errors": [{
    "ibeCode": 1205,
    "description": "CRM CheckPromoCode function returned error.",
    "messages": [{
      "code": "undefined",
      "text": "Promotion code not found or canceled"
    }]
  }
],
  "success": false
}
```

## Метод DELETE promotions

**Описание:** позволяет удалять добавленный промокод.

**Формат запроса** /api/apiVersion/promotions

### Примеры запросов

/api/promotions?number=oGVy099w

### Параметры запроса

Параметр	Описание
number	Промокод, который требуется удалить

### Формат ответа

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



## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
messages	Массив объектов, содержащий ошибки из ответа CRM.
code	Код ошибки CRM
text	Текст ошибки CRM

## Примеры неуспешных запросов и ответов

Если удаляемый промокод не существует, метод вернёт ошибку 404 Not found.

## Коды и тексты ошибок от сервиса проверки промокодов

Ошибка CRM	Текст в движке
code: "1", text: "Promocode not found"	РУС: Промокод не найден ENG: Promocode not found
code: "2", text: "Early purchase is not available for these flight dates" или code: "2", text: "Promotional code is not suitable for this return date" или code: "2", text: "Promotional code is not suitable for this flight date"	РУС: Данный промокод не может быть применен к перелетам с указанной датой ENG: This promotional code is not suitable for this flight date

code: "3", text: "It's not possible to apply a promotional code to a given destination"	<p>РУС: Данный промокод не может быть применен для указанного направления</p> <p>ENG: It is not possible to apply this promotional code to a given destination</p>
code: "4", text: "Promotional code is suitable for return flight"	<p>РУС: Промокод можно применить только купив билеты туда-обратно</p> <p>ENG: This promotional code is suitable for return flight</p>
code: "5", text: "This promotional code can not be applied to this number of passengers"	<p>РУС: Данный промокод не может быть применен к данному количеству пассажиров</p> <p>ENG: This promo code is not suitable for this number of passengers</p>
code: "6", text: "City of registration is not correct" или code: "6", text: "Loyalty card number for search return flight not passed" или code: "6", text: "The holder of this loyalty card can not apply the promotional code" или code: "6", text: "Date of registration is not correct"	<p>РУС: Ваша карта лояльности не подходит для указанного промокода</p> <p>ENG: Your loyalty card can not be applied to this promotional code</p>
code: "7", text: "This promotional code is available for purchase N ticket(s)"	<p>РУС: Билеты по данному промокоду закончились</p> <p>ENG: Tickets for this promo are over</p>
code: "8", text: "The entered personal promotional code is not available for this member"	<p>РУС: Введенный персональный промокод недоступен для Вашего аккаунта</p> <p>ENG: The entered personal promotional code is not available for your account</p>

code: "9", text: "This promocode is only for cardholders. Enter card number"	<p>РУС: Данный промокод доступен только для держателей карт. Пожалуйста, введите номер карты</p> <p>ENG: This promocode is only for cardholders. Please, enter card number</p>
--	--

## Сервис для получения скидок

<b>Сервис API</b>	discounts
<b>Краткое описание</b>	Позволяет получить применимую скидочную компанию по заданным параметрам поиска
<b>Поддерживаемые HTTP(S)-методы</b>	POST
<b>Используемые методы смежных систем</b>	<b>CRM:</b> CalculateDiscount_12, CalculateDiscountList2

### Метод POST discounts

**Описание:** запрашивает из CRM информацию по применимой скидочной компании по заданным параметрам запроса.

**Формат запроса** /api/apiVersion/discounts

### Тело запроса

```
{
  "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
  }
]
}

```

### Параметры запроса

Параметр	Описание
origin	ИАТА-код аэропорта отправления
destination	ИАТА-код аэропорта назначения
departureDate	Дата вылета туда
returnDate	Дата вылета обратно
adults	Количество взрослых пассажиров
children	Количество детей
pensioners	Количество пенсионеров
youths	Количество пассажиров типа «молодёжь»
subsidizedChildren	Количество детей, для которых предназначены субсидии

infants	Количество младенцев
fareFamilies	Массив с информацией о семействах тарифов
flights	Массив объектов с информацией о перелётах
flightId	Идентификатор рейса
flightNumber	Номер рейса
departureAirport	Аэропорт отправления
arrivalAirport	Аэропорт прибытия
bookingClass	Класс бронирования. В общем случае это первый символ fareBasis
referenceNumber	Ссылочный номер на сегмент на рейсе

### Формат ответа

```
{
  "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
}

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул сервис
id	Уникальный ID кампании скидок в CRM
name	Название кампании скидок в CRM
amount	Размер скидки
discountType	Тип скидки 0 – скидка в процентах 1 – скидка в рублях
usageType	Тип применения скидки 0 – скидка на тариф 1 – скидка на заказ
startDate	Дата начала действия кампании скидок
endDate	Дата завершения действия кампании скидок
<b>fareFamilies</b>	Массив с информацией о семействах тарифов
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE

	<b>messages</b>	Массив объектов, содержащий ошибки из ответа CRM.
	code	Код ошибки CRM
	text	Текст ошибки CRM

## Сервис получения и отправки на почту маршрутной квитанции

<b>Сервис API</b>	receipt
<b>Краткое описание</b>	Позволяет получать и отправлять на почту маршрутную квитанцию для текущего бронирования.
<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST
<b>Используемые методы смежных систем</b>	<b>CRM:</b> Receipt/PDF, Receipt/Send

### Метод GET receipt

**Описание:** метод запрашивает маршрутную квитанцию из CRM для текущего бронирования в контексте. Перед запросом этого метода нужно запросить информацию о бронировании при помощи метода GET api/reservation.

**Формат запроса** /api/apiVersion/receipt

### Параметры запроса

Дополнительные параметры не требуются.

### Формат ответа

PDF-файл с маршрутной квитанцией, закодированный в base64.

### Примеры неуспешных запросов и ответов

Если по сессии не находится информация о бронировании

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

```
}
```

### Таймаут при получении маршрутной квитанции

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

### Метод POST receipt

**Описание:** метод позволяет отправить на почту маршрутную квитанцию для текущего бронирования в контексте. Перед запросом этого метода нужно запросить информацию о бронировании при помощи метода GET api/reservation.

### Тело запроса

```
{
  "email": string
}
```

### Параметры запроса

Параметр	Описание
email	Электронная почта, на которую будет отправлена маршрутная квитанция.

### Формат ответа

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



}

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

## Примеры неуспешных запросов и ответов

Ошибка отправки маршрутной квитанции:

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

## Сервис добровольного возврата билетов и услуг

<b>Сервис API</b>	refund
<b>Краткое описание</b>	Позволяет рассчитать стоимость к возврату и вернуть билеты и услуги для бронирования
<b>Поддерживаемые HTTP(S)-методы</b>	POST
<b>Используемые методы смежных систем</b>	<p><b>Amadeus:</b></p> <p>Получение информации по бронированию - PNR_Retrieve, Ticket_RetrieveListOfTSM, Ticket_ProcessEDoc, PNR_AddMultiElement</p> <p>Инициация возврата - DocRefund_InitRefund, DocRefund_IgnoreRefund, DocRefund_UpdateRefund, DocRefund_ProcessRefund</p> <p>Отмена бронирования - PNR_Cancel</p> <p><b>CRM:</b></p> <p>Создание заявки на возврат - ServiceReqInsert_Input</p> <p><b>ВРС:</b></p> <p>Регистрация заказа на автовозврат - bmmRefund</p> <p><b>GOI:</b></p> <p>Получение НДС - GET payment/pnr</p>

### Метод POST refund/calculate

**Описание:** позволяет получить разбивку по ценам к возврату для бронирования в разрезе пассажиров.

**Формат запроса:** /api/apiVersion/refund/calculate

### Параметры запроса:

Параметр	Обязательность	Описание
pnrNumber	да	Номер PNR, уникальный шестисимвольный идентификатор бронирования

<b>passengers</b>	да	Массив с информацией о пассажирах, для которых требуется рассчитать суммы к возврату
lastName	да	Фамилия пассажира
passengerReferenceNumber	да	Ссылка на пассажира. Связано с полем data → passengers → referenceNumber
passengerType	да	Тип пассажира. Возможные типы пассажиров представлены в приложении 2.

## Примеры запросов

### Возврат взрослого пассажира

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

### Возврат взрослого и ребёнка

```
{
  "pnrNumber": "OWCXEV",
  "passengers": [
    {
      "lastName": "TESTOV",
      "passengerType": 0,
      "passengerReferenceNumber": "3"
    },
    {
      "lastName": "TESTOV",
      "passengerType": 3,
      "passengerReferenceNumber": "2"
    }
  ]
}
```

```
}
```

## Формат ответа

```
{
  "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
},
"serviceRefundDetails": [{
  "code": string,
  "refundBonuses": integer,
  "refundCurrency": integer,
  "refundTotal": integer,
  "taxDetailsInformation": [{
    "taxCategory": string,
    "taxDetails": [{
      "rate": string,
      "currencyCode": string,
      "type": string
    }
  ]
}
],
"ticketNumber": string
},
"partnerProducts": [{
  "document": string,
  "refundAmount": integer,
  "isRefundAmountUnknown": boolean,
  "partnerId": integer,
  "partnerType": integer,
  "bookingStatus": integer,
```

```

        "code": string,
        "amount": integer,
        "currency": string,
        "passengerReference": string,
        "isInfantProduct": boolean,
        "bookingId": string
    }
    ]
    },
    "refundWithTicketOnlyServices": [
        "string"
    ]
},
"errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
        "code": string,
        "text": string
    }
    ]
}
],
"success": boolean
}

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
passengerRefundDetails	Массив с информацией о возврате в разрезе пассажиров
hasInfant	Признак, показывающий летит ли данный пассажир с младенцем
passengerType	Тип пассажира Возможные типы пассажиров перечислены в Приложении 2
referenceNumber	Ссылка на пассажира
surname	Фамилия пассажира

		<b>ticketRefundDetails</b>	Массив с информацией о возврате по билетам
		isInfantTicket	Признак, является ли билет документом для младенца
		commissionAmount	Сумма комиссии за возврат
		refundBonuses	Количество бонусов "Крылья" к возврату
		refundCurrency	Количество рублей к возврату
		refundTotal	Всего к возврату, сумма refundBonuses и refundCurrency
		ticketNumber	Номер АТС Номер билета должен содержать 13 цифр
		<b>taxDetailsInformation</b>	Служебная информация о тарифах и сборах.  Полную информацию о значениях полей можно найти в документации Amadeus.  <b>ВАЖНО:</b> информацию из данного раздела можно не учитывать при возвратах.  Нужно использовать поля refundBonuses и refundCurrency для отображения пользователю.
		taxCategory	Категория сбора
		<b>taxDetails</b>	Массив, содержащий подробную информацию о тарифах/сборах
		rate	Сумма тарифа/сбора
		currencyCode	Код валюты
		type	Тип тарифа/сбора
		<b>serviceRefundDetails</b>	Массив с информацией о возврате по услугам
		code	Код услуги Возможные коды услуг перечислены в Приложении 2
		refundBonuses	Количество бонусов "Крылья" к возврату
		refundCurrency	Количество рублей к возврату
		refundTotal	Всего к возврату, сумма refundBonuses и refundCurrency
		ticketNumber	Номер EMD Номер билета должен содержать 13 цифр

		<b>taxDetailsInformation</b>	Служебная информация о тарифах и сборах.
		taxCategory	Категория сбора
		<b>taxDetails</b>	Массив, содержащий подробную информацию о тарифах/сборах
		rate	Сумма тарифа/сбора
		currencyCode	Код валюты
		type	Тип тарифа/сбора
		<b>partnerProducts</b>	Массив с информацией о возврате по партнерским услугам
		amount	Стоимость партнерской услуги
		bookingId	Уникальный идентификатор заказа партнерской услуги
		bookingStatus	Статус заказа партнерской услуги
		code	Код тарифа партнерской услуги. См. Приложение 4. Партнёрские услуги.
		currency	Код валюты
		document	Номер документа партнерской услуги. Для страховок - номер выпущенного полиса.
		isInfantProduct	Признак, показывающий, для младенца ли выпущена услуга
		isRefundAmountUnknown	Признак, показывающий, известна ли нам сумма к возврату партнерской услуги
		partnerId	Идентификатор партнера. См. Приложение 4. Партнёрские услуги.
		partnerType	Тип партнера. См. Приложение 4. Партнёрские услуги.
		passengerReference	Ссылка на пассажира, для которого возвращается партнерская услуга
		refundAmount	Количество рублей к возврату
		<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
		ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
		description	Описание ошибки, см. Приложение 1. Коды ошибок IBE

	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus.  Массив может быть пуст, рекомендуется анализировать ошибку по полю <code>ibeCode</code> .
	<code>code</code>	Код ошибки Amadeus
	<code>text</code>	Текст ошибки Amadeus

### Метод POST `refund/partial`

**Описание:** позволяет выполнить возврат билетов для бронирования в разрезе пассажиров.

**Формат запроса:** `/api/apiVersion/refund/partial`

#### Параметры запроса:

Параметр	Обязательность	Описание
<code>pnrNumber</code>	да	Номер PNR, уникальный шестисимвольный идентификатор бронирования
<b>passengers</b>	да	Массив с информацией о пассажирах, для которых требуется рассчитать суммы к возврату
<code>isTicketRefund</code>	да	Признак того, возвращается билет с услугами или только услуги
<b>partnerProducts</b>	нет	Массив партнерских услуг к возврату (параметры массива аналогичны параметрам ответа <code>apiVersion/partners/id</code> )
<code>lastName</code>	да	Фамилия пассажира
<code>passengerReferenceNumber</code>	да	Ссылка на пассажира.  Связано с полем <code>data</code> → <code>passengers</code> → <code>referenceNumber</code>
<code>passengerType</code>	да	Тип пассажира.  Возможные типы пассажиров представлены в приложении 2.



<b>services</b>	нет	Массив с информацией о приобретенных и возвращаемых доп. услугах авиакомпании
code	нет	Код возвращаемой доп. услуги
ticketNumber	нет	Номер билета возвращаемой доп. услуги

**Формат ответа:** совпадает с методом refund/calculate, добавляется поле `crmRefundRequestId` с номером созданной заявки на возврат в CRM (если не отправлена заявка на автовозврат в ВРС).

### Параметры ответа (краткая запись):

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок <code>errors</code> пуст, то значение поля будет <code>true</code>
<b>data</b>	Данные, которые вернул запрос
<b>passengerRefundDetails</b>	Массив с информацией о возврате. Полностью аналогичен данным в ответе метода <code>api/refund/calculate</code>
crmRefundRequestId	Номер созданной заявки на возврат в CRM. Например, 1-46МJI9D.
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.

### Примеры запросов

Пример запроса полного возврата билета с партнерскими услугами

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

```

        "code": "AVIADYNAMIC"
      }
    ],
    "services": [{
      "ticketNumber": "2624552270079",
      "code": "PETC"
    }
  ]
}

```

### Пример запроса возврата партнерских услуг без билета

```

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

```

### Пример запроса возврата доп. услуг авиакомпании без билета

```

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

```

}

## Описание процесса добровольного возврата

Добровольный возврат - возврат по собственному желанию клиента авиакомпании.

Краткое описание бизнес-процесса добровольного возврата.

1. Клиент заходит в МУБ (модуль управления бронированием) с использованием номера бронирования и фамилии.
2. Выбирает опцию "Вернуть билеты", открывается окно с информацией о пассажирах, имеющих в бронировании.
3. Выбирает Добровольный возврат.
4. Выбирает пассажиров, для которых нужно выполнить возврат, и опцию "Рассчитать сумму" (к возврату).
5. Выполняется запрос `api/refund/calculate`, сервис IBE рассчитывает суммы к возврату и клиенту показываются суммы к возврату (бонусов и рублей) и штрафы за возврат. Фактического возврата на данном шаге не происходит.
6. Клиент выбирает опцию "Подтвердить возврат" и выполняется запрос `api/refund/partial`.
7. Сервис IBE выполняет возврат услуг и/или авиабилета в АПИ партнеров и Амадеус
8. Сервис IBE валидирует возможность регистрации заказа на автовозврат.
9. Сервис IBE регистрирует заказ к возврату (с помощью отдельного метода для регистрации заказа на возврат в ПМ. Регистрация всегда на исходный PNR, по которому была оплата) .
10. ПМ проверяет возможность автовозврата для текущего заказа: запрашивает `GET api/refund` , проводит валидацию состава заказа и сумм к возврату.
11. ПМ отправляет запрос на возврат в банк.
12. ПМ отправляет статус запроса на возврат: вариант А - заказ принят банком к возврату вариант В - заказ не принят к возврату, либо получена ошибка от ПМ
13. Далее вариант А:
  1. Сервис IBE информирует пользователя об успешно выполненном запросе на возврат с информацией о том, что денежные средства поступят на карту в течение некоторого времени в зависимости от банка
  2. ПМ фискализует чек на возврат. Чек отправляется клиенту.
  3. При повторном входе в МУБ отображается информация о том, что на заказ оформлен возврат.
14. Далее вариант В:
  1. Сервис IBE формирует заявку на возврат денежных средств в CRM.
  2. В случае неуспешного возврата показывается одно из возможных сообщений об ошибке.
  3. Сервис IBE информирует пользователя об успешно созданной заявке на возврат с информацией о номере заявки.
  4. В течение 30 дней заявка в CRM обрабатывается вручную оператором, при этом из движка запрашиваются суммы к возврату.

5. После возврата средств операция подтверждается в ИВЕ (в базу данных сохраняется информация о том, что возврат средств был выполнен).
6. Возврат считается выполненным.

#### Общая техническая информация.

1. Для каждого возврата по бронированию (при вызове метода api/refund) в базе данных ИВЕ создаётся запись с уникальным ID сессии возврата.
2. С этой сессией могут быть связаны 3 сущности: а) сущность с информацией о возврате билетов б) о возврате услуг авиакомпании в) о возврате партнёрских услуг.
3. Сессия возврата имеет несколько состояний. При возврате движок бронирования переводит сессию из одного состояния в другое, выполняя соответствующие операции:
  1. Initiated - сессия возврата создана;
  2. VatCalculated - рассчитана информация о НДС по возврату;
  3. PartnerServicesCanceled - выполнена отмена по партнёрским услугам, если это необходимо;
  4. PnrSplitted - бронирование разделено на несколько (в случае частичного возврата, когда бронирование возвращается не полностью);
  5. CrmRequestCreated - создана заявка в CRM на добровольный возврат (в случае, если невозможно оформить автовозврат);
  6. PnrCanceled - бронирование и билеты отменены (операцию невозможно отменить). В случае частичного возврата выполняется отмена дочернего PNR и соответствующих ему билетов.
  7. ManualAnalysis - не удалось докатить сессию добровольного возврата в случае ошибки. См. пункт 5.
  8. AutoRefundCompleted - отправлена заявка на автовозврат в ВРС.
  9. ServicesCanceled - выполнена отмена по услугам АК, если это необходимо.
4. Сущности с информацией о билетах, услугах АК и партнёрских услугах также имеют несколько состояний:
  1. NotAvailable - сущность недоступна для получения информации о суммах к возврату;
  2. Available - можно запросить информацию о суммах к возврату по билету/услуге;
  3. Processed - возврат средств подтверждён;
  4. ManualProcessingRequired - необходим ручной разбор ситуации с возвратом средств;
  5. Failed - подтверждение возврата средств завершилось с ошибкой.
5. Докат сессии возврата.

Если в процессе возврата возникает ошибка и операция зависает в одном из промежуточных статусов (до PnrCanceled), то специальное задание по докату периодически пытается завершить этот процесс.

Если по истечению времени, отведённого на докат, сессия возврата не перешла в

конечное состояние, движок логирует ошибку и переводит сессию в состояние ManualAnalysis.

Ошибка обрабатывается в Azure Insights и по этому событию присылается письменное оповещение. Далее идёт ручной разбор этой ситуации.

Для получения информации о сессии возврата и подтверждения возврата средств используются методы GOI-proxy GET refund/pnr и PUT refund/pnr.

Эти методы описаны отдельно в документации GOI.

### **Сервис добровольного возврата билетов с невозвратным тарифом: промо-лайт, промо, безбагажный**

Добровольный ручной возврат является разновидностью обычного добровольного возврата и применяется для билетов с невозвратным тарифом: промо-лайт, промо, безбагажный.

Процесс аналогичен обычному добровольному возврату, но с некоторыми исключениями.

1. Клиент заходит в МУБ (модуль управления бронированием) с использованием номера бронирования и фамилии.
2. Выбирает опцию "Вернуть билеты", открывается окно с информацией о пассажирах, имеющих в бронировании.
3. Выбирает пассажиров, для которых нужно выполнить возврат, и опцию "Рассчитать сумму" (к возврату).
4. Выполняется запрос `api/refund/calculate`. **Показано сообщение** "Ваш билет является невозвратным. Возврату подлежат только аэропортовые сборы и стоимость дополнительных услуг по неиспользованным участкам перевозки".  
**Сумму к возврату за билеты не показываем, отдаем `ticketRefundDetails.refundTotal = 0`.**  
Если в бронировании не было доп услуг, блок "Итого к возврату" не показываем. Если доп услуги были, то в блоке "Итого к возврату" показана сумма только за доп услуги (без детализации).
5. Клиент выбирает опцию "Подтвердить возврат" и выполняется запрос `api/refund`.
6. В случае успешного возврата пользователю показывается сообщение с информацией об ID заявки в CRM.
7. В случае неуспешного возврата показывается одно из возможных сообщений об ошибке.
8. В течение 30 дней заявка в CRM обрабатывается вручную оператором.

## Примеры ошибочных запросов и ответов

Описание	Пример запроса	Пример ответа
<p>В ответе api/refund нет информации о созданной заявке в CRM, поле crmRefundRequestId не заполнено.</p> <p>Считается, что возврат завис в одном из промежуточных статусов. Дополнительных действий по возврату данного бронирования не требуется. Возврат будет либо завершён автоматически, либо разобран вручную, когда количество попыток автоматического доката истечёт.</p>	-	-
<p>Бронирование для 2 взрослых.</p> <ol style="list-style-type: none"> <li>1. Инициирован возврат по первому взрослому, но бронирование не завершилось до конца.</li> <li>2. Выполнена попытка инициировать возврат по второму взрослому.</li> </ol>	<p>Запрос для пункта 2.</p> <pre>{   "pnrNumber": "OY2P8Q",   "passengers": [     {       "lastName": "TESTOV",       "passengerReferenceNumber": 3,       "passengerType": 0     }   ] }</pre>	<pre>{   "errors": [     {       "ibeCode": 1033,       "description": "There are incomplete refund processes.",       "messages": []     }   ],   "success": false }</pre>

<p>Допустим, в бронирование OY2P8Q добавлен взрослый пассажир (тип пассажира 0), правильная ссылка на пассажира - 2.</p> <p>Ошибка будет возникать в случае, если:</p> <ol style="list-style-type: none"> <li>1. Передана неправильная ссылка на пассажира.</li> <li>2. Передан неправильный тип пассажира (любой, кроме 0).</li> </ol>	<p>Запрос для пункта 1.</p> <pre>{   "pnrNumber": "OY2P8Q",   "passengers": [     {       "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 0     }   ] }</pre> <p>Запрос для пункта 2.</p> <pre>{   "pnrNumber": "OY2P8Q",   "passengers": [     {       "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 1     }   ] }</pre>	<pre>{   "errors": [     {       "ibeCode": 1041,       "description": "Some passengers are not exist in reservation.",       "messages": []     }   ],   "success": false }</pre>
<p>Бронирование для 1 взрослого и 1 ребёнка.</p> <p>Невозможно отменить только взрослого, возникает ошибка "Оставшиеся пассажиры в бронировании имеют неправильный тип".</p>	<pre>{   "pnrNumber": "OWWDII",   "passengers": [     {       "lastName": "TESTOV", "passengerReferenceNumber": 3, "passengerType": 0     }   ] }</pre>	<pre>{   "errors": [     {       "ibeCode": 1042,       "description": "The remaining passengers are of the wrong type.",       "messages": []     }   ],   "success": false }</pre>

## Сервис вынужденного возврата билетов. Упрощенная схема.

<b>Сервис API</b>	<b>InvoluntaryRefund</b>
<b>Краткое описание</b>	Позволяет создать заявку на вынужденный возврат
<b>Поддерживаемые HTTP(S)-методы</b>	POST
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> PNR_Retrieve, PNR_AddMultiElements <b>CRM:</b> Создание заявки на возврат - ServiceReqInsert_Input

### Метод POST InvoluntaryRefund

**Описание:** В рамках упрощенной схемы какие-либо операции с PNR в Amadeus не выполняются. Метод позволяет создать заявку в CRM на вынужденный возврат.

**Формат запроса:** /api/apiVersion/InvoluntaryRefund

### Параметры запроса:

Параметр	Обязательность	Описание
pnrNumber	да	Номер PNR, уникальный шестисимвольный идентификатор бронирования
lastName	да	Фамилия пассажира
<b>passengers</b>	да	Массив с информацией о пассажирах
lastName	да	Фамилия пассажира
passengerReferenceNumber	да	Ссылка на пассажира. Связано с полем data → passengers → referenceNumber
passengerType	да	Тип пассажира. Возможные типы пассажиров представлены в приложении 2.



<b>refundReason</b>	да	1 - в связи с отменой рейса 2 - другая причина
<b>contact</b>	да	
firstName	да	Имя пассажира
surname	да	Фамилия пассажира
middleName	нет	Отчество пассажира
email	да	Почта пассажира
phone	да	Телефон пассажира
<b>comment</b>	да	Применяется только для только для <b>refundReason = 2</b> . Поле комментария, в котором пассажир указывает причину вынужденного возврата.

### Примеры запросов:

#### 1. Для варианта "В связи с отменой рейса":

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

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

```
}  
}
```

2. Для варианта "**Другая причина**". В этом варианте обязательно должно быть заполнено поле "comment" и загружен документ, подтверждающий причину возврата.

```
-----WebKitFormBoundary56NzvwpTROrAE6Pc  
Content-Disposition: form-data; name="files"; filename="photo_2019-09-10_12-  
27-09.jpg"  
Content-Type: image/jpeg
```

```
-----WebKitFormBoundary56NzvwpTROrAE6Pc  
Content-Disposition: form-data; name="refundDetails"  
  
{  
  "pnrNumber": "URSOXM",  
  "lastName": "KOLTSOVA",  
  "passengers":  
  [  
    {  
      "lastName": "KOLTSOVA",  
      "passengerType": 4,  
      "passengerReferenceNumber": "3",  
    }  
  ],  
  "refundReason": 2,  
  "comment": "Не смогла явиться на рейс по причине болезни",  
  "contact":  
  {  
    "firstName": "Ольга",  
    "surname": "Кольцова",  
    "middleName": "Михайловна",  
    "email": "koltsovaolga5@gmail.com",  
    "phone": "88120007676"  
  }  
}
```

### Параметры ответа (краткая запись):

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
data	Данные, которые вернул запрос

crmRefundRequestId	Номер созданной заявки на возврат в CRM. Например, 1-46MJI9D.
errors	Информация об ошибках, возникших во время работы сервиса.

## Описание упрощенной схемы вынужденного возврата

1. Клиент заходит в МУБ (модуль управления бронированием) с использованием номера бронирования и фамилии.
2. Выбирает опцию "Вернуть билеты", переходит на вкладку "Вынужденный".
3. Выбирает причину возврата.  
По умолчанию установлена опция "Другая причина". В этом варианте пользователю необходимо описать причину возврата и загрузить документ, подтверждающий причину возврата. В варианте "В связи с отменой рейса" этого делать не требуется.
4. Выбирает пассажиров, для которых нужно выполнить возврат.
5. Заполняет поля персональных данных: ФИО, номер телефона, электронную почту.
6. Выбирает опцию "Подтвердить возврат" и выполняется запрос `api/InvoluntaryRefund`.
7. В случае успешного создания заявки на возврат пользователю показывается сообщение с информацией об ID заявки в CRM.
8. В случае неуспешного создания заявки показывается одно из возможных сообщений об ошибке.

Подсчета сумм к возврату не производится.

История заявок на вынужденные возвраты в БД не хранится.

Контроля исполнения заявки на возврат на стороне оператора нет.

Ограничений по количеству заявок, оформляемых для одного PNR, нет.

## Сервис изменения данных о пассажире

Сервис API	passenger
Краткое описание	Позволяет изменить данные о пассажире в бронировании
Поддерживаемые HTTP(S)-методы	PUT
Используемые методы смежных систем	Amadeus: PNR_Retrieve, PNR_ChangeElement, PNR_NameChange, PNR_AddMultiElements, PNR_Cancel

**ВАЖНО:** на данный момент при изменении данных в бронировании никак не изменяются данные для партнёрских услуг.

**Метод не следует использовать, если такие услуги есть в бронировании. Функционал будет доработан.**

## Метод PUT passenger

**Описание:** позволяет изменять данные о пассажире в бронировании

Реализованная логика по изменению данных:

1. Для полей «Фамилия», «Имя» разрешать изменять не более 3-х символов (в сумме для всех полей) и не более 1-го раза.
2. Следующие поля информации о пассажире можно менять бесплатно сколько угодно много раз: почта, телефон, гражданство, пол, номер документа, дата рождения. Дату рождения возможно изменить только с учётом допустимого возраста для конкретного типа пассажира.

**Формат запроса:** /api/apiVersion/passenger

## Параметры запроса

Параметр	Обязательность	Описание
referenceNumber	да	Ссылка на пассажира
passengerType	да	Тип изменяемого пассажира
surname	нет	Фамилия
firstName	нет	Имя
middleName	нет	Отчество
email	нет	Электронная почта
phone	нет	Телефон
nationality	нет	Национальность
sex	нет	Пол
birthDate	нет	Дата рождения
documentNumber	нет	Номер документа

## Формат ответа

```
{  
  "data": {  
    "id": number,  
  }  
}
```

```

    "hasInfant": boolean,
    "referenceNumber": string,
    "firstName": string,
    "middleName": string,
    "surname": string,
    "passengerType": number,
    "birthDate": date,
    "sex": number,
    "nationality": string,
    "documentNumber": string,
    "documentExpiryDate": date,
    "email": string,
    "phone": string,
    "documentType": number,
    "isAdult": boolean,
    "isChanged": boolean
  },
  "errors": [{
    "ibeCode": number,
    "description": string,
    "messages": [{
      "code": string,
      "text": string
    }]
  }],
  "success": boolean
}

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
id	ID пользователя в бронировании (данные IBE)
isAdult	Относится ли пассажир к типам ADT/SRC/YTH
hasInfant	Привязан ли к пассажиру младенец в бронировании
isChanged	Признак, изменялись ли в бронировании имя и фамилия. При изменении других полей этот признак будет иметь значение false.
referenceNumber	Ссылка на пассажира (данные Amadeus)
firstName	Имя
middleName	Отчество

	surname	Фамилия
	passengerType	Тип пассажира
	birthDate	Дата рождения
	sex	Пол
	nationality	Национальность
	documentNumber	Номер документа
	documentType	Тип документа
	documentExpiryDate	Дата действия документа
	email	Электронная почта
	phone	Телефон
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

### Пример ответа

Описание	Пример запроса	Пример ответа
----------	----------------	---------------

<p>Не передан обязательный параметр referenceNumber</p>	<pre>{   "surname": "Test" }</pre>	<pre>{   "errors": [     {       "ibeCode": 1008,       "description": "Data validation error.",       "messages": [         {           "code": "ReferenceNumber",           "text": "'Reference Number' should not be empty."         }       ]     }   ],   "success": false }</pre>
<p>Передано неправильное значение параметра passengerType</p>	<pre>{   "referenceNumber": 2,   "passengerType": 1 }</pre>	<pre>{   "errors": [     {       "ibeCode": 1082,       "description": "Passenger not found in reservation",       "messages": []     }   ],   "success": false }</pre>

<p>Попытка изменить для полей surname / firstName больше, чем 3 символа</p>	<p>Фамилия пассажира в бронировании: Testov (фамилия в бронировании хранится в верхнем регистре)</p> <pre>{   "referenceNumber": 2,   "passengerType": 0,   "surname": "Testov" }</pre>	<pre>{   "errors": [     {       "ibeCode": 1018,       "description":         "Unable to change         more then maximum         allowed symbols",       "messages": []     }   ],   "success": false }</pre>
<p>Повторная попытка изменить поле surname / firstName</p>	<p>Для предыдущего примера отправляем сначала валидный запрос</p> <pre>{   "referenceNumber": 2,   "passengerType": 0,   "surname": "TESTOVA" }</pre> <p>Затем пытаемся изменить фамилию снова</p> <pre>{   "referenceNumber": 2,   "passengerType": 0,   "surname": "TESTOV" }</pre>	<pre>{   "errors": [     {       "ibeCode": 1019,       "description":         "Unable to change         element twice",       "messages": []     }   ],   "success": false }</pre>



<p>Попытка изменить дату рождения на невалидную для данного пассажира</p>	<pre>{   "referenceNumber": 2,   "passengerType": 0,   "birthDate": "2019-02-01" }</pre>	<pre>{   "errors": [     {       "ibeCode": 1008,       "description": "Data validation error.",       "messages": [         {           "text": "Passenger age is incorrect"         }       ]     }   ],   "success": false }</pre>
---	--	---

<p>Корректный запрос на изменение</p>	<p><b>Пассажир в бронировании</b></p> <pre>{   "id": 1,   "hasInfant": false,   "referenceNumber": "2",   "firstName": "ALEKSANDR",   "middleName": "",   "surname": "TESTOV",   "passengerType": 0,   "birthDate": "1988- 08-12T00:00:00",   "sex": 0,   "nationality": "RUS",   "documentNumber": "1409111227",   "documentExpiryDate": "2029-07-16T00:00:00",   "email": "TEST@YANDEX.RU",   "phone": "79510002233",   "documentType": 0,   "isAdult": true,   "isChanged": false }</pre> <p><b>Запрос</b></p> <pre>{   "referenceNumber": 2,   "passengerType": 0,   "birthDate": "1988- 08-13",   "surname": "TESTOVA",   "firstName": "ALEKSANDRA",   "middleName": "KONSTANTINOVNA",   "documentNumber": "14152220011",   "email": "testova@gmail.com" }</pre>	<pre>{   "data": {     "id": 1,     "hasInfant": false,     "referenceNumber": "2",     "firstName": "ALEKSANDRA",     "middleName": "KONSTANTINOVNA",     "surname": "TESTOVA",     "passengerType": 0,     "birthDate": "1988-08- 13T00:00:00",     "sex": 0,     "nationality": "RUS",     "documentNumber": "14152220011",     "documentExpiryDate": "2029-07- 16T00:00:00",     "email": "TESTOVA@GMAIL.COM",     "phone": "79510002233",     "documentType": 0,     "isAdult": true,     "isChanged": true   },   "errors": [],   "success": true }</pre>
---------------------------------------	---	---

## Сервис онлайн-регистрации

<p><b>Сервис API</b></p>	<p>Checkin, Checkin/GetPassengerCheckinStatus</p>
<p><b>Краткое описание</b></p>	<p>Позволяет из МУБ проверить возможность онлайн-регистрации на рейс и, если она доступна, перейти в TAIS</p>

<b>Поддерживаемые HTTP(S)-методы</b>	GET
<b>Используемые методы смежных систем</b>	TAIS: CheckInGetData

## Метод GET Checkin

**Описание:** позволяет получить данные о возможности регистрации у сервиса CheckInGetData системы ТАИС.

Если регистрация доступна, возвращает время, оставшееся до окончания регистрации на вылет первого рейса туда/обратно.

**Формат запроса** /api/apiVersion/checkin

### Параметры запроса

Параметр	Описание
lastName	Фамилия одного из пассажиров, добавленных в бронирование. Параметр передаётся латиницей.
pnrNumber	Номер PNR, для которого нужно получить информацию
ticketNumber	Номер билета пассажира

### Формат ответа

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

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Информация об онлайн-регистрации
canRegister	Признак доступности онлайн-регистрации
remainingTime	Оставшееся время до окончания онлайн-регистрации в секундах
<b>passengers</b>	Список пассажиров
ticketNumber	Номер билета пассажира
registered	Признак совершения регистрации
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса
ibeCode	Код ошибки, см. Приложение 1 Коды ошибок IBE
description	Описание ошибки, см. Приложение 1 Коды ошибок IBE
messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus

text	Текст ошибки Amadeus
------	----------------------

### Примеры успешных ответов

Ответ, когда регистрация открыта и второй пассажир уже зарегистрирован

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

Ответ, когда регистрация еще не открыта:

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

Ответ, когда регистрация уже закрыта:

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

### Пример неуспешного ответа

```
{
  "errors": [
    {
      "ibeCode":1700,
      "massages": [
        {
          "code": "NOT_FOUND",
          "text": ""
        }
      ]
    }
  ],
  "success": false
}
```

### Метод GET Checkin/GetPassengerChekinStatus

**Описание:** позволяет получить данные о возможности регистрации в разрезе каждого сегмента у сервиса CheckInGetData системы ТАИС.

**Формат запроса** /api/apiVersion/Checkin/GetPassengerChekinStatus

#### Параметры запроса:

Параметр	Описание
lastName	Фамилия одного из пассажиров, добавленных в бронирование. Параметр передаётся латиницей.
pnrNumber	Номер PNR, для которого нужно получить информацию
ticketsNumberCommaSeprated	Массив номеров билетов всех пассажиров в бронировании

#### Формат ответа

```

{
  "data": [
    {
      "ticketNumber": string,
      "registered": boolean
    }
  ],

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

  "success": boolean
}

```

### Параметры ответа

Параметр	Описание
<b>data</b>	Массив объектов с информацией о доступности регистрации пассажиров
ticketNumber	Номер билета пассажира
registered	Признак доступности регистрации
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса
ibeCode	Код ошибки, см. Приложение 1 Коды ошибок IBE
description	Описание ошибки, см. Приложение 1 Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true

## Сервис аннулирования билетов

<b>Сервис API</b>	annul
<b>Краткое описание</b>	Позволяет аннулировать билеты и получать информацию по аннулированным бронированиям
<b>Поддерживаемые HTTP(S)-методы</b>	GET, POST
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> Получение информации по бронированию - PNR_Retrieve, Ticket_RetrieveListOfTSM, Ticket_ProcessEDoc, PNR_AddMultiElement  Отмена бронирования - PNR_Cancel

### Метод POST api/annul

**Описание:** позволяет аннулировать билеты для всех пассажиров в бронировании

**Формат запроса:** /api/apiVersion/Annul

**Параметры запроса:** в хедере передается X-Session

### Формат ответа:

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

### Параметры ответа:

Параметры	Описание
-----------	----------



success	Признак успешности выполнения запроса.  Если массив ошибок errors пуст, то значение поля будет true
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
messages	Массив объектов, содержащий ошибки из ответа Amadeus.  Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

### Метод GET api/annul

**Описание:** позволяет получить информацию по аннулированию бронирования

**Параметры запроса:** в хедере передается X-Session

Параметр	Обязательность	Описание
pnrNumber	да	Номер PNR, уникальный шестисимвольный идентификатор бронирования

### Формат ответа:

```
{
  "data": [{
    "pnrNumber": string,
    "createdDate": datetime
  }]
```

```

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

```

### Параметры ответа:

Параметры	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
pnrNumber	Номер PNR
createdDate	Дата и время, когда было оформлено аннулирование
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

### Сервис получения схемы мест в самолете без PNR

<b>Сервис API</b>	SeatmapStandalone
<b>Краткое описание</b>	Сервис позволяет получать информацию о схеме мест в самолёте без созданного бронирования
<b>Поддерживаемые HTTP(S)-методы</b>	POST

**Метод POST SeatmapStandalone**

**Описание:** получает информацию о схеме мест в самолёте без предварительно созданного бронирования.

**Формат запроса:** api/apiVersion/SeatmapStandalone

**Тело запроса**

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

**Параметры запроса**

Параметр	Описание
<b>passengers</b>	Массив с информацией о пассажирах
id	Идентификатор пассажира
firstName	Имя пассажира
middleName	Отчество пассажира

surname	Фамилия пассажира
passengerType	Тип пассажира. Возможные значения: Приложение 2. Справочник типов пассажиров
birthDate	Дата рождения. По данному полю вычисляется доступность или недоступность мест в аварийном ряду.
sex	Пол Возможные значения: Приложение 2. Справочники
loyaltyCard	Номер карты лояльности (если имеется)
<b>disability</b>	Сведения о людях с ограниченными возможностями
isDeaf	Нарушение слуха
isBlind	Нарушение зрения
hasGuideDog	Необходимо сопровождение собаки-поводыря, должно отправляться только вместе с признаком isBlind
assistanceType	Тип сопровождения пассажира Возможные значения: справочник типов сопровождения
wheelchairType	Тип инвалидного кресла Возможные значения: справочник типов инвалидных кресел
commentary	Комментарий (не обязателен)
hasInfant	Признак того, летит ли пассажир с младенцем
origin	Аэропорт отправления
destination	Аэропорт назначения
departureDate	Дата и время отправления
flightNumber	Номер рейса
classOfService	Информация о выбранном тарифе  Данные берутся из ответа api/flights:  data.outboundFlights.flights.flightNumber  или  data.returnFlights.flights.flightNumber

**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
}

```

## Параметры ответа

Параметр		Описание
success		Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>		Данные, которые вернул запрос
	<b>isEmpty</b>	Признак вернулись ли данные из Amadeus. По этому полю можно определить доступность выбора мест на указанном сегменте.
<b>compartmentDetails</b>		Массив объектов с информацией о местах по рядам
	number	Номер ряда
	<b>seats</b>	Массив объектов, который содержит подробную информацию по местам
	seat	Место. Возможные значения: A-F
	number	Номер ряда для места
	description	Описание места 9 – центральное место A – aisle seat, место у прохода W – window seat, место у окна
	characteristics	Характеристики места Возможные значения: <a href="#">[link]</a> справочник характеристик
	occupation	Признак занятости места Возможные значения: O – occupied, занято
	<b>passengerRelatedPrice</b>	Массив объектов с информацией о цене мест для конкретных клиентов. Цены присутствуют только для мест с характеристикой CH = Chargeable seat, платное место.
	name	Имя пассажира

			type	Тип пассажира Возможные значения: [link] справочник типов пассажиров
			uniqueCustomerIdentifier	Уникальный номер пассажира
			seatPrice	Цена места
			isSeatUnavailable	Флаг доступности места
			hasInfant	Признак того, летит ли пассажир с младенцем
			classDesignator	Информация о классе, к которому принадлежит место Возможные значения: ECONOMY, BUSINESS
			<b>passengersTotalData</b>	Массив объектов с информацией о стоимости мест в разрезе пассажиров
			passengerReference	Ссылка на пассажира
			<b>passengerPriceLevels</b>	Массив объектов стоимостей мест, доступных для пассажира
			price	Стоимость места
			priceLevel	Категория стоимостей мест
			<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
			ibeCode	Код ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>
			description	Описание ошибки, см. <a href="#">Приложение 1. Коды ошибок IBE</a>
			<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
			code	Код ошибки Amadeus
			text	Текст ошибки Amadeus
			<b>additionalInfo</b>	Дополнительная информация по ошибкам
			ssrType	Тип ssr
			info	Информация

### Примеры успешных запросов и ответов

## Сервис получения каталога услуг без PNR

<b>Сервис API</b>	CatalogueStandalone
<b>Краткое описание</b>	Сервис позволяет получать информацию о доступных дополнительных услугах без созданного бронирования
<b>Поддерживаемые HTTP(S)-методы</b>	POST
<b>Используемые методы смежных систем</b>	<b>Amadeus:</b> Service_StandaloneCatalogue

### Метод POST CatalogueStandalone

**Описание:** получает информацию о каталоге доступных дополнительных услуг без предварительно созданного бронирования

**Формат запроса:** api/apiVersion/CatalogueStandalone

### Тело запроса

```
{
  "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
  }
]
}

```

## Параметры запроса

Параметр	Описание
<b>flights</b>	Объект с информацией о перелётах
<b>outboundFlights</b>	Массив объектов StandaloneInputFlight с информацией о перелётах в на
<b>returnFlights</b>	Массив объектов StandaloneInputFlight с информацией о перелётах в на
<b>fares</b>	Объект с информацией о тарифах

	<b>outboundFareFamily</b>	Объект с информацией о тарифах для направления "туда"
	code	Код семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.code
	category	Категория семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.category
	isSubsidized	Признак субсидированного тарифа.
	<b>returnFareFamily</b>	Объект с информацией о тарифах для направления "обратно"
	code	Код семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.code
	category	Категория семейства тарифа. Данные берутся из ответа api/flights: data.fareFamilies.category
	isSubsidized	Признак субсидированного тарифа.
	<b>fares</b>	Массив с информацией о тарифах для направлений "туда" и "обратно"
	passengerType	Тип пассажира. Возможные типы приведены в справочнике "Типы пассажиров" в Прил
	flightIndex	Номер сегмента для конкретного перелёта (нумерация с 1)
	fareBasis	Название базового тарифа. Данные берутся из ответа api/flights: data.recommendations.outboundFare data.recommendations.returnFareDetails.fareBasis
	isBreakpoint	Признак того, заканчивает ли тариф своё действие на текущем сегменте Данные берутся из ответа api/flights: data.recommendations.outboundFare data.recommendations.returnFareDetails.isBreakpoint
	<b>passengers</b>	Объект с информацией о пассажирах
	id	Порядковый номер пассажира (нумерация с 1) (обязательный параметр
	firstName	Имя пассажира

	middleName	Отчество пассажира
	surname	Фамилия пассажира
	passengerType	Тип пассажира (обязательный параметр) Возможные значения: справочник типов пассажиров
	birthDate	Дата рождения
	sex	Пол пассажира Возможные значения: справочник "пол пассажиров"
	loyaltyCard	Номер карты лояльности «Крылья»
	<b>disability</b>	Сведения о людях с ограниченными возможностями
	isDeaf	Нарушение слуха
	isBlind	Нарушение зрения
	hasGuideDog	Необходимо сопровождение собаки-поводыря, должно отправляться то isBlind
	assistanceType	Тип сопровождения пассажира Возможные значения: справочник типов сопровождения
	wheelchairType	Тип инвалидного кресла Возможные значения: справочник типов инвалидных кресел
	commentary	Комментарий (не обязателен)
	hasInfant	Признак наличия у данного пассажира младенца

### StandaloneInputFlight

Параметр	Описание
origin	Аэропорт отправления
destination	Аэропорт прибытия

departureDate	<p>Дата и время отправления.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.outboundFlights.flights.departureDateTime</p> <p>или</p> <p>data.returnFlights.flights.departureDateTime</p> <p>Формат: уууу-ММ-ддТНН:мм:сс</p> <p>Пример: 2018-06-20T01:00:00</p>
arrivalDate	<p>Дата и время прибытия.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.outboundFlights.flights.arrivalDateTime</p> <p>или</p> <p>data.returnFlights.flights.arrivalDateTime</p> <p>Формат: уууу-ММ-ддТНН:мм:сс</p> <p>Пример: 2018-06-20T01:00:00</p>
flightNumber	<p>Номер рейса.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.outboundFlights.flights.flightNumber</p> <p>или</p> <p>data.returnFlights.flights.flightNumber</p>
classOfService	<p>Информация о выбранном тарифе.</p> <p>Данные берутся из ответа api/flights:</p> <p>data.recommendations.outboundFareDetails.bookingCode</p> <p>или</p> <p>data.recommendations.returnFareDetails.bookingCode</p>

marketingCarrier	Маркетинговая авиакомпания Например: U6, S7
operatingCarrier	Авиакомпания, по факту исполняющая перелёт Например: U6, ОК
flightIndex	Номер сегмента для конкретного перелёта (нумерация с 1)

### Формат ответа

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

```

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

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
<b>services</b>	Массив объектов с информацией о доступных услугах авиакомпании
code	Код услуги Возможные значения: справочник доп. услуг
<b>flightReferences</b>	Массив с информацией о сегментах и пассажирах, для которых доступны услуги, а также о стоимости этих услуг.
referenceNumber	Ссылка на сегмент перелёта
<b>passengerPrices</b>	Массив объектов с информацией о пассажире и стоимости услуги
referenceNumber	Ссылка на конкретного пассажира
amount	Стоимость услуги для конкретного пассажира
<b>packs</b>	Массив с доступными пакетами услуг
includedSeats	Массив характеристик доступных мест в рамках пакета услуг
excludedSeats	Массив характеристик недоступных мест в рамках пакета услуг
includedServiceCode	Код включенной в пакет услуги
code	Код пакета услуг. Возможные значения: Приложение 2. Справочник пакетов услуг

		<b>flightReferences</b>	Массив с информацией о сегментах и пассажирах, для которых доступны услуги, а также о стоимости этих услуг.
		referenceNumber	Ссылка на сегмент перелета
		<b>passengerPrices</b>	Массив объектов с информацией о пассажире и стоимости услуги
		referenceNumber	Ссылка на конкретного пассажира
		amount	Стоимость услуги для конкретного пассажира
		<b>serviceOptions</b>	Массив с информацией о том, какие услуги нужно автоматически предвыбирать на фронтенде движка бронирования.
		category	Категория услуги
		code	Код услуги Возможные значения: справочник доп. услуг
		flightReferences	Массив с информацией о сегментах, для которых доступны услуги
		preselect	Служебное поле. Если true, то услугу необходимо автоматически предвыбирать в UI. <i>Примечание: логика изначально добавлялась для использования в Web-клиенте.</i>
		priority	Служебное поле. Используется вместе с полем preselect. Означает приоритет автоматического добавления услуги в UI, 0 - наивысший. Например, если услуга с приоритетом 0 недоступна для предвыбора, то добавится услуга с приоритетом 1 и т.д.
		<b>seatServices</b>	Массив объектов с информацией о доступных местах для выбора на рейсе. <b>ВАЖНО:</b> Сервис api/seatmap возвращает более полную информацию о доступных местах и должен использоваться для бронирования услуги «выбор места».



		seatCharacteristics	Массив характеристик места. Возможные значения: Приложение 2. Справочник характеристик мест
		code	Код услуги, SEAT
		<b>flightReferences</b>	Массив с информацией о сегментах и пассажирах, для которых доступны услуги, а также о стоимости этих услуг.
		referenceNumber	Ссылка на сегмент перелёта
		<b>passengerPrices</b>	Массив объектов с информацией о пассажире и стоимости услуги
		referenceNumber	Ссылка на конкретного пассажира
		amount	Стоимость услуги для конкретного пассажира
		<b>petsServiceConfiguration</b>	Массив объектов, которые содержат информацию о доступности сервиса «провоз животных» на разных сегментах. Необходим для контроля провоза животных на тарифах семейства «бизнес».
		petsList	Массив строк, содержит список доступных к провозу животных
		flightReferences	Массив строк, содержит список сегментов, на которых доступна услуга
		<b>insuranceProducts</b>	Массив объектов с информацией о доступных страховках для выбора
		code	Код страховки Возможные значения: справочник доп. услуг (категория insurance)
		amount	Стоимость страховки
		<b>serviceCategories</b>	Массив объектов, содержит информацию о сервисах и порядке их отображения на сайте.
		code	Код услуги Возможные значения: справочник доп. услуг
		category	Категория услуги
		order	Порядок расположения на странице дополнительных услуг. 1 – верх страницы

	fullDirection	Флаг, обозначающий распространяется ли услуга на весь перелет
	<b>partnerProducts</b>	Массив объектов, содержит информацию о партнёрских сервисах.
	partnerId	Уникальный ID партнёра в движке IBE. Альфастрахование – 1, Sealine страхование – 2, Аэроэкспресс – 4
	partnerType	Тип партнёра. Страхование – 1, Аэроэкспресс – 2
	code	Код партнёрской услуги. Для каждого партнёра заведены уникальные коды услуг. См. коды партнёрских услуг в приложении 5.
	amount	Стоимость партнёрской услуги.
	currency	Валюта тарификации.
	bookingStatus	Статус партнёрской услуги. Возможные значения: 0 - неизвестно, 1 - услуга не выпущена, 2 - услуга выпущена, 3 - выпуск завершился с ошибкой, 4 - выполнен возврат, 5 - услуга не может быть выпущена
	passengerReference	Ссылка на пассажира, для которого вернулась услуга
	isInfantProduct	Флаг, обозначающий предоставляется ли услуга младенцу
	bookingId	Уникальный Id заказа партнёрской услуги в данном бронировании
	internalError	Код ошибки при получении партнёрской услуги
	internalErrorMessage	Сообщение об ошибке при получении партнёрской услуги

	<b>portions</b>	Служебное поле
	type	Служебное поле
	number	Служебное поле
	flightReferenceNumbers	Служебное поле
	<b>moneyWallCoefficient</b>	Коэффициент, для расчета ежемесячного платежа при оплате в рассрочку
	<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus
	<b>additionalInfo</b>	Массив с дополнительной информацией об ошибке
	ssrType	Тип ssr
	info	Дополнительная информация

## Примеры успешных запросов и ответов

Приложение X. Примеры ответов от сервисов

## Сервис управления сертификатами

<b>Сервис API</b>	certificate
<b>Краткое описание</b>	Сервис позволяет получать информацию о сертификатах и удалять их для текущего бронирования
<b>Поддерживаемые HTTP(S)-методы</b>	GET, DELETE
<b>Используемые методы смежных систем</b>	нет

### Метод GET api/certificate

**Описание:** Получение информации о сертификате

**Формат запроса** /api/apiVersion/certificate

**Параметры запроса:** в хедере передается X-Session

Параметр	Обязательность	Описание
certNumber	да	Номер сертификата EMD-S
surname	да	Фамилия владельца сертификата

**Формат ответа**

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

**Параметры ответа**

Параметр	Описание
<b>success</b>	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
number	Номер сертификата
surname	Фамилия владельца сертификата

	amount	Сумма сертификата в валюте RUB
	notValidAfter	Дата завершения срока действия сертификата
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

### Метод DELETE api/certificate

**Описание:** позволяет удалять примененный сертификат.

**Формат запроса** /api/apiVersion/certificate

### Параметры запроса

Отсутствуют. Бронирование для отмены определяется по ID сессии, переданной в заголовке X-Session.

### Формат ответа

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

## Параметры ответа

Параметр	Описание
<b>success</b>	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

## Метод GET api/certificate/CheckCertificate

**Описание:** Получение информации о сертификате

**Формат запроса** /api/apiVersion/certificate/CheckCertificate

## Параметры запроса:

Параметр	Обязательность	Описание
certNumber	да	Номер сертификата EMD-S
surname	да	Фамилия владельца сертификата

## Формат ответа

```

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

```

### Параметры ответа

Параметр	Описание
<b>data</b>	Данные, которые вернул запрос
<b>documents</b>	Информация о сертификате
number	Номер сертификата
surname	Фамилия владельца сертификата
status	Статус сертификата (open - открыт, close - закрыт)
notValidAfter	Дата, до которой действует сертификат
amount	Сумма сертификата
currency	Валюта (RUB)
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus

	text	Текст ошибки Amadeus
<b>success</b>		Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true



# Описание служебных методов API

## Сервис для продления сессии Amadeus

<b>Сервис API</b>	renew (служебный)
<b>Краткое описание</b>	Позволяет продлевать сессию Amadeus для предварительного бронирования.
<b>Поддерживаемые HTTP(S)-методы</b>	POST

### Метод POST renew

**Описание:** продлевает Stateful-сессию Amadeus для предварительного бронирования. Метод необходимо использовать, если хочется дать пользователю больше времени на заполнение контактных данных.

Вызов метода не имеет смысла, если бронирование создаётся сразу со всеми данными: о перелётах, пассажирах и тарифах.

**Формат запроса** /api/apiVersion/renew

### Параметры запроса

Отсутствуют. Необходимая информация о сессии Амадеуса находится в контексте бронирования.

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

### Примеры неуспешных запросов и ответов

Не удалось найти параметры сессии в контексте

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

### Сервис для получения списка аэропортов

Сервис API	airports (служебный)
Краткое описание	Позволяет получить список аэропортов, в которые осуществляет перелёты компания U6
Поддерживаемые HTTP(S)-методы	GET

### Метод GET airports

**Описание:** возвращает список городов и аэропортов вместе с соответствующими iata-кодами.

**Формат запроса** /api/apiVersion/airports

### Параметры запроса

Отсутствуют. Не нуждается в предварительном получении сессии.

### Формат ответа

```
{
  "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
}
```

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
country	Название страны, в которой находится город
cityRu	Название города на русском
cityEn	Название города на английском

	iataCity	IATA-код города
	airportRu	Название аэропорта на русском
	airportEn	Название аэропорта на английском
	iataAirport	IATA-код аэропорта
<b>errors</b>		Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

## Сервис для проверки работоспособности движка бронирования

<b>Сервис API</b>	health (служебный)
<b>Краткое описание</b>	Позволяет получить статус по работе сервисов движка бронирования
<b>Поддерживаемые HTTP(S)-методы</b>	GET

### Метод GET health

**Описание:** возвращает информацию о сервисах и о проценте предупреждений / ошибок, которые возникли для них за установленный период. Также содержит информацию о количестве созданных и оплаченных бронирований за тот же период.

На данный момент период равен 10 минутам.

### Формат запроса /api/health

Обратите внимание, что для вызова этого сервиса не требуется указывать версию API.

### Параметры запроса

Отсутствуют. Не нуждается в предварительном получении ID сессии.

### Формат ответа

```

{
  "data": {
    "endpointsHealth": [{
      "endpoint": string,
      "severity": integer,
      "sessionsCount": integer,
      "errorsPercentAvg": integer,
      "warningsPercentAvg": integer,
      "state": integer
    }
  ],
  "bookingStats": {
    "retrievedBookingsCount": integer,
    "createdBookingsCount": integer
  }
},
"errors": [],
"success": boolean
}

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул запрос
<b>endpointsHealth</b>	Массив объектов с информацией по сервисам
endpoint	Название сервиса Список сервисов: [link] справочник
severity	Важность сервиса 1 – низкая, 2 – средняя, 3 – высокая  Важность сервисов: [link] справочник  Высокую важность имеют сервисы, работоспособность которых влияет на создание бронирования.

	sessionsCount	Количество запросов для данного сервиса за период
	errorsPercentAvg	Средний процент ошибок для сервиса
	warningsPercentAvg	Средний процент предупреждений для сервиса
	state	<p>Состояние сервиса</p> <p>Возможные состояния для сервиса:</p> <p>1 – ОК, нормальное состояние сервиса.</p> <p>2 – Warning, в сервисе наблюдаются не критичные проблемы.</p> <p>Статус 2 для сервиса устанавливается в следующих случаях:</p> <ol style="list-style-type: none"> <li>1. Процент предупреждений больше 25%</li> <li>2. Количество сессий меньше 10 и процент ошибок больше 25%</li> <li>3. Количество сессий больше 10 и процент ошибок от 25% до 75%</li> </ol> <p>3 – Error, корректная работа сервиса нарушена</p> <p>Статус 3 для сервиса возвращается в следующих случаях:</p> <p>Количество сессий больше 10 и процент ошибок больше 75%.</p>
	bookingStats	Объект с информацией о бронированиях
	retrievedBookingsCount	Количество оплаченных бронирований за период
	createdBookingsCount	Количество созданных бронирований за период
	<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
	ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus

text	Текст ошибки Amadeus
------	----------------------

### Детальное описание работы сервиса

1. Сервис api/health запрашивает информацию через REST API для Azure Application Insights, таймаут ожидания для этого запроса установлен в 10 секунд. Через это время IBE либо вернёт ожидаемую информацию о состоянии сервисов, либо ошибку с кодом ibeCode = 1009. После получения такого кода ошибки необходимо снова запросить информацию у сервиса.
2. Сервис api/health возвращает информацию об ошибках / предупреждениях в процентах по следующей причине: может возникнуть ситуация, когда ошибка возникла у одного клиента (то есть в рамках одной «сессии» движка бронирования, токена) и он многократно её воспроизводит. Например, нажимает кнопку «продолжить», хотя получил уже ошибку обновления бронирования.

Это негативно сказывается на общей статистике, т.к. остальные клиенты могут вообще не наблюдать подобных ошибок. Поэтому возвращается не общее количество ошибок, а процент ошибок по каждому сервису. Сначала ошибки для сервиса группируются по сессии пользователя, а затем находится средний процент ошибок.

1. Информация о работе сервисов запрашивается за определённый период. На данный момент этот период равен 10 минутам. Например, если сервис вызвали в 11:17:10, значит информация будет запрашиваться с 11:07:10.
2. Из-за того, что информация по работе сервисов поступает в Application Insights не мгновенно, рекомендуется запрашивать сервис api/health примерно раз в 2-3 минуты. При возникновении проблем возможен более частый вызов сервиса.
3. Если сервис api/health возвращает ошибки с кодом 5xx, значит движок бронирования полностью неработоспособен.

### Сервис отправки отчёта о проблеме

<b>Сервис API</b>	problemNotification (служебный)
<b>Краткое описание</b>	Позволяет пользователям отправлять отчёт о проблеме
<b>Поддерживаемые HTTP(S)-методы</b>	POST

### Метод POST problemNotification

**Описание:** позволяет пользователям отправлять отчёт о проблеме. На фронте доступен при получении «общей технической ошибки» с кодом 1000.

## Формат запроса /api/apiVersion/problemNotification

### Тело запроса

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

### Параметры запроса

Параметр	Описание
from	Отправитель. Любая строка, идентифицирующая пользователя.
description	Описание проблемы.
details	Технические детали проблемы. Например, ID сессии, в которой возникла проблема.

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
errors	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок ИВЕ



	description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
	messages	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
	code	Код ошибки Amadeus
	text	Текст ошибки Amadeus

### Примеры успешных запросов и ответов

Успешная отправка информации об ошибке

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

### Сервис для получения курса валют

<b>Сервис API</b>	exchangerates (служебный)
<b>Краткое описание</b>	Позволяет получить информацию по курсам валют
<b>Поддерживаемые HTTP(S)-методы</b>	GET

#### Метод GET exchangerates

**Описание:** запрашивает информацию по курсам валют.

**Формат запроса** /api/apiVersion/exchangerates

#### Параметры запроса

Отсутствуют. Не нуждается в предварительном получении ID сессии.

#### Формат ответа

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

```

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

```

## Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true
<b>data</b>	Данные, которые вернул сервис
name	Название валюты, например, «Azerbaijan Manat»
code	Код валюты, например, «AZN»
value	Текущий курс валюты к рублю
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

## Пример ответа

```

{
  "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
}

```

## Сервис для получения правил транслитерации кириллицы

<b>Сервис API</b>	settings/transliteration (служебный)
-------------------	--------------------------------------

<b>Краткое описание</b>	Позволяет получить правила транслитерации кириллицы
<b>Поддерживаемые HTTP(S)-методы</b>	GET

### Метод GET settings/transliteration

**Описание:** запрашивает правила транслитерации кириллицы

**Формат запроса** /api/apiVersion/settings/transliteration

### Параметры запроса

Отсутствуют. Не нуждается в предварительном получении ID сессии.

### Формат ответа

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

### Параметры ответа

Параметр	Описание
success	Признак успешности выполнения запроса. Если массив ошибок errors пуст, то значение поля будет true

<b>data</b>	Данные, которые вернул сервис
nativeLetter	Символ кириллицы
latinLetter	Символ латиницы
<b>errors</b>	Информация об ошибках, возникших во время работы сервиса.
ibeCode	Код ошибки, см. Приложение 1. Коды ошибок IBE
description	Описание ошибки, см. Приложение 1. Коды ошибок IBE
<b>messages</b>	Массив объектов, содержащий ошибки из ответа Amadeus. Массив может быть пуст, рекомендуется анализировать ошибку по полю ibeCode.
code	Код ошибки Amadeus
text	Текст ошибки Amadeus

### Пример ответа

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

## Сервис для отслеживания состояния резервного контура

<b>Сервис API</b>	system/Availability
<b>Краткое описание</b>	Позволяет отслеживать состояния резервного контура движка в IT Grad
<b>Поддерживаемые HTTP(S)-методы</b>	GET

**Описание:** Метод является аналогом уже существующего метода healthcheck. Сделан специально для проверки состояния сервисов на резервном контуре движка.

Проверяет:

1. БД - наличие соединения
2. Redis - наличие соединения
3. Amadeus - получение сессии по логин/паролю
4. SIG - авторизацию, запрос флайтов
5. Rabbit - наличие соединения
6. CRM - наличие соединения, три эндпоинта: скидки, авторизация, рест
7. Partners API - доступность Aeroexpress, Alfastrah, Sealine
8. GOI - наличие соединения
9. ВРС - наличие соединения
10. Artsofte - справочник аэропортов, getHash

**Формат запроса:** /api/system/Availability

**Параметры запроса:** Отсутствуют.

**Формат ответа:**

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

**Параметры ответа**

Параметр	Описание
----------	----------

service	название сервиса
severityLevel	важность сервиса Low = 1, Medium = 2, High = 3
isAvailable	признак доступности сервиса

Последнее поле isAvailable показывает общую работоспособность движка и принимает значение true, только когда доступны все сервисы с severityLevel = 3

### Сервис для отслеживания состояния резервного контура

<b>Сервис API</b>	system/Availability
<b>Краткое описание</b>	Позволяет отслеживать состояния резервного контура движка в IT Grad
<b>Поддерживаемые HTTP(S)-методы</b>	GET

**Описание:** Метод является аналогом уже существующего метода healthcheck. Сделан специально для проверки состояния сервисов на резервном контуре движка.

Проверяет:

1. БД - наличие соединения
2. Redis - наличие соединения
3. Amadeus - получение сессии по логин/паролю
4. SIG - авторизацию, запрос флайтов
5. Rabbit - наличие соединения
6. CRM - наличие соединения, три эндпоинта: скидки, авторизация, рест
7. Partners API - доступность Aeroexpress, Alfastrah, Sealine
8. GOI - наличие соединения
9. ВРС - наличие соединения
10. Artsofte - справочник аэропортов, getHash

**Формат запроса:** /api/system/Availability

**Параметры запроса:** Отсутствуют.

**Формат ответа:**

{

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

### Параметры ответа

Параметр	Описание
service	название сервиса
severityLevel	важность сервиса Low = 1, Medium = 2, High = 3
isAvailable	признак доступности сервиса

Последнее поле isAvailable показывает общую работоспособность движка и принимает значение true, только когда доступны все сервисы с severityLevel = 3



# Приложения

## Приложение 1. Коды ошибок IBE

Код ошибки	Описание, возвращаемое IBE	Описание ошибки
<b>Ошибки IBE</b>		
1000	Unknown IBE error	Необработанное исключение, причина неизвестна.
1001	Operation can not be performed due to invalid reservation state	Ошибка возникает, если какой-то метод API был вызван в момент, неподходящий по процессу бронирования.  Например, если метод api/services добавления услуг был вызван сразу после api/session (бронирование не было создано).
1002	IBE session pool waiting timeout exceeded	Ошибка может возникать в момент, когда движок пытается получить сессию Амадеуса, но пул сессий не отвечает.
1003	IBE session pool stateful sessions limit exceeded	Ошибка возникает в момент, когда допустимое количество Stateful-сессий Амадеус исчерпано. Stateful-сессии используются для создания предварительного бронирования.
1004	Session pool cleaning operation error	Ошибка удаления сессий, время жизни которых истекло.
1005	Session pool is unable to acquire new session from Amadeus	Невозможно получить новую сессию Амадеус.
1006	Failed to resolve timezone	Ошибка может возникать при неудачной попытке обновления справочника таймзон.

1007	Session pool is unable to acquire stateful session as it's expired	Движок пытается получить информацию по сессии в Session Pool, но такой сессии нет.
1008	Data validation error	Валидационная ошибка, может возникать на всех этапах бронирования.  Например, при отправке неверных данных пассажиров.
1009	IBE health check failed	Невозможно получить информацию от метода api/health.  Метод использует Azure для получения данных.
1010	Promotion is not found	Промокод не найден.
1011	Operation can not be performed due to invalid payment order status	Ошибка возникает в случае попытки модифицировать бронирование в той же сессии после получения ссылки на оплату, чтобы не было двух одновременных заказов.
1012	Unable to cancel additional service with issued ticket	Ошибка возникает при попытке отменить услуги при помощи сервиса api/services, если отменяемая услуга была оплачена и по ней выпущен билет.
1013	Unable to load IBE reservation context	Невозможно получить информацию по ID открытой сессии движка бронирования.
1014	IBE authentication error	Ошибка возникает при одновременной передаче в запрос заголовка Authorization и X-API-Key, что невозможно.
1015	File type not allowed	Ошибка возникает в случае попытки передачи в методы движка бронирования файлов, которые не подходят по типам.

1016	File size is too large	Ошибка возникает в случае попытки передачи в методы движка бронирования файлов, которые имеют большой размер.
1017	Request body is too large	Ошибка возникает в случае, когда тело запроса слишком большое.
1018	Unable to change more then maximum allowed symbols	<p>Ошибка возникает в случае попытки изменить в информации о пассажире символов больше, чем разрешено.</p> <p>На данный момент используется для проверки при изменении фамилии и имени.</p>
1019	Unable to change element twice	<p>Ошибка возникает в случае повторной попытки изменения пассажирских данных.</p> <p>На данный момент используется для проверки при изменении фамилии и имени.</p>
1020	Failed to update or retrieve search log data	Служебная ошибка. Возникает при неудачной попытке записи логов в базу данных.
1021	Failed to update or retrieve reservation log data	Служебная ошибка. Возникает при неудачной попытке чтения логов из базы данных.
1022	Failed connect to redis	Служебная ошибка. Возникает при неудачной попытке подключения к Redis.
1030	Unknown Refund error	Неизвестная ошибка при инициализации процесса возврата.
1031	Refund sessions store has failed to process an operation	Ошибка возникает в случае неудачной попытки чтения/записи информации о возврате в базу данных.

1032	Refund process is locked by another process	Ошибка возникает в случае попытки одновременного возврата по одному PNR.
1033	There are incomplete refund processes	Ошибка возникает в случае, если один из процессов возврата по PNR находится не в конечном статусе. См. раздел с описанием сервиса api/refund.
1040	The ticket refund info was not found	Ошибка возникает в случае, если состояние билетов в базе данных IBE и в PNR на стороне Amadeus различается.
1041	Some passengers are not exist in reservation	Ошибка возникает в случае, если переданные в запросе пассажиры не были найдены в бронировании.
1042	The remaining passengers are of the wrong type	Ошибка возникает в случае, если оставшиеся в бронировании пассажиры имеют неподходящий тип. Например, при попытке отменить взрослого пассажира в бронировании с 1 взрослым и 1 ребёнком.
1043	The PNR was created not in IBE	Ошибка возникает в случае, если бронирование было создано не в движке бронирования (без использования IBE API).
1044	One of the specified refund items is not found or not available for the confirmation	Ошибка возникает в случае, если невозможно подтвердить возврат по одному из билетов (сервис GOI-проху, PUT refund/pnr)
1050	The partner refund process returned error	Ошибка возникает при неуспешной попытке отменить партнёрские услуги.
1060	The refund session has a problem	Ошибка возникает в случае, если возврат бронирования не удалось довести до конечного статуса.

1080	OrderInfo client returned error	Ошибка возникает в случае, если GOI на стороне U6 вернул ошибку.
1081	OrderInfo client returned unauthorized error	Ошибка возникает в случае, если сервис GOI вернул 401 ошибку (не авторизован).
1500	Failed to receive exchange rates.	Невозможно получить информацию от метода api/exchangerates. Метод использует API ЦБ РФ для получения данных.
1600	Send grid error	Ошибка отправки письма о технической ошибке через Send Grid.
1800	Unknown social network error	Необработанное исключение, причина неизвестна.
1801	Social network connectivity error	Ошибка может возникать при попытке авторизации через социальные сети.
1802	Social network connection timeout exceeded	Ошибка может возникать при попытке авторизации через социальные сети, возник таймаут обращения к системам.
2000	Unknown Connector error	Необработанное исключение, причина неизвестна.
2200	Airports service error	Невозможно получить информацию от метода api/airports. Метод использует API <a href="http://ub.ru">ub.ru</a> для получения данных.
2201	Airports cache is empty	Метод api/airports не вернул данные.
2202	Airports API returned invalid result	Метод api/airports вернул неверные данные.
3000	Unknown Soap receiver service error	Неизвестная ошибка при загрузке рейсов-исключений через сервис IBE Soap receiver.

5000	Certificate is invalid	Метод api/Certificate/CheckCertificate вернул информацию о том, что запрашиваемый сертификат не найден. Например, при некорректно указанном номере сертификата или фамилии владельца
<b>Ошибки Amadeus</b>		
1100	Unknown Amadeus error	Необработанное исключение, причина неизвестна.
1101	Amadeus connectivity error	Возникает при ошибке подключения к сервисам Амадеус.
1102	Amadeus connection timeout exceeded	Возникает в случае, когда сервис Амадеуса не отвечает за установленный таймаут.
1103	Amadeus Service_IntegratedCatalogue function returned error	Возникает в случае, когда произошла ошибка при получении списка дополнительных услуг от Амадеус.
1104	Amadeus Insurance_Pricing function returned error	Возникает в случае, когда произошла ошибка получения информации по страховке, заведённой в Амадеус.
1105	Amadeus Fare_PricePNRWithBookingClass function returned error	Возникает в случае, когда произошла ошибка подтверждения цены билетов в бронировании (прайсинга).
1106	Amadeus Service_IntegratedPricing function returned error	Возникает в случае, когда произошла ошибка подтверждения цены дополнительных услуг в бронировании.
1107	Amadeus Insurance_Booking function returned error	Возникает в случае, когда произошла ошибка бронирования страховки, заведённой в Амадеус.

1108	Amadeus Ticket_CreateTSTFromPricing function returned error	Ошибка создания TST для бронирования.
1109	Amadeus Ticket_CreateTSMFromPricing function returned error	Ошибка создания TSM для бронирования.
1110	Amadeus PNR_AddMultiElements function returned error	Ошибка добавления/удаления информации в бронирование.
1111	Amadeus PNR_Retrieve function returned error	Ошибка получения информации по бронированию.
1112	Amadeus PNR_Cancel function returned error	Ошибка отмены бронирования.
1113	Amadeus Air_RetrieveSeatMap function returned error	Ошибка получения информации о рассадке на борту.
1114	Amadeus Ticket_RetrieveListOfTSM function returned error	Ошибка получения информации о дополнительных услугах АК в бронировании.
1117	Amadeus Fare_PricePNRWithBookingClass function return error. No fare for booking code	Ошибка возникает при попытке забронировать тариф, билеты по которому отсутствуют.
1118	Amadeus Ticket_DisplayTSMP function returned error	Возникла ошибка в методе Ticket_DisplayTSMP
1119	Amadeus Ticket_UpdateTSMP function returned error	Возникла ошибка в методе Ticket_UpdateTSMP
1120	Amadeus service returned fault of type Session	Использована неверная сессия Amadeus
1121	Amadeus PNR_Split function returned error	Возникла ошибка в методе PNR_Split
1122	Unable to split PNR if children are left in PNR	Служебная ошибка. На данный момент нигде не используется.
1130	Unknown Refund error	Неизвестная ошибка, возникшая в Amadeus при попытке вернуть билеты по бронированию.
1131	Unable to refund	Невозможно вернуть билеты

1132	Amadeus DocRefund_IgnoreRefund function returned error	Возникла ошибка в методе DocRefund_IgnoreRefund (метод вызывается после расчёта информации к возврату через Amadeus).
1133	No ATC tickets to refund	Ошибка возникает при попытке инициировать добровольный возврат для бронирования с невыпущенными билетами.
1150	Amadeus PNR not found. No match for record locator.	Ошибка возникает при попытке запросить информацию по несуществующему бронированию.  Связана с ошибкой 1931 "Application NO MATCH FOR RECORD LOCATOR" Amadeus.
<b>Ошибки CRM</b>		
1200	Unknown CRM error	Необработанное исключение, причина неизвестна.
1201	CRM connectivity error	Возникает при ошибке подключения к сервисам CRM.
1202	CRM connection timeout exceeded	Возникает в случае, когда сервис CRM не отвечает за установленный таймаут.
1203	CRM HoldPromoCode function returned error	Ошибка холдирования промокода.
1204	CRM AgreePersonalInfo function returned error	Ошибка сервиса по отправке информации о подписке на рассылку.
1205	CRM CheckPromoCode function returned error	Ошибка проверки промокода.
1206	CRM CalculateDiscount function returned error	Ошибка сервиса получения скидки.
1207	CRM REST API getUser method returned error	Ошибка сервиса авторизации в CRM.
1208	CRM CalculateDiscountList function returned error	Ошибка сервиса получения списка скидок.



1209	On this phone already created involuntary refund	Ошибка возникает в случае, если вынужденный возврат уже инициирован по данному номеру телефона.
1210	Unknown Involuntary refund error	Неизвестная ошибка при вынужденном возврате.
1211	Involuntary refund not found	Вынужденный возврат не найден.
1400	CRM receipt service returned error.	Ошибка сервиса получения маршрутной квитанции.
1401	CRM receipt send email service returned error.	Ошибка сервиса отправки маршрутной квитанции на email.
1402	CRM receipt timeout exceeded.	Таймаут от сервиса получения/отправки маршрутной квитанции.
<b>Ошибки ВРС</b>		
1300	Unknown Payment gate error	Необработанное исключение, причина неизвестна.
1301	Payment gate hash value is incorrect	Неправильный хэш по заказу.
1302	Unknown Payment History error	Ошибка возникает в случае, если невозможно получить информацию по истории заказов от ВРС.  Сервис используется в МУБ при докупке дополнительных услуг.
1303	Payment History Login returned error	Ошибка возникает в случае, если не удалось авторизоваться в сервис ВРС.
1304	Payment History By Pnr returned error	Ошибка возникает в случае при неудачной попытке получить историю заказов по бронированию.
<b>Ошибки взаимодействия с TAIS</b>		

1700	Unknown Tais error	Необработанное исключение, причина неизвестна.
1701	Tais Connection Timeout	Возникает в случае, когда сервис ТАИС не отвечает за установленный таймаут.
1702	Tais Connectivity Error	Ошибка подключения к сервисам ТАИС.
1703	Tais Ftp server return error.	Ошибка отправки отчёта на FTP-сервер ТАИС
<b>Ошибки при получении партнёрских услуг</b>		
1900	Aeroexpress services return error	Сервис Аэроэкспресс вернул ошибку.
1901	Aeroexpress connection timeout exceeded	Возникает в случае, когда сервис Аэроэкспресс не отвечает за установленный таймаут.
1902	Aeroexpress connectivity error	Ошибка подключения к сервисам Аэроэкспресс.
1950	Unknown Partner	Ошибка возникает, если ID партнёра неизвестен.
1951	Empty partner products list	Служебная ошибка. Возникает в случае, если при выгрузке отчёта в SOFI невозможно получить список партнёрских услуг из бронирования.
1952	Partner product wasn't issued	Ошибка возникает в случае, если партнёрские билеты не выпущены, и пользователь пытается инициировать их переотправку.
1953	Unknown partner product booking ID	Ошибка возникает в случае, если по ID невозможно найти партнёрскую услугу в бронировании.

1957	A partner type is not available for current App Type	Ошибка возникает при попытке добавить в бронирование партнёрскую услугу для типа приложения, для которого она недоступна. Например, Альфастрахование для мобильного приложения.
<b>Ошибки DMS</b>		
2100	Unknown DMS error	Необработанное исключение, причина неизвестна.
2101	DMS connection timeout exceeded	Возникает в случае, когда сервис DMS не отвечает за установленный таймаут.
<b>Ошибки OrderInfo (проxy-сервис для GOI)</b>		
4000	OrderInfo proxy service has failed to process request	Возникает в случае, когда проxy не смог обработать поступивший запрос.
4001	OrderInfo proxy service has not found requested reservation or failed to load it	Не удалось найти запрошенное бронирование или загрузить информацию по нему.
4002	OrderInfo proxy service has failed to prepare partners products for issue operation	Ошибка возникает в случае, если не удалось подготовить партнёрские услуги к выпуску.
4003	OrderInfo proxy service has failed to issue partners products	Не удалось выпустить партнёрские услуги.
4004	OrderInfo proxy service has failed to decrease fop amount on insurance amount	Ошибка возникает в случае, если не удалось уменьшить стоимость бронирования на сумму партнёрских услуг.
4100	OrderInfo service has failed to process request	Ошибка возникает в случае, если GOI-проxy не смог выполнить запрос.

**Коды ошибок при добавлении услуг в бронирование (только для версии API старше 2.1)**

<b>Код ошибки</b>	<b>Описание, возвращаемое IBE</b>	<b>Описание ошибки</b>
-------------------	-----------------------------------	------------------------

1000	Unknown error	Неизвестная ошибка. Любая ошибка, которая не обрабатывается движком.
1001	Invalid passenger or segment reference	Ошибка возникает при попытке добавить услугу для пассажира или на сегмент, который отсутствует в бронировании.
2001	Seat is unavailable for this passenger type.	Ошибка возникает при попытке забронировать для ребёнка место на аварийном ряду или при попытке забронировать место для взрослого с младенцем не в центральном ряду.
2002	Seat is unavailable	Ошибка возникает при попытке забронировать ранее занятое место.
3001	Unknown insurance code	Ошибка возникает при попытке добавить в бронирование страховку с неверным кодом.
3002	Catalogue of insurance services is unavailable	Ошибка возникает, когда при добавлении страховки в бронирование недоступен сервис страховок Амадеус.
4001	Catalogue of services is unavailable	Ошибка возникает, когда при добавлении услуг в бронирование недоступен сервис Catalogue Амадеус.
4002	Invalid service code	Ошибка возникает при попытке добавить в бронирование услугу с неверным кодом.

5000	Invalid product code	Ошибка возникает при попытке добавить в бронирование партнёрскую услугу с неверным кодом.
5001	A partner type is not available for current App Type.	Ошибка возникает при попытке добавить в бронирование партнёрскую услугу для типа приложения, для которого она недоступна.

## Приложение 2. Справочники

### Типы пассажиров

Значение поля <code>раxType</code>	Расшифровка	Тип пассажира
0	ADT, взрослый	Обычный
1	SRC, пенсионер	Субсидированный
2	YTH, молодёжь	Субсидированный
3	CHD, ребёнок	Обычный
4	INF, младенец	Обычный
5	CNN, ребёнок	Субсидированный
6	UNN, ребёнок	Обычный, несопровождаемый ребёнок
7	STR, взрослый	Субсидированный для Дальнего Востока

### Пол пассажиров

Значение поля <code>sex</code>	Расшифровка
0	Мужской
1	Женский

### Типы документов

Значение поля <code>documentType</code>	Расшифровка
0	Паспорт (общегражданский)
1	Заграничный паспорт

2	Свидетельство о рождении
3	Иной

### Типы сопровождения пассажиров

Значение поля assistanceType	Расшифровка
0	Без сопровождения
1	Нужно инвалидное кресло по пути в/из самолета
2	Нужна помощь, чтобы подняться и спуститься с самолета
3	Пункты 1 и 2 вместе

### Типы инвалидных кресел

Значение поля wheelchairType	Расшифровка
0	Ручной тип кресла
1	Электрический тип с сухой батареей
2	Электрический с батареей из жидкостных элементов

### Типы дополнительных услуг

Значение поля services.code	Категория	Расшифровка
ABAG	luggage	Первая дополнительная сумка
BBAG	luggage	Вторая дополнительная сумка
CBAG	luggage	Третья дополнительная сумка
HBAG	luggage	Увеличение дополнительной сумки или дефолтного багажа (если доступно)
CHML	meal	Child meal, детское питание
HTML	meal	Hot meal, горячее (мясное) питание
MOML	meal	Muslim halal meal, халяльное питание
SNML	meal	Sandwich meal, сендвич
SFML	meal	Sea food meal, рыбное питание

VGML	meal	Vegeterian meal, вегетарианское питание
CAML	meal	Dessert, десерт
KSML	meal	Kosher, кошерное питание
BGML	meal	Burger, Бургер
AVML	meal	Vegetarian Hindu, Вегетарианское хинду
HNML	meal	Hindu, Хинду
VJML	meal	Vegetarian jain, Джайн
TCML	meal	Hot breakfast, Горячий завтрак
GFML	meal	Gluten free, Безглютеновое
BBML	meal	Baby meal, Для младенцев
CPML	meal	Cheese platter, Сырная тарелка
MPML	meal	Meat platter, Мясная тарелка
GRML	meal	Greek salad, Греческий салат
MTML	meal	Meat Steak Meal, Стейк из мяса
UDML	meal	Seafood Udon Meal, Удон с морепродуктами
FTML	meal	Fish Steak Meal, Стейк из рыбы
ECML	meal	English Breakfast, Английский завтрак
FCML	meal	French Breakfast, Французский завтрак
ACML	meal	American Breakfast, Американский завтрак
RCML	meal	Russian Breakfast, Русский завтрак
CSML	meal	Caucasian Cuisine, Кавказская кухня
RUML	meal	Russian Cuisine, Русская кухня
ORML	meal	Asian Cuisine, Азиатская кухня
MEML	meal	Italian Cuisine, Итальянская кухня
GEML	meal	German Cuisine, Немецкая кухня
ASML	meal	Central Asian Cuisine, Среднеазиатская кухня
PETC	pets	Провоз животного
URFL	insurance	Страховка, авиационный пакет
URTR	insurance	Страховка, трэвел-пакет

SEAT	seat	Выбор места
BLOU	room	Бизнес-зал
VLOU	room	Vip-зал
CVID	covidExpress	Экспресс-анализ на covid-19
XBLL	handLuggage	Ручная кладь до 10 кг в салон (багаж в салон)
XBLS	handLuggage	Дополнительное место ручной клади до 10 кг в салон

**Типы дополнительных услуг, поддерживаемые движком (не продаются в движке)**

Значение поля serviceCategories.code	Категория	Расшифровка
CBBG	luggage	Багаж в салоне самолета с отдельным местом
BIKE	luggage	Велосипед в багажном отсеке
SKI	luggage	Горнолыжное снаряжение
BULK	luggage	Дополнительная ручная кладь
XBAG	luggage	Негабаритный багаж
HBGS	luggage	Сверхнормативный багаж
HBGT	luggage	Сверхнормативный багаж
DBGS	luggage	Увеличение веса багажа до 32 кг
DBGT	luggage	Увеличение веса багажа до 32 кг
FRAG	luggage	Хрупкий багаж
WBAG	luggage	оформление платы за превышение веса регистрируемого багажа с 10 до 23 кг
SRXBLL	luggage	оформление платы за превышение веса ручной клади с 5 до 10 кг, не превышающей габариты 55*40*20
SRXBLS	luggage	оформление платы за дополнительное место ручной клади, весом не более 5 кг и не превышающей габариты 55*40*20
MEQT	luggage	Медицинское оборудование
WEAP	luggage	Оружие
GOLF	luggage	Снаряжение для игры в гольф



SPEQ	luggage	Спортивное снаряжение
RVML	meal	Raw vegetarian meal, Вегетарианское питание
VLML	meal	Vegetarian lacto-ovo meal, Вегетарианское питание
VOML	meal	Vegetarian oriental meal, Вегетарианское питание
DBML	meal	Диабетическое питание
BLML	meal	Диетическое питание
LCML	meal	Низкокалорийное питание
FPML	meal	Питание из фруктов
LFML	meal	Питание с пониженным содержанием жира
NLML	meal	Питание с пониженным содержанием лактозы
LSML	meal	Питание с пониженным содержанием соли
NFML	meal	Питание, не содержащее рыбу
SPML	meal	Специальное питание
JPML	meal	Японское питание
CFML	meal	Кофе натуральный капучино
CCML	meal	Кофе натуральный американо
AVIH	pets	Животное в багажном отсеке
SVAN	pets	Животное в салоне
EXST	seat	Дополнительное свободное место рядом
STRQ	seat	SEAT REQUEST, Место
GPST	seat	Group seat request, specify number and preferred location, Место
RQST	seat	Seat request-include seat number or preference, Место
CHNM	data	CHANGE NAME, Изменение имени
NAMM	data	Оплата сбора за изменение имени пассажира в билете
NAME	data	Name - when airline holds reservation under a different name, Изменение имени пассажира

PFEE	data	Оплата сбора за переоформление в call-центре
UMNR	other	Несопровождаемый ребенок
BUSS	other	Трансфер автобусом
TIME	other	Услуга отложенной оплаты
URFO	insurance	Авиа
NSP1	insurance	Отмена поездки с суммой страхового покрытия 5000 руб
NSP2	insurance	Отмена поездки с суммой страхового покрытия 10000 руб
NSP3	insurance	Отмена поездки с суммой страхового покрытия 20000 руб
NSP4	insurance	Отмена поездки с суммой страхового покрытия 30000 руб
NSP5	insurance	Отмена поездки с суммой страхового покрытия 40000 руб

### Пакеты услуг

Значение поля packs.code	Значение поля packs.includedServiceCode	Описание и значения поля packs.includedSeats	Расшифровка
ПАКА	SFML	COMF SEAT, [L, E]	Место повышенной комфортности + Рыбное питание
ПАКВ	TCML	COMF SEAT, [L]	Место повышенной комфортности + Горячий завтрак
ПАКС	PETC	CH SEAT, [CH, W, A]	Стандартное платное место + Питомец

PAKD	ABAG	COMF SEAT, [L,E]	Место повышенной комфортности + Багаж, 23 кг
PAKE	ABAG	CH SEAT, [CH, W, A]	Стандартное платное место + Багаж, 23 кг
PAKF	HTML	COMF SEAT, [L,E]	Место повышенной комфортности + Горячее питание
PKDP	ABAG	COMF SEAT, [L,E]	Место повышенной комфортности + Багаж, 23 кг для ПРОМО тарифа
PKEP	ABAG	CH SEAT, [CH, W, A]	Стандартное платное место + Багаж, 23 кг для ПРОМО тарифа
PAKR	RUML	COMF SEAT, [L,E]	Место повышенной комфортности + Русская кухня
PAKZ	ECML	COMF SEAT, [L,E]	Место повышенной комфортности + Английский завтрак

#### Типы животных

Значение поля services.text	Расшифровка
DOG	Собака
CAT	Кошка
PARROT	Попугай

CANARY	Канарейка
--------	-----------

### Наиболее частые характеристики мест для Seatmap

Значение поля characteristics	Расшифровка
A	Aisle seat, место у прохода
CH	Chargeable seat, платное место
E	Exit row seat, место у выхода
I	Seat suitable for adult with an infant, место для пассажиров с младенцами
L	Leg space seat, место с дополнительным пространством для ног
V	Seat to be left vacant or offered last, места, которые предпочтительно предлагать последними
8	No seat at this location, нет места

Полный список характеристик можно найти в документе Amadeus для сервиса Air\_RetrieveSeatMap 14.2.

### Справочник статусов промокодов

Значение поля promotionStatus	Расшифровка
0	New, новый
1	Used, погашен
2	Sent, отправлен

3	Saved, сохранён
4	Received, получен
5	Hold, холд

### Опции тарифов

Код опции	Категория	Расшифровка
bestSeats	availablePaid	Рассадка на первых рядах салона экономического класса (до 15 го ряда) при наличии свободных мест на момент регистрации
upTo5bonuses	bonus	Начисление бонусов до 5% бонусов
upTo10bonuses	bonus	Начисление бонусов до 10% бонусов
upTo15bonuses	bonus	Начисление бонусов до 15% бонусов
flexibilityOnChangesAndCancellations	ffServices	Возможность возврата и обмена билета
refundExchangeProhibited	ffServices	Возврат билета запрещён
handLuggage5kg	handBag	Ручная кладь 5 кг
handLuggage15kg	handBag	Ручная кладь 15 кг
oneSuitcase10kg	luggage	Один чемодан 10 кг
oneSuitcase23kg	luggage	Один чемодан 23 кг
oneSuitcase32kg	luggage	Один чемодан 32 кг
twoSuitcase32kg	luggage	Два чемодана 32 кг
mealsOnBoard	menu	Стандартное питание
boarding	priorityBoarding	Приоритетная посадка в самолёт
checkin	priorityCheckIn	Приоритетная регистрация на рейс
businessClassCatering	specialMenu	Питание Бизнес-класса

upgradeToBusinessClass	upgradeToBusiness	Повышение обслуживания до класса Бизнес
vipLounges	vipLounge	Бизнес-зал
twoSeatsOnRow	vipSeats	Рассадка по двое на блок из трех кресел
vipSeats	vipSeats	Кресло Бизнес-класса
noMealsOnBoard	menu	Без питания

### Основные категории семейств тарифов

Категория Amadeus	Категория SIG	Расшифровка
ECOLIGHT	PR	Тариф «промо»
ECOFLEX	EC	Тариф «эконом»
ECOPREM	FL	Тариф «премиум-эконом»
BUSFLEX	BS	Тариф «бизнес»
COMFORT	CF	Тариф «комфорт»
PRLIGHT	PL	Тариф «промо-лайт»
BAGLESS	LC1	Тариф «безбагажный»
LUGGAGE	LC2	Тариф «багажный»
SUBSIDIZED	SUBSY	Тариф «субсидированный»

Тариф бизнес-лайт BUSLIGHT был удалён.

### Важность сервисов движка бронирования

№	Название сервиса	Код	Расшифровка
1	Airports/Get	1	Низкая
2	Calendar/Get	2	Средняя
3	Catalogue/Get	2	Средняя
4	Discounts/Get	2	Средняя
5	Flights/Get	3	<b>Высокая</b>
6	Households/Get	1	Низкая

7	Payment/Get	2	Средняя
8	Payment/Post версия сервиса для API 1.0	3	<b>Высокая</b>
9	Payment/PostV2 версия сервиса для API 1.1	3	<b>Высокая</b>
10	Pricing/Post	3	<b>Высокая</b>
11	Promotions/Post	1	Низкая
12	Promotions/Delete	1	Низкая
13	Receipt/Get	2	Средняя
14	Receipt/Post	2	Средняя
15	Renew/Post	2	Средняя
16	Reservation/Get	3	<b>Высокая</b>
17	Reservation/Post	3	<b>Высокая</b>
18	Reservation/Delete	2	Средняя
19	Seatmap/Get	2	Средняя
20	Services/Post	2	Средняя
21	ServicesPricing/Post	2	Средняя
22	Token/Get	3	<b>Высокая</b>
23	Token/Post	2	Средняя
24	User/Post	2	Средняя

Высокую важность имеют сервисы, работоспособность которых влияет на создание бронирования.

### Типы офисов Amadeus

№ п/п	ID офиса в Amadeus	Тип офиса	Описание
1	SVXU6040A (устарел), SVXU608MP (актуальный)	1	Мобильный офис, бронирование через мобильное приложение
2	SVXU608AZ, SVXU608AA	2	Веб-офис, бронирование через сайт

3	SVXU608SS	2	Веб-офис, бронирование, созданное переходом из Skyscanner
4	SVXU608AS	2	Веб-офис, бронирование, созданное переходом из AviaSales
5	MOWU608AK	3	IVR, офис для колл-центра
6	SVXU608YA	2	Веб-офис, бронирование, созданное переходом из Яндекс.Авиа

## Страны

Код страны	Название на русском	Название на английском
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
LYB	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

### Приложение 3. Ремарки/служебные SSR

Движок бронирования для технических целей добавляет в PNR ремарки. Ниже перечислены типы, формат ремарок и их назначение.

Служебные SSR для скидок

**SSR с информацией о применённой скидке.**

**Описание:** SSR добавляются при бронировании тарифа со скидкой

**Формат:** DISCOUNT/DSNN/UsageType/CampaignId/LESSNN/SegmentReference/Passenger Reference

**Примеры:**

DISCOUNT/DS30/0/1-45RT6KP/LESS1972/S2/P2

DISCOUNT/DS07/0/1-45S5YUL/LESS363/S1/P2

Поле	Описание	Комментарий
DISCOUNT	Константа	-
DSNN	DS - константа NN - процент скидки	Примеры: DS05, DS15

UsageType	Тип применения скидки	Fare = 0, на тариф Order = 1, на заказ
CampaignId	Уникальный Id кампании скидок в CRM	Пример: 1-45RT6KP
LESSNN	NN - размер скидки в рублях	Пример: LESS1972
SegmentRefernese	Ссылка на сегмент	Скидка добавляется на все сегменты, либо не добавляется вовсе.  В PNR добавляются SSR для всех сегментов.
PassengerReference	Ссылка на пассажира	-

### SSR с информацией о карте "Крылья"

**Описание:** SSR добавляются при бронировании тарифа со скидкой

**Формат:** DISCOUNTCARD/CardNumber

CardNumber - карта крылья

**Пример:** DISCOUNTCARD/1000444608

Служебные SSR для промокодов

**Описание:** SSR добавляется при применении промокода

**Формат SSR для промокода на**

**перелёт:** PROMO/Promocode/CampaignId/DSNN/LESSNN/SegmentRefernese/PassengerReference

**Пример:** PROMO/IBE21/1-45H4OIC/DS06/LESS1264/S1/P2

Поле	Описание	Комментарий
PROMO	Константа	-
Promocode	Примененный промокод	Пример: IBE21

CampaignId	Уникальный Id промо кампании в CRM	Пример: 1-45H4OIC
DSNN	DS - константа NN - процент скидки	Примеры: DS05, DS15
LESSNN	NN - размер скидки в рублях	Пример: LESS1264
SegmentRefernece	Ссылка на сегмент	Скидка добавляется на все сегменты, либо не добавляется вовсе. В PNR добавляются SSR для всех сегментов.
PassengerReference	Ссылка на пассажира	-

#### Формат SSR для промокода на

услугу: PROMOCODE/ServiceCode/Promo/CampaignId/DSNN/LESSNN/SegmentRefernece/PassengerReference

**Пример:** PROMOCODE/CHML/CHML01/1-45SADSN/DS07/LESS35/S1/P2

Поле	Описание	Комментарий
PROMOCODE	Константа	-
ServiceCode	Код услуги	Возможные коды услуг описаны в приложении 2
Promo	Применённый промокод	-
CampaignId	Уникальный Id промо кампании в CRM	Пример: 1-45H4OIC
DSNN	DS - константа NN - процент скидки	Примеры: DS05, DS15
LESSNN	NN - размер скидки в рублях	Пример: LESS1264

SegmentReferenece	Ссылка на сегмент	Скидка добавляется на все сегменты, либо не добавляется вовсе.  В PNR добавляются SSR для всех сегментов.
PassengerReference	Ссылка на пассажира	-

#### Ремарка для категории тарифа

**Описание:** ремарка добавляется для возможности отображения опций тарифа на последней странице после оплаты.

**Формат:** outboundFare/returnFare

#### Примеры:

Для One way: "ECOFLEXRU/"

Для Return Trip: "ECOFLEXRU/BUSFLEXRU"

Постфикс "RU" может отличаться в зависимости от выбранного тарифа.

Основные категории семейств тарифов можно найти в [Приложении 2](#).

#### Служебные SSR для партнёрских услуг

**Описание:** SSR добавляется при добавлении партнерской услуги в бронирование для отправки информации по партнёрской услуге в sbr файле

**Формат (аналогичен формату ремарки):** PTR/PartnerId/ServiceId/Code/OrderId/Currency/Amount/Status/RefundAmount/PassengerReference/InfantFlag

#### Ремарка для партнёрских услуг

**Описание:** ремарка добавляется для отправки информации в BPC/CRM о забронированной партнёрской услуге.

**Формат:** PTR/PartnerId/ServiceId/Code/OrderId/Currency/Amount/Status/RefundAmount/PassengerReference/InfantFlag

**Пример**

(Аэроэкспресс): "PTR/4/65241A17717E4BEEB378B82763795A03/STANDARTOW/520501/RUB/500.00/1/0.00/P2/"

Поле	Описание	Комментарий
PTR	Константа	-
PartnerId	Уникальный идентификатор партнера	Возможные типы описаны в <a href="#">приложении 5</a>
ServiceId	Уникальный идентификатор заказа	GUID
Code	Код партнёрской услуги	Возможные типы описаны в <a href="#">приложении 5</a>
OrderId	Идентификатор заказа на оформление услуги в системе партнёра	-
Currency	Валюта тарификации	RUB
Amount	Стоимость услуги	Дробное число

Status	<p>Статус услуги.</p> <p>Возможные значения:</p> <p>NotIssued = 1, услуга добавлена в бронирование, ожидает оформления</p> <p>Issued = 2, услуга оформлена, получен номер отчётного документа</p> <p>IssueFailed = 3, услуга добавлена в бронирование, но во время оформления произошла ошибка</p> <p>Refunded = 4, услуга отменена, запрошен возврат средств</p> <p>IssueError = 5, услуга не может быть выпущена из-за тех. ошибки</p>	<p>В ремарке используются числовой код статуса.</p> <p>Если статус услуги изменяется, то сначала удаляется ремарка со старым кодом, затем добавляется ремарка с новым кодом.</p>
RefundAmount	Стоимость к возврату	Если возврат не был оформлен, поле имеет значение 0.00
PassengerReference	Ссылка на пассажира, для которого оформлена услуга	Если услуга относится к пассажиру, а не ко всему заказу, то при создании ремарки добавляются ссылки на соответствующих пассажиров.

InfantFlag	Принимает значение “I”, если услуга относится к инфанту	Для аэроэкспресса: никогда не заполняется
------------	---	---

**Ремарка для применённых наценок**

**Описание:** ремарка добавляется при применении наценок на дополнительные услуги авиакомпании.

Наценки загружаются из системы DMS.

**Формат:**

PRICE/ServiceCode/ChargeAmount/SegmentReference/PassengerReference/ServiceReference/ChargeType

**Пример:** PRICE/ABAG/107/S1/P2/SSR12/1

Поле	Описание	Комментарий
PRICE	Константа	-
ServiceCode	Код услуги	Возможные коды услуг описаны в приложении 2
ChargeAmount	Размер наценки в рублях	Только целые значения
SegmentReference	Ссылка на сегмент, на который была добавлена услуга с наценкой	-

PassengerReference	Ссылка на пассажира, для которого была добавлена услуга с наценкой	-
ServiceReference	Ссылка на доп. услугу, на которую была добавлена наценка	-
ChargeType	Тип добавленной наценки	Возможные значения: 1 - наценка DMS 2 - динамическая наценка, полученная из getContactMarkup

#### Приложение 4. Партнёрские услуги.

##### Коды партнёров

Код партнёра	Название
1	Альфастрахование
2	Sealine страхование
3	Служебный
4	Аэроэкспресс
5	Ostrovok
6	TUI
7	Телемедицина (Sberhealth)
8	Упаковка багажа
9	Трансфер



10	Аренда авто
11	Альфастрахование (Антиковид)
12	Sealine (Страхование АнтиВирус)

### Типы партнёров

Тип партнёра	Название
1	Страхование
2	Аэроэкспресс
3	Медицина

### Коды услуг Аэроэкспресс

№	Код услуги	Название тарифа
1	STANDARTOW	Стандарт

Описание тарифов можно найти на официальном сайте Аэроэкспресс: <https://aeroexpress.ru/aero/prices.html>

### Коды услуг Альфастрахование

№	Код услуги	Название тарифа
1	AVIA	Авиа
2	TRAVEL	Трэвел
3	LUGGAGE	Багаж
4	SPORT	Спорт
5	NOFLIGHT	Невылет
6	AVIADYNAMIC	Авиа с динамическим расчетом стоимости

7	AVIADYNAMICBUS	Авиа для Бизнес тарифов с динамическим расчетом стоимости
---	----------------	---

Описание тарифов <https://www.uralairlines.ru/passengers-info/partner-services/insurance/>

#### Коды услуг Альфастрахование (Антиковид)

№	Код услуги	Название тарифа
1	ANTICOVID19	Антиковид

#### Коды услуг Sealine

№	Код услуги	Название тарифа	Доступность
1	SLAVIA	АВИА	Да
2	SLAVIAPLUS	АВИА+	Да
3	SLSPORT	СПОРТ	Да
4	SLPROMO	ПРОМО	Да
5	SLTRAVEL	ТРЕВЭЛ	Да

#### Коды услуг Sealine (Страхование от COVID)

№	Код услуги	Название тарифа
1	SLANTICOVID	Страхование АнтиВирус

#### Коды услуг Sberhealth

№	Код услуги	Название тарифа	Доступность
1	SBERDOC	Терапевт и Педиатр Онлайн	Да
2	SBERVET	Ветеринар Онлайн	Да

## **Приложение X. Примеры ответов от сервисов**

### **Примеры использования поля `flightDurationUntilNextStop`**

Запрос:

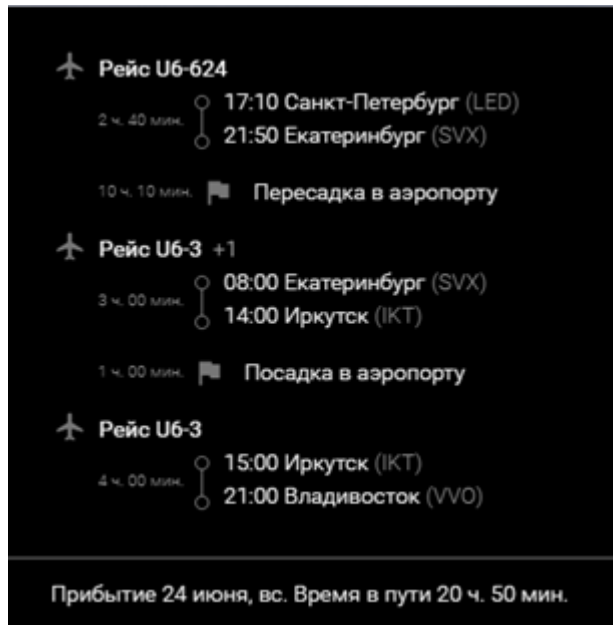
```
api/flights?orig=LED&dest=VVO&dep=20180623&ret=&adt=1&chd=0&inf=0&src=0&yth=0
```

Разберём один из элементов массива `outboundFlights`:

```

{
  "flightGroupId": 1,
  "flights": [{
    "flightId": 0,
    "departureAirport": "LED",
    "departureTerminal": "1",
    "arrivalAirport": "SVX",
    "departureDateTime": "2018-06-23T17:10:00",
    "arrivalDateTime": "2018-06-23T21:50:00",
    "aircraft": "Airbus A320",
    "flightDuration": "2:40",
    "stopDuration": "10:10",
    "flightNumber": "624",
    "marketingCarrier": "U6",
    "operatingCarrier": "U6",
    "hash": "801455404"
  }, {
    "flightId": 1,
    "departureAirport": "SVX",
    "arrivalAirport": "VVO",
    "departureDateTime": "2018-06-24T08:00:00",

```



```
"arrivalDateTime": "2018-06-24T21:00:00",

"aircraft": "Airbus A321",

"techStops": [{

  "arrivalDateTime": "2018-06-24T14:00:00",

  "departureDateTime": "2018-06-24T15:00:00",

  "airportCode": "IKT",

  "stopDuration": "1:00",

  "flightDurationUntilNextStop ": "4:00"

}],

"flightDuration": "8:00",

"flightDurationUntilNextStop": "3:00",

"flightNumber": "3",

"marketingCarrier": "U6",

"operatingCarrier": "U6",

"hash": "1931975080"

},

"flightGroupDuration": "20:50",

"hash": "1072566894"

}
```

Первый и последний аэропорты в списке соответствуют `flights[0] -> departureAirport (LED)` и `flights[1] -> arrivalAirport (VVO)`.

Аналогичным образом соответствует начальное время вылета `flights[0] -> departureDateTime (17:10)` и `flights[1] -> arrivalDateTime (21:00)` для этих аэропортов.

Если перелёт не содержит технических остановок, то его длительность берётся из поля `flights -> flightDuration` (как для первого перелёта, `"flightDuration": "2:40"`), а время пересадки из `flights -> stopDuration ("stopDuration": "10:10")`.

Второй перелёт содержит техническую остановку, поэтому значение `"flightDuration": "8:00"` мы получим, если сложим все значения `flightDurationUntilNextStop` (время перелёта до следующей остановки) и `stopDuration` (время остановки).

Время перелёта SVX – ИКТ берётся из `flights[1] -> flightDurationUntilNextStop = 03:00`, что совпадает с вычисленным значением: `flights[1] -> techStops -> arrivalDateTime (14:00)` вычесть `flights[1] -> departureDateTime (08:00)`, то есть  $14:00 - 08:00 = 06:00$  минус три часа разницы, равно 3 часа.

Далее время остановки берётся из поля `flights[1] -> techStops -> stopDuration`.

Следующее время перелёта берётся из поля `flights[1] -> techStops -> flightDurationUntilNextStop`.

Значение `"flightGroupDuration": "20:50"` совпадает с суммой длительности всех перелётов и остановок:  $2:40 + 10:10 + 3:00 + 1:00 + 4:00$ .

Разберём ещё один пример:

```

{
  "flightGroupId": 2,
  "flights": [{
    "flightId": 0,
    "departureAirport": "LED",
    "departureTerminal": "1",
    "arrivalAirport": "DME",
    "departureDateTime": "2018-06-23T18:50:00",
    "arrivalDateTime": "2018-06-23T20:20:00",
    "aircraft": "Airbus A319",
    "flightDuration": "1:30",
    "stopDuration": "3:25",
    "flightNumber": "80",
    "marketingCarrier": "U6",
    "operatingCarrier": "U6",
    "hash": "395175226"
  }, {
    "flightId": 1,
    "departureAirport": "DME",
    "arrivalAirport": "SVX",
    "departureDateTime": "2018-06-23T23:45:00",

```

**Рейс U6-80**  
 1 ч. 30 мин. 18:50 Санкт-Петербург (LED) → 20:20 Москва (DME)  
 3 ч. 25 мин. **Пересадка в аэропорту**

**Рейс U6-271 +1**  
 2 ч. 20 мин. 23:45 Москва (DME) → 04:05 Екатеринбург (SVX)  
 3 ч. 55 мин. **Пересадка в аэропорту**

**Рейс U6-3**  
 3 ч. 00 мин. 08:00 Екатеринбург (SVX) → 14:00 Иркутск (IKT)  
 1 ч. 00 мин. **Посадка в аэропорту**

**Рейс U6-3**  
 4 ч. 00 мин. 15:00 Иркутск (IKT) → 21:00 Владивосток (VVO)

**Прибытие 24 июня, вс. Время в пути 19 ч. 10 мин.**

```
"arrivalDateTime": "2018-06-24T04:05:00",

"aircraft": "Airbus A320",

"flightDuration": "2:20",

"stopDuration": "3:55",

"flightNumber": "271",

"marketingCarrier": "U6",

"operatingCarrier": "U6",

"hash": "1569210211"

}, {

"flightId": 2,

"departureAirport": "SVX",

"arrivalAirport": "VVO",

"departureDateTime": "2018-06-24T08:00:00",

"arrivalDateTime": "2018-06-24T21:00:00",

"aircraft": "Airbus A321",

"techStops": [{

"arrivalDateTime": "2018-06-24T14:00:00",

"departureDateTime": "2018-06-24T15:00:00",

"airportCode": "IKT",

"stopDuration": "1:00",
```



<pre> "flightDurationUntilNextStop": "4:00"     }   ],   "flightDuration": "8:00",   "flightDurationUntilNextStop": "3:00",   "flightNumber": "3",   "marketingCarrier": "U6", "operatingCarrier": "U6",   "hash": "1931975080"     }   ],   "flightGroupDuration": "19:10",   "hash": "-1214789730" } </pre>	
---	--

Первый перелёт:

flights[0] -> departureAirport - LED

flights[0] -> arrivalAirport - DME

Нет технических остановок, поэтому длительность перелёта

flights[0] -> flightDuration - 1:30

Пересадка в аэропорту

flights[0] -> stopDuration - 3:25

Второй перелёт:

flights[1] -> departureAirport - DME

flights[1] -> arrivalAirport - SVX

Нет технических остановок, поэтому длительность перелёта

flights[1] -> flightDuration - 2:20

Пересадка в аэропорту

flights[1] -> stopDuration - 3:55

Третий перелёт:

flights[2] -> departureAirport - SVX

flights[2] -> arrivalAirport - VVO

Есть технические остановки (посадки), поэтому

Длительность первой части перелёта (SVX-ИКТ):

flights[2] -> flightDurationUntilNextStop - 3:00

Длительность технической остановки:

flights[2] -> techStops -> stopDuration - 1:00

Длительность второй части перелёта (ИКТ-VVO):

flights[2] -> techStops -> flightDurationUntilNextStop - 4:00

Сумма всех перелётов и остановок

$1:30 + 3:25 + 2:20 + 3:55 + 3:00 + 1:00 + 4:00 = 19:10$

**Сервис reservation, метод POST. Создание предварительного бронирования**

Перелёт в одну сторону
------------------------

Запрос, отправка данных о перелёте	Ответ
------------------------------------	-------

<pre> {   "flights": {     "outboundFlights": [{       "origin": "LED",       "destination": "SVX",       "departureDate": "2018-06-20T01:00:00.000+03:00",       "flightNumber": "3",       "classOfService": "M"     }   ],   "returnFlights": null,   "passengersCount": 1 } </pre>	<pre> { "data": {   "journey": {     "outboundFlights": [{       "aircraft": "320",       "referenceNumber": "1",       "origin": "LED",       "destination": "SVX",       "departureDate": "2018-06-20T01:00:00",       "arrivalDate": "2018-06-20T05:30:00",       "flightNumber": "3",       "classOfService": "M",       "marketingCarrier": "U6",       "operatingCarrier": "U6",       "flightDuration": "2:30"     } ],     "outboundFareFamily": { "code": "" },     "outboundDuration": "2:30",     "origin": "LED",     "destination": "SVX",     "departureDate": "2018-06-20T01:00:00"   },   "seats": [] } </pre>
--	--

	<pre>}, "errors": [], "success": true }</pre>
Запрос, отправка данных о пассажирах (1взрослый)	Ответ

<pre>{   "flights": null,   "passengers": {     "passengers": [{       "id": 1,       "firstName": "Тест",       "middleName": "Тестович",       "surname": "Тестов",       "passengerType": 0,       "birthDate": "1988-01-01",       "sex": 0,       "nationality": "RUS",       "documentNumber": "1409345890",       "documentType": 0,       "documentExpiryDate": null,       "loyaltyCard": "1000332032",       "phone": "89510001133",       "email": "test@yandex.ru",       "disability": null     }   ],   "userAgreement": true }</pre>	<pre>{   "data": {     "journey": {       "outboundFlights": [{         "aircraft": "320",         "referenceNumber": "1",         "origin": "LED",         "destination": "SVX",         "departureDate": "2018-06-20T01:00:00",         "arrivalDate": "2018-06-20T05:30:00",         "flightNumber": "3",         "classOfService": "M",         "marketingCarrier": "U6",         "operatingCarrier": "U6",         "flightDuration": "2:30"       }     ],     "outboundFareFamily": {       "code": ""     },     "outboundDuration": "2:30",     "origin": "LED",   } }</pre>
---	--

<pre>} }</pre>	<pre>"destination": "SVX",  "departureDate": "2018-06-20T01:00:00"  },  "number": "PNRNUMBER",  "reservationDate": "2018-06-01T15:23:00",  "passengers": [{    "id": 1,    "hasInfant": false,    "referenceNumber": "2",    "firstName": "TEST",    "middleName": "TESTOVICH",    "surname": "TESTOV",    "passengerType": 0,    "isAdult": true,    "birthDate": "1988-01-01T00:00:00",    "sex": 0,    "nationality": "RUS",    "documentNumber": "1409345890",    "documentExpiryDate": "2028-06-01T00:00:00",    "email": "TEST@YANDEX.RU",    "phone": "89510001133",    "documentType": 0</pre>
--------------------	--

	<pre> } ], "services": [], "seats": [] }, "errors": [], "success": true }  "number": "PNRNUMBER" – в ответе будет присутствовать реальный номер PNR </pre>
<b>Перелёт туда и обратно</b>	
Запрос, отправка данных о перелёте	Ответ



```

{
  "flights": {
    "outboundFlights": [{
      "origin": "LED",
      "destination": "DME",
      "departureDate": "2018-07-05T18:50:00.000+03:00",
      "flightNumber": "80",
      "classOfService": "M"
    }
  ],
  "returnFlights": [{
    "origin": "DME",
    "destination": "LED",
    "departureDate": "2018-07-15T11:20:00.000+03:00",
    "flightNumber": "79",
    "classOfService": "K"
  }
  ],
  "passengersCount": 2
}

```

```

{
  "data": {
    "journey": {
      "outboundFlights": [{
        "aircraft": "319",
        "referenceNumber": "1",
        "origin": "LED",
        "destination": "DME",
        "departureDate": "2018-07-05T18:50:00",
        "arrivalDate": "2018-07-05T20:20:00",
        "flightNumber": "80",
        "classOfService": "M",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightDuration": "1:30"
      }
    ],
    "outboundFareFamily": {
      "code": ""
    },
    "outboundDuration": "1:30",
    "returnFlights": [{

```

	<pre>"aircraft": "319", "referenceNumber": "2", "origin": "DME", "destination": "LED", "departureDate": "2018-07-15T11:20:00", "arrivalDate": "2018-07-15T12:50:00", "flightNumber": "79", "classOfService": "K", "marketingCarrier": "U6", "operatingCarrier": "U6",   "flightDuration": "1:30" } ], "returnDuration": "1:30", "origin": "LED", "destination": "DME", "departureDate": "2018-07-05T18:50:00", "returnDate": "2018-07-15T11:20:00" }, "seats": [], }, "errors": [],</pre>
--	---

	<pre>"success": true }</pre>
Запрос, отправка данных о пассажирах (1 взрослый и 1 ребёнок)	Ответ

<pre> {   "flights": null,   "passengers": {     "passengers": [{       "id": 1,       "firstName": "Иван",       "middleName": "ИВАНОВИЧ",       "surname": "ИВАНОВ",       "passengerType": 0,       "birthDate": "1988-01-01",       "sex": 0,       "nationality": "RUS",       "documentNumber": "1409345890",       "documentType": 0,       "documentExpiryDate": null,       "loyaltyCard": "1000332032",       "phone": "89510001133",       "email": "ivanov@<a href="mailto:ivanov@yandex.ru">yandex.ru</a>",       "disability": null     }, {       "id": 2,       "firstName": "Сергей", </pre>	<pre> {   "data": {     "journey": {       "outboundFlights": [{         "aircraft": "319",         "referenceNumber": "1",         "origin": "LED",         "destination": "DME",         "departureDate": "2018-07-05T18:50:00",         "arrivalDate": "2018-07-05T20:20:00",         "flightNumber": "80",         "classOfService": "M",         "marketingCarrier": "U6",         "operatingCarrier": "U6",         "flightDuration": "1:30"       }     ],     "outboundFareFamily": {       "code": ""     },     "outboundDuration": "1:30",     "returnFlights": [{ </pre>
---	--

<pre> "middleName": null, "surname": "Орлик", "passengerType": 3, "birthDate": "2007-12-12", "sex": 0, "nationality": "RUS", "documentNumber": "77123123", "documentType": 2, "documentExpiryDate": null, "loyaltyCard": null, "phone": null, "email": null, "disability": null } ], "userAgreement": true } } </pre>	<pre> "aircraft": "319", "referenceNumber": "2", "origin": "DME", "destination": "LED", "departureDate": "2018-07-15T11:20:00", "arrivalDate": "2018-07-15T12:50:00", "flightNumber": "79", "classOfService": "K", "marketingCarrier": "U6", "operatingCarrier": "U6", "flightDuration": "1:30" } ], "returnDuration": "1:30", "origin": "LED", "destination": "DME", "departureDate": "2018-07-05T18:50:00", "returnDate": "2018-07-15T11:20:00" }, "number": "PNRNUMBER", "reservationDate": "2018-06-01T16:08:00", "passengers": [{ </pre>
---	---

	<pre>"id": 1,  "hasInfant": false,  "referenceNumber": "3",  "firstName": "IVAN",  "middleName": "IVANOVICH",  "surname": "IVANOV",  "passengerType": 0,  "isAdult": true,  "birthDate": "1988-01-01T00:00:00",  "sex": 0,  "nationality": "RUS",  "documentNumber": "1409345890",  "documentExpiryDate": "2028-06-01T00:00:00",  "email": "IVANOV@YANDEX.RU",  "phone": "89510001133",  "documentType": 0  }, {  "id": 2,  "hasInfant": false,  "referenceNumber": "2",  "firstName": "SERGEY",  "middleName": "",</pre>
--	---

	<pre>"surname": "ORLIK", "passengerType": 3, "isAdult": false, "birthDate": "2007-12-12T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "77123123", "documentExpiryDate": "2028-06-01T00:00:00", "email": "<a href="mailto:IVANOV@YANDEX.RU">IVANOV@YANDEX.RU</a>", "phone": "89510001133", "documentType": 0 } ], "services": [], "seats": [] }, "errors": [], "success": true }</pre>
--	--

**Сервис reservation, метод GET. Получение информации о бронировании**

**Бронирование с дополнительными услугами и скидками. Перелёт в одну сторону**

**Строка запроса:** /api/v1/reservation?pnrNumber=MPJ65W&lastName=VETRODUEV

**Ответ:**

```

{
  "data": {
    "tickets": [
      {
        "text": "PAX 262-
2437100000/ETU6/RUB36481/03JUN18/SVXU608AA/00777534",
        "passengerReference": "2",
        "price": 36481
      }
    ],
    "outboundFareFamily": {
      "fareFamilyId": 0,
      "code": "ECOPREM",
      "options": [
        {
          "code": "handLuggage5kg",
          "category": "handBag",
          "order": 2,
          "showInPopup": true,
          "showInSummary": true,
          "showInUpsell": false,
          "showInFareIncluded": false,
          "showForAdult": true,
          "showForInfant": false
        },
        {
          "code": "mealsOnBoard",
          "category": "menu",
          "order": 3,
          "showInPopup": true,
          "showInSummary": false,
          "showInUpsell": false,
          "showInFareIncluded": false,
          "showForAdult": true,
          "showForInfant": false
        },
        {
          "code": "upgradeToBusinessClass",
          "category": "upgradeToBusiness",
          "order": 7,
          "showInPopup": true,
          "showInSummary": false,
          "showInUpsell": true,
          "showInFareIncluded": true,
          "showForAdult": true,
          "showForInfant": true
        },
        {
          "code": "flexibilityOnChangesAndCancellations",
          "category": "ffServices",
          "order": 5,
          "showInPopup": true,
          "showInSummary": false,
          "showInUpsell": false,
          "showInFareIncluded": false,
          "showForAdult": true,
          "showForInfant": true
        }
      ]
    }
  }
}

```



```

    },
    {
      "code": "upTo10bonuses",
      "category": "bonus",
      "order": 4,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": false,
      "showInFareIncluded": true,
      "showForAdult": true,
      "showForInfant": true
    },
    {
      "code": "bestSeats",
      "category": "availablePaid",
      "order": 6,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": true,
      "showInFareIncluded": true,
      "showForAdult": true,
      "showForInfant": true
    },
    {
      "code": "handLuggage5kg",
      "category": "handBag",
      "order": 2,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": false,
      "showInFareIncluded": true,
      "showForAdult": false,
      "showForInfant": true
    },
    {
      "code": "oneSuitcase10kg",
      "category": "luggage",
      "order": 1,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": false,
      "showInFareIncluded": true,
      "showForAdult": false,
      "showForInfant": true
    },
    {
      "code": "oneSuitcase23kg",
      "category": "luggage",
      "order": 1,
      "showInPopup": true,
      "showInSummary": true,
      "showInUpsell": false,
      "showInFareIncluded": false,
      "showForAdult": true,
      "showForInfant": false
    }
  ],

```

```

    "order": 0,
    "commercialFamily": "U6ECONOMY"
  },
  "tsmData": [
    {
      "passengerReference": "2",
      "price": 2000,
      "referenceType": "SSR",
      "referenceNumber": "20"
    },
    {
      "passengerReference": "2",
      "price": 2000,
      "referenceType": "SSR",
      "referenceNumber": "21"
    },
    {
      "passengerReference": "2",
      "price": 2000,
      "referenceType": "SSR",
      "referenceNumber": "22"
    },
    {
      "passengerReference": "2",
      "price": 2000,
      "referenceType": "SSR",
      "referenceNumber": "23"
    },
    {
      "passengerReference": "2",
      "price": 500,
      "referenceType": "SSR",
      "referenceNumber": "24"
    },
    {
      "passengerReference": "2",
      "price": 500,
      "referenceType": "SSR",
      "referenceNumber": "25"
    },
    {
      "passengerReference": "2",
      "price": 2500,
      "referenceType": "SSR",
      "referenceNumber": "26"
    },
    {
      "passengerReference": "2",
      "price": 2500,
      "referenceType": "SSR",
      "referenceNumber": "27"
    }
  ],
  "appliedDiscounts": [
    {
      "flightReference": "1",
      "amount": 6,

```

```

        "id": "1-1PSJTRU",
        "type": "DISCOUNT",
        "passengerReference": "2"
    },
    {
        "flightReference": "2",
        "amount": 6,
        "id": "1-1PSJTRU",
        "type": "DISCOUNT",
        "passengerReference": "2"
    }
],
"baseFarePrices": [
    {
        "currency": "RUB",
        "passengerReference": "2",
        "isInfantFare": false,
        "totalPrice": 36481,
        "basePrice": 32792
    }
],
"journey": {
    "outboundFlights": [
        {
            "aircraft": "Airbus A320",
            "referenceNumber": "1",
            "origin": "AER",
            "destination": "SVO",
            "departureDate": "2018-06-22T19:05:00",
            "arrivalDate": "2018-06-22T21:35:00",
            "flightNumber": "450",
            "classOfService": "Q",
            "marketingCarrier": "U6",
            "operatingCarrier": "U6",
            "flightDuration": "2:30",
            "stopDuration": "13:45"
        },
        {
            "aircraft": "Airbus A319",
            "referenceNumber": "2",
            "origin": "DME",
            "destination": "LED",
            "departureDate": "2018-06-23T11:20:00",
            "arrivalDate": "2018-06-23T12:50:00",
            "flightNumber": "79",
            "classOfService": "Q",
            "marketingCarrier": "U6",
            "operatingCarrier": "U6",
            "flightDuration": "1:30"
        }
    ],
    "outboundFareFamily": {
        "code": "ECOPREMURU"
    },
    "outboundDuration": "17:45",
    "origin": "AER",
    "destination": "LED",

```

```

    "departureDate": "2018-06-22T19:05:00"
  },
  "number": "PNRNUMBER",
  "reservationDate": "2018-06-03T11:57:00",
  "passengers": [
    {
      "id": 1,
      "hasInfant": false,
      "referenceNumber": "2",
      "firstName": "VLADISLAV",
      "middleName": "IVANOVICH",
      "surname": "VETRODUEV",
      "passengerType": 0,
      "isAdult": true,
      "birthDate": "1988-01-01T00:00:00",
      "sex": 0,
      "nationality": "RUS",
      "documentNumber": "1409345890",
      "documentExpiryDate": "2028-06-03T00:00:00",
      "email": "V.VETER@YANDEX.RU",
      "phone": "89510001133",
      "documentType": 0
    }
  ],
  "services": [
    {
      "code": "ABAG",
      "status": "HK",
      "category": "luggage",
      "passengerReference": "2",
      "flightReference": "1",
      "referenceNumber": "20",
      "price": 2000
    },
    {
      "code": "ABAG",
      "status": "HK",
      "category": "luggage",
      "passengerReference": "2",
      "flightReference": "2",
      "referenceNumber": "21",
      "price": 2000
    },
    {
      "code": "BBAG",
      "status": "HK",
      "category": "luggage",
      "passengerReference": "2",
      "flightReference": "1",
      "referenceNumber": "22",
      "price": 2000
    },
    {
      "code": "BBAG",
      "status": "HK",
      "category": "luggage",
      "passengerReference": "2",

```

```

    "flightReference": "2",
    "referenceNumber": "23",
    "price": 2000
  },
  {
    "code": "SFML",
    "status": "HK",
    "category": "meal",
    "passengerReference": "2",
    "flightReference": "1",
    "referenceNumber": "24",
    "price": 500
  },
  {
    "code": "SFML",
    "status": "HK",
    "category": "meal",
    "passengerReference": "2",
    "flightReference": "2",
    "referenceNumber": "25",
    "price": 500
  },
  {
    "code": "PETC",
    "status": "HK",
    "category": "pets",
    "text": "DOG 1PC 8KG 45X35X25",
    "passengerReference": "2",
    "flightReference": "1",
    "referenceNumber": "26",
    "price": 2500
  },
  {
    "code": "PETC",
    "status": "HK",
    "category": "pets",
    "text": "DOG 1PC 8KG 45X35X25",
    "passengerReference": "2",
    "flightReference": "2",
    "referenceNumber": "27",
    "price": 2500
  }
],
"seats": [
  {
    "number": 4,
    "title": "A",
    "code": "RQST",
    "status": "HK",
    "passengerReference": "2",
    "flightReference": "1",
    "referenceNumber": "28"
  },
  {
    "number": 3,
    "title": "A",
    "code": "RQST",

```

```

        "status": "HK",
        "passengerReference": "2",
        "flightReference": "2",
        "referenceNumber": "31"
    }
  ],
  "insurance": {
    "number": "26771038",
    "code": "URFL",
    "status": "HK",
    "referenceNumber": "59"
  }
},
"errors": [],
"success": true
}

```

### Бронирование для перелёта туда и обратно

**Строка запроса:** `api/v1/reservation?pnrNumber=MPPU4W&lastName=VETRODUEV`

### Ответ:

```

{
  "data": {
    "tickets": [
      {
        "text": "PAX 262-
2437100848/ETU6/RUB23652/03JUN18/SVXU608AA/00777534",
        "passengerReference": "2",
        "price": 23652
      }
    ],
    "outboundFareFamily": {
      "fareFamilyId": 0,
      "code": "ECOFLEX",
      "options": [
        {
          "code": "handLuggage5kg",
          "category": "handBag",
          "order": 2,
          "showInPopup": true,
          "showInSummary": true,
          "showInUpsell": false,
          "showInFareIncluded": false,
          "showForAdult": true,
          "showForInfant": false
        },
        {
          "code": "mealsOnBoard",
          "category": "menu",
          "order": 3,
          "showInPopup": true,

```

```

    "showInSummary": false,
    "showInUpsell": false,
    "showInFareIncluded": false,
    "showForAdult": true,
    "showForInfant": false
  },
  {
    "code": "flexibilityOnChangesAndCancellations",
    "category": "ffServices",
    "order": 5,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": true,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true
  },
  {
    "code": "upTo10bonuses",
    "category": "bonus",
    "order": 4,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": true,
    "showForInfant": true
  },
  {
    "code": "oneSuitcase10kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": false,
    "showForInfant": true
  },
  {
    "code": "handLuggage5kg",
    "category": "handBag",
    "order": 2,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": false,
    "showInFareIncluded": true,
    "showForAdult": false,
    "showForInfant": true
  },
  {
    "code": "oneSuitcase23kg",
    "category": "luggage",
    "order": 1,
    "showInPopup": true,
    "showInSummary": true,
    "showInUpsell": true,

```

```

        "showInFareIncluded": true,
        "showForAdult": true,
        "showForInfant": false
    }
],
"order": 0,
"commercialFamily": "U6ECONOMY"
},
"returnFareFamily": {
    "fareFamilyId": 0,
    "code": "ECOFLEX",
    "options": [
        {
            "code": "handLuggage5kg",
            "category": "handBag",
            "order": 2,
            "showInPopup": true,
            "showInSummary": true,
            "showInUpsell": false,
            "showInFareIncluded": false,
            "showForAdult": true,
            "showForInfant": false
        },
        {
            "code": "mealsOnBoard",
            "category": "menu",
            "order": 3,
            "showInPopup": true,
            "showInSummary": false,
            "showInUpsell": false,
            "showInFareIncluded": false,
            "showForAdult": true,
            "showForInfant": false
        },
        {
            "code": "flexibilityOnChangesAndCancellations",
            "category": "ffServices",
            "order": 5,
            "showInPopup": true,
            "showInSummary": true,
            "showInUpsell": true,
            "showInFareIncluded": true,
            "showForAdult": true,
            "showForInfant": true
        },
        {
            "code": "upTo10bonuses",
            "category": "bonus",
            "order": 4,
            "showInPopup": true,
            "showInSummary": true,
            "showInUpsell": false,
            "showInFareIncluded": true,
            "showForAdult": true,
            "showForInfant": true
        },
        {

```



```

        "code": "oneSuitcase10kg",
        "category": "luggage",
        "order": 1,
        "showInPopup": true,
        "showInSummary": true,
        "showInUpsell": false,
        "showInFareIncluded": true,
        "showForAdult": false,
        "showForInfant": true
    },
    {
        "code": "handLuggage5kg",
        "category": "handBag",
        "order": 2,
        "showInPopup": true,
        "showInSummary": true,
        "showInUpsell": false,
        "showInFareIncluded": true,
        "showForAdult": false,
        "showForInfant": true
    },
    {
        "code": "oneSuitcase23kg",
        "category": "luggage",
        "order": 1,
        "showInPopup": true,
        "showInSummary": true,
        "showInUpsell": true,
        "showInFareIncluded": true,
        "showForAdult": true,
        "showForInfant": false
    }
],
"order": 0,
"commercialFamily": "U6ECONOMY"
},
"tsmData": [],
"appliedDiscounts": [],
"baseFarePrices": [
    {
        "currency": "RUB",
        "passengerReference": "2",
        "isInfantFare": false,
        "totalPrice": 23652,
        "basePrice": 18015
    }
],
"journey": {
    "outboundFlights": [
        {
            "aircraft": "Airbus A320",
            "referenceNumber": "1",
            "origin": "LED",
            "destination": "SVX",
            "departureDate": "2018-06-10T01:00:00",
            "arrivalDate": "2018-06-10T05:30:00",
            "flightNumber": "3",

```

```

        "classOfService": "K",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightDuration": "2:30",
        "stopDuration": "6:30"
    },
    {
        "aircraft": "Airbus A320",
        "referenceNumber": "2",
        "origin": "SVX",
        "destination": "DME",
        "departureDate": "2018-06-10T12:00:00",
        "arrivalDate": "2018-06-10T12:30:00",
        "flightNumber": "274",
        "classOfService": "O",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightDuration": "2:30",
        "stopDuration": "22:50"
    }
],
"outboundFareFamily": {
    "code": "ECOFLEXRU"
},
"returnFareFamily": {
    "code": "ECOFLEXRU"
},
"outboundDuration": "10:20",
"returnFlights": [
    {
        "aircraft": "Airbus A319",
        "referenceNumber": "3",
        "origin": "DME",
        "destination": "LED",
        "departureDate": "2018-06-20T11:20:00",
        "arrivalDate": "2018-06-20T12:50:00",
        "flightNumber": "79",
        "classOfService": "B",
        "operatingCarrier": "U6",
        "flightDuration": "1:30"
    }
],
"returnDuration": "1:30",
"origin": "LED",
"destination": "DME",
"departureDate": "2018-06-10T01:00:00",
"returnDate": "2018-06-20T11:20:00"
},
"number": "PNRNUMBER",
"reservationDate": "2018-06-03T13:01:00",
"passengers": [
    {
        "id": 1,
        "hasInfant": false,
        "referenceNumber": "2",
        "firstName": "VLADISLAV",
        "middleName": "IVANOVICH",
    }
]

```

```

        "surname": "VETRODUEV",
        "passengerType": 0,
        "isAdult": true,
        "birthDate": "1988-01-01T00:00:00",
        "sex": 0,
        "nationality": "RUS",
        "documentNumber": "1409345890",
        "documentExpiryDate": "2028-06-03T00:00:00",
        "email": "V.VETER@YANDEX.RU",
        "phone": "89510001133",
        "documentType": 0
    }
  ],
  "services": [],
  "seats": [],
  "insurance": {
    "number": "26771040",
    "code": "URFL",
    "status": "HK",
    "referenceNumber": "18"
  }
},
"errors": [],
"success": true
}

```

**Сервис catalogue, метод GET. Получение информации об услугах**

**Строка запроса:** /api/v1.1/catalogue

**Ответ:**

**Сервис services, метод POST. Добавление услуг**

<b>Добавление услуг:</b>	
<b>выбор места, страховка, багаж и провоз питомца для 2 пассажиров</b>	
Запрос	Ответ

<pre>{   "seats": [{     "number": 5,     "title": "A",     "passengerReference": "3",     "flightReference": "1"   }, {     "number": 5,     "title": "A",     "passengerReference": "2",     "flightReference": "2"   } ],   "insurance": "URFL",   "services": [{     "code": "ABAG",     "passengerReference": "3",     "flightReference": "1"   }, {     "code": "ABAG",     "passengerReference": "3",     "flightReference": "2"   } ]</pre>	<pre>{   "data": {     "journey": {       "outboundFlights": [{         "aircraft": "320",         "referenceNumber": "1",         "origin": "LED",         "destination": "SVX",         "departureDate": "2018-06-04T01:00:00",         "arrivalDate": "2018-06-04T05:30:00",         "flightNumber": "3",         "classOfService": "K",         "marketingCarrier": "U6",         "operatingCarrier": "U6",         "flightDuration": "2:30"       }, {         "aircraft": "320",         "referenceNumber": "2",         "origin": "SVX",         "destination": "DME",         "departureDate": "2018-06-04T06:50:00",         "arrivalDate": "2018-06-04T07:20:00",       }     ]   } }</pre>
---	---

<pre> }, {   "code": "PETC",   "text": "dog",   "passengerReference": "3",   "flightReference": "1" }, {   "code": "PETC",   "text": "dog",   "passengerReference": "3",   "flightReference": "2" }, {   "code": "ABAG",   "passengerReference": "2",   "flightReference": "1" }, {   "code": "ABAG",   "passengerReference": "2",   "flightReference": "2" } ] } </pre>	<pre> "flightNumber": "264", "classOfService": "A", "marketingCarrier": "U6", "operatingCarrier": "U6", "flightDuration": "2:30" } ], "outboundFareFamily": {   "code": "ECOFLEXRU" }, "outboundDuration": "5:00", "origin": "LED", "destination": "DME", "departureDate": "2018-06-04T01:00:00" }, "number": "PNRNUMBER", "reservationDate": "2018-06-03T19:01:00", "passengers": [{   "id": 1,   "hasInfant": false,   "referenceNumber": "3",   "firstName": "VLADISLAV", </pre>
--	---

	<pre>"middleName": "IVANOVICH", "surname": "VETRODUEV", "passengerType": 0, "isAdult": true, "birthDate": "1988-01-01T00:00:00", "sex": 0, "nationality": "RUS", "documentNumber": "1409345890", "documentExpiryDate": "2028-06-03T00:00:00", "email": "V.VETER@YANDEX.RU", "phone": "89510001133", "documentType": 0 }, { " id": 2, "hasInfant": false, "referenceNumber": "2", "firstName": "MARIYA", "middleName": "VIKTOROVNA", "surname": "MANOKHINA", "passengerType": 0, "isAdult": true, "birthDate": "2005-10-10T00:00:00",</pre>
--	--

	<pre>"sex": 1,  "nationality": "AUS",  "documentNumber": "IVU123123",  "documentExpiryDate": "2028-06-03T00:00:00",  "email": "M<u><a href="mailto:MARIAMANOHINA@MAIL.RU">MARIAMANOHINA@MAIL.RU</a></u>",  "phone": "89061230099",  "documentType": 0  }  ],  "services": [{    "code": "ABAG",    "status": "HK",    "passengerReference": "3",    "flightReference": "1",    "referenceNumber": "23"  }, {    "code": "ABAG",    "status": "HK",    "passengerReference": "3",    "flightReference": "2",    "referenceNumber": "24"  }, {</pre>
--	--

	<pre>"code": "PETC", "status": "HK", "text": "DOG 1PC 8KG 45X35X25", "passengerReference": "3", "flightReference": "1", "referenceNumber": "25" }, { "code": "PETC", "status": "HK", "text": "DOG 1PC 8KG 45X35X25", "passengerReference": "3", "flightReference": "2", "referenceNumber": "26" }, { "code": "ABAG", "status": "HK", "passengerReference": "2", "flightReference": "1", "referenceNumber": "27" }, { "code": "ABAG", "status": "HK",</pre>
--	--



	<pre>"passengerReference": "2", "flightReference": "2", "referenceNumber": "28" } ], "seats": [{   "number": 5,   "title": "A",   "code": "RQST",   "status": "HK",   "passengerReference": "3",   "flightReference": "1",   "referenceNumber": "29" }, {   "number": 5,   "title": "A",   "code": "RQST",   "status": "HK",   "passengerReference": "2",   "flightReference": "2",   "referenceNumber": "32" }</pre>
--	---

```
],  
  
"insurance": {  
  "number": "26771066",  
  "code": "URFL",  
  "status": "HK",  
  "referenceNumber": "60"  
}  
  
},  
  
"errors": [],  
  
"success": true  
  
}
```

**Сервис seatmap, метод GET. Схема самолёта**

Запрос	Ответ
--------	-------

/api/seatmap?referenceNumber=1&  
origin=KGD&destination=DME&  
departureDate=2018-06-  
22T16%3A55%3A00.000%2B03%3A00&  
flightNumber=168&classOfService=K&  
commercialFamily=U6ECONOMY

```
{  
  "data": {  
    "compartmentDetails": [{  
      "number": 1,  
      "seats": [{  
        "seat": "A",  
        "description": "W",  
        "characteristics": ["I", "V"],  
        "occupation": "O",  
        "classDesignator": "BUSINESS"  
      }, {  
        "seat": "C",  
        "description": "A",  
        "characteristics": ["I", "V"],  
        "occupation": "O",  
        "classDesignator": "BUSINESS"  
      }, {  
        "seat": "D",  
        "description": "A",  
        "characteristics": ["I", "V"],  
        "occupation": "O",  
        "classDesignator": "BUSINESS"  
      }  
    ]  
  }  
}
```

	<pre>}, {   "seat": "F",   "description": "W",   "characteristics": ["I", "V"],   "occupation": "O",   "classDesignator": "BUSINESS" } ] }, {   "number": 2,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": ["I"],     "occupation": "O",     "classDesignator": "BUSINESS"   }, {     "seat": "C",     "description": "A",     "characteristics": ["I"],     "occupation": "O",     "classDesignator": "BUSINESS"</pre>
--	---

	<pre>}, {   "seat": "D",   "description": "A",   "characteristics": ["I"],   "occupation": "O",   "classDesignator": "BUSINESS" }, {   "seat": "F",   "description": "W",   "characteristics": ["I"],   "occupation": "O",   "classDesignator": "BUSINESS" } ] }, {   "number": 3,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": ["CH", "L", "V"],     "passengerRelatedPrice": [{       "name": "VLADISLAV",</pre>
--	---

	<pre> "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 1000.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["CH", "I", "L", "V"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 1000.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { </pre>
--	--

	<pre> "seat": "C",  "description": "A",  "characteristics": ["CH", "L", "V"],  "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",    "seatPrice": 1000.0,   "hasInfant": false } ],  "classDesignator": "ECONOMY" }, {  "seat": "D",  "description": "A",  "characteristics": ["CH", "L", "V"],  "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",    "seatPrice": 1000.0, </pre>
--	--

	<pre> "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["CH", "I", "L", "V"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 1000.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "F", "description": "W", "characteristics": ["CH", "L", "V"], </pre>
--	---



	<pre> "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 1000.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, { "number": 4, "seats": [{   "seat": "A",   "description": "W",   "characteristics": ["CH", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2", </pre>
--	---

	<pre>"seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["CH", "I", "V"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 1000.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A",</pre>
--	---

	<pre> "characteristics": ["CH", "V"], "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, {   "seat": "D",   "description": "A",   "characteristics": ["CH", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } </pre>
--	--

	<pre>],   "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["CH", "I", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",</pre>
--	---

	<pre> "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, { "number": 5, "seats": [{ "seat": "A", "description": "W", "characteristics": ["CH", "V"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false </pre>
--	--

	<pre>    }   ],   "classDesignator": "ECONOMY" }, {   "seat": "B",   "description": "9",   "characteristics": ["CH", "I", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH", "V"],   "passengerRelatedPrice": [{</pre>
--	--

	<pre>"name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["CH", "V"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY"</pre>
--	--

	<pre>}, {   "seat": "E",   "description": "9",   "characteristics": ["CH", "I", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",</pre>
--	---



	<pre> "seatPrice": 500.0,  "hasInfant": false  }  ],  "classDesignator": "ECONOMY"  }  ]  }, {  "number": 6,  "seats": [{  "seat": "A",  "description": "W",  "characteristics": ["CH"],  "passengerRelatedPrice": [{  "name": "VLADISLAV",  "type": "ADT",  "uniqueCustomerIdentifier":  "2",  "seatPrice": 500.0,  "hasInfant": false  }  ],  "classDesignator": "ECONOMY" </pre>
--	---

	<pre>}, {   "seat": "B",   "description": "9",   "characteristics": ["CH", "I"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",</pre>
--	---

	<pre>"seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["CH", "I"],</pre>
--	--

	<pre>"passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 1000.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],</pre>
--	---

	<pre>"classDesignator": "ECONOMY" } ] }, { "number": 7, "seats": [{ "seat": "A", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["CH", "I"],</pre>
--	---

	<pre>"passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 1000.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],</pre>
--	---

	<pre>"classDesignator": "ECONOMY" }, {   "seat": "D",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["CH", "I"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",</pre>
--	--

	<pre>"2",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" } ]</pre>
--	---



```
}, {  
  
  "number": 8,  
  
  "seats": [{  
  
    "seat": "A",  
  
    "description": "W",  
  
    "characteristics": ["CH"],  
  
    "passengerRelatedPrice": [{  
  
      "name": "VLADISLAV",  
  
      "type": "ADT",  
  
      "uniqueCustomerIdentifier":  
"2",  
  
      "seatPrice": 500.0,  
  
      "hasInfant": false  
  
    }  
  
  ],  
  
  "classDesignator": "ECONOMY"  
  
}, {  
  
  "seat": "B",  
  
  "description": "9",  
  
  "characteristics": ["CH", "I"],  
  
  "passengerRelatedPrice": [{  
  
    "name": "VLADISLAV",  
  
    "type": "ADT",
```

	<pre>"2",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "D",</pre>
--	---

	<pre>"description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["CH", "I"], "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 1000.0,   "hasInfant": false</pre>
--	--

	<pre>    }   ],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   }   ],   "classDesignator": "ECONOMY" } ] }, {   "number": 9,   "seats": [{     "seat": "A",</pre>
--	--

	<pre>"description": "W", "characteristics": ["V"], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["V"], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["V"], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["V"], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["V"],</pre>
--	--

	<pre>"classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["V"],   "classDesignator": "ECONOMY" } ] }, {   "number": 10,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": ["CH", "E", "L", "V"],     "passengerRelatedPrice": [{       "name": "VLADISLAV",       "type": "ADT",       "uniqueCustomerIdentifier": "2",       "seatPrice": 1000.0,       "hasInfant": false     }   ] }</pre>
--	---

	<pre>],   "classDesignator": "ECONOMY" }, {   "seat": "B",   "description": "9",   "characteristics": ["CH", "E", "L", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 1000.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH", "E", "L", "V"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",</pre>
--	--

	<pre>"type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 1000.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["CH", "E", "L", "V"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 1000.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, {</pre>
--	--



	<pre> "seat": "E",  "description": "9",  "characteristics": ["CH", "E", "L", "V"],  "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",    "seatPrice": 1000.0,   "hasInfant": false } ],  "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH", "E", "L", "V"],  "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",    "seatPrice": 1000.0, </pre>
--	---

	<pre>"hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, { "number": 11, "seats": [{ "seat": "A", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, {</pre>
--	---

	<pre>"seat": "B", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{</pre>
--	--

	<pre>"name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "F", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2",</pre>
--	---

	<pre>"seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, { "number": 12, "seats": [{ "seat": "A", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY"</pre>
--	--

	<pre>}, {   "seat": "B",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "D",   "description": "A",   "characteristics": ["CH"],</pre>
--	---

	<pre>"passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",</pre>
--	--

	<pre>"2",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" } ] }, {   "number": 13,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": ["CH"],     "passengerRelatedPrice": [{       "name": "VLADISLAV",       "type": "ADT",       "uniqueCustomerIdentifier": "2",       "seatPrice": 500.0,       "hasInfant": false     }   ] }</pre>
--	--



	<pre>],    "classDesignator": "ECONOMY" }, {    "seat": "B",    "description": "9",    "characteristics": ["I"],    "classDesignator": "ECONOMY" }, {    "seat": "C",    "description": "A",    "characteristics": ["CH"],    "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],    "classDesignator": "ECONOMY" }, {    "seat": "D",</pre>
--	---

	<pre>"description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",</pre>
--	--

	<pre> "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, { "number": 14, "seats": [{ "seat": "A", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false </pre>
--	--

	<pre>    }   ],   "classDesignator": "ECONOMY" }, {   "seat": "B",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   }   ],   "classDesignator": "ECONOMY" }, {</pre>
--	---

	<pre>"seat": "D", "description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "F", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{</pre>
--	--

	<pre> "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, { "number": 15, "seats": [{ "seat": "A", "description": "W", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, </pre>
--	--

	<pre>"hasInfant": false } ], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": ["CH"], "passengerRelatedPrice": [{ "name": "VLADISLAV", "type": "ADT", "uniqueCustomerIdentifier": "2", "seatPrice": 500.0, "hasInfant": false } ], "classDesignator": "ECONOMY"</pre>
--	---

	<pre>}, {   "seat": "D",   "description": "A",   "characteristics": ["CH"],   "passengerRelatedPrice": [{     "name": "VLADISLAV",     "type": "ADT",     "uniqueCustomerIdentifier": "2",     "seatPrice": 500.0,     "hasInfant": false   } ],   "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": ["CH"],</pre>
--	---



	<pre>"passengerRelatedPrice": [{   "name": "VLADISLAV",   "type": "ADT",   "uniqueCustomerIdentifier": "2",   "seatPrice": 500.0,   "hasInfant": false } ], "classDesignator": "ECONOMY" } ] }, {   "number": 16,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": [],     "classDesignator": "ECONOMY"   }, {     "seat": "B",     "description": "9",     "characteristics": ["I"],</pre>
--	--

	<pre>"classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": [],   "classDesignator": "ECONOMY" }, {   "seat": "D",   "description": "A",   "characteristics": [],   "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": [],   "classDesignator": "ECONOMY" }</pre>
--	--

	<pre>] }, {   "number": 17,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": [],     "classDesignator": "ECONOMY"   }, {     "seat": "B",     "description": "9",     "characteristics": ["I"],     "classDesignator": "ECONOMY"   }, {     "seat": "C",     "description": "A",     "characteristics": [],     "classDesignator": "ECONOMY"   }, {     "seat": "D",     "description": "A",     "characteristics": [],</pre>
--	--

	<pre>"classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": [],   "classDesignator": "ECONOMY" } ] }, {   "number": 18,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": [],     "classDesignator": "ECONOMY"   }, {     "seat": "B",</pre>
--	--

	<pre>"description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "F", "description": "W", "characteristics": [],</pre>
--	---

	<pre>"classDesignator": "ECONOMY" } ] }, { "number": 19, "seats": [{ "seat": "A", "description": "W", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "D",</pre>
--	--

	<pre>"description": "A", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "F", "description": "W", "characteristics": [], "classDesignator": "ECONOMY" } ] }, { "number": 20, "seats": [{ "seat": "A", "description": "W", "characteristics": [], "classDesignator": "ECONOMY"</pre>
--	---

	<pre>}, {   "seat": "B",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "C",   "description": "A",   "characteristics": [],   "classDesignator": "ECONOMY" }, {   "seat": "D",   "description": "A",   "characteristics": [],   "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",</pre>
--	--



	<pre>"description": "W", "characteristics": [], "classDesignator": "ECONOMY" } ] }, { "number": 21, "seats": [{ "seat": "A", "description": "W", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "B", "description": "9", "characteristics": ["I"], "classDesignator": "ECONOMY" }, { "seat": "C", "description": "A", "characteristics": [], "classDesignator": "ECONOMY"</pre>
--	---

	<pre>}, {   "seat": "D",   "description": "A",   "characteristics": [],   "classDesignator": "ECONOMY" }, {   "seat": "E",   "description": "9",   "characteristics": ["I"],   "classDesignator": "ECONOMY" }, {   "seat": "F",   "description": "W",   "characteristics": [],   "classDesignator": "ECONOMY" } ] }, {   "number": 22,   "seats": [{     "seat": "A",     "description": "W",</pre>
--	---

	<pre>"characteristics": [],  "classDesignator": "ECONOMY"  }, {  "seat": "B",  "description": "9",  "characteristics": ["I"],  "classDesignator": "ECONOMY"  }, {  "seat": "C",  "description": "A",  "characteristics": [],  "classDesignator": "ECONOMY"  }, {  "seat": "D",  "description": "A",  "characteristics": [],  "classDesignator": "ECONOMY"  }, {  "seat": "E",  "description": "9",  "characteristics": ["I"],  "classDesignator": "ECONOMY"</pre>
--	---

	<pre>}, {   "seat": "F",   "description": "W",   "characteristics": [],   "classDesignator": "ECONOMY" } ] }, {   "number": 23,   "seats": [{     "seat": "A",     "description": "W",     "characteristics": [],     "classDesignator": "ECONOMY"   }, {     "seat": "B",     "description": "9",     "characteristics": [],     "classDesignator": "ECONOMY"   }, {     "seat": "C",     "description": "A",</pre>
--	--

	<pre>"characteristics": ["C", "V"], "classDesignator": "ECONOMY" }, { "seat": "D", "description": "A", "characteristics": ["C", "V"], "classDesignator": "ECONOMY" }, { "seat": "E", "description": "9", "characteristics": [], "classDesignator": "ECONOMY" }, { "seat": "F", "description": "W", "characteristics": [], "classDesignator": "ECONOMY" } ] } ] },</pre>
--	---

	<pre> "errors": [], "success": true } </pre>
--	--

## Сервис user, метод POST. Авторизация в CRM

**Запрос:** api/v1.0/user

**Тело запроса:** {"login":"1000332032","password":"tes756"}

**Ответ:**

```

{
  "data": {
    "userId": "1-1PIWGTP",
    "firstName": "Владислав",
    "firstNameLat": "Vladislav",
    "middleName": "Иванович",
    "middleNameLat": "Ivanovich",
    "lastName": "Ветродуев",
    "lastNameLat": "Vetroduev",
    "birthDate": "1988-01-01T00:00:00",
    "email": "v.veter@yandex.ru",
    "cellPhone": "89510001133",
    "homePhone": "",
    "address": {
      "city": "Чита",
      "country": "Russia",
      "region": "Chitinskaya obl.",
      "street": " Profsobznaya, 26",
      "zip": ""
    },
  },
  "documents": [{
    "number": "1409345890",
    "title": "Общегражданский Паспорт",
    "lastUpdateDate": "2018-03-29T15:36:47",
    "documentType": 0
  }
],
  "logins": [{
    "loginName": "v.veter@yandex.ru",
    "loginType": "Электронная почта"
  }, {
    "loginName": "89510001133",
    "loginType": "Телефон"
  }
],
  "loyalties": [{

```

```

"memberId": "1-1PIWGTV",
"level": "Синий",
"program": "Крылья",
"programArt": "WingsProgramArticul",
"joinDate": "0001-01-01T00:00:00",
"currentPoints": 0,
"levelPoints": 0,
"levelBonus": 0,
"levelStartDate": "18.09.2017",
"lastFlightDate": "",
"cards": [{
  "active": true,
  "ffn": "1000332032",
  "name": "TESLA1 TESLA5",
  "cardLevel": "Синий",
  "cardType": "Обычный",
  "issueDate": "2017-09-18T18:00:00",
  "validFromDate": "2017-09-18T18:00:00",
  "validToDate": "2117-09-19T08:03:15"
}
],
"activeCard": {
  "active": true,
  "ffn": "1000332032",
  "name": "TESLA1 TESLA5",
  "cardLevel": "Синий",
  "cardType": "Обычный",
  "issueDate": "2017-09-18T18:00:00",
  "validFromDate": "2017-09-18T18:00:00",
  "validToDate": "2117-09-19T08:03:15"
}
}
],
"sex": 0,
"wingsLoyalty": {
  "memberId": "1-1PIWGTV",
  "level": "Синий",
  "program": "Крылья",
  "programArt": "WingsProgramArticul",
  "joinDate": "0001-01-01T00:00:00",
  "currentPoints": 0,
  "levelPoints": 0,
  "levelBonus": 0,
  "levelStartDate": "18.09.2017",
  "lastFlightDate": "",
  "cards": [{
    "active": true,
    "ffn": "1000332032",
    "name": "TESLA1 TESLA5",
    "cardLevel": "Синий",
    "cardType": "Обычный",
    "issueDate": "2017-09-18T18:00:00",
    "validFromDate": "2017-09-18T18:00:00",
    "validToDate": "2117-09-19T08:03:15"
  }
}
],
"activeCard": {

```

```

    "active": true,
    "ffn": "1000332032",
    "name": "TESLA1 TESLA5",
    "cardLevel": "Синий",
    "cardType": "Обычный",
    "issueDate": "2017-09-18T18:00:00",
    "validFromDate": "2017-09-18T18:00:00",
    "validToDate": "2117-09-19T08:03:15"
  }
}
},
"errors": [],
"success": true
}

```

## Сервис households, метод GET. Получение связанных пассажиров

**Запрос:** api/v1.0/households

**Ответ:**

```

{
  "data": [{
    "userId": "1-1PJVCU",
    "firstName": "Сергей",
    "firstNameLat": "Sergey",
    "middleName": "",
    "middleNameLat": "",
    "lastName": "Орлик",
    "lastNameLat": "Орлик",
    "birthDate": "0001-01-01T00:00:00"
  }, {
    "userId": "1-1PIWGTP",
    "firstName": "Владислав",
    "firstNameLat": "Vladislav",
    "middleName": "Иванович",
    "middleNameLat": "Ivanovich",
    "lastName": "Ветродуев",
    "lastNameLat": "Ветродуев",
    "birthDate": "0001-01-01T00:00:00"
  }, {
    "userId": "1-1PIR4T7",
    "firstName": "Мария",
    "firstNameLat": "Maria",
    "middleName": "Викторовна",
    "middleNameLat": "Viktorovna",
    "lastName": "Манохина",
    "lastNameLat": "Манохина",
    "birthDate": "0001-01-01T00:00:00"
  }, {
    "userId": "1-1025FHL",
    "firstName": "Младенчик",
    "firstNameLat": "Mladenichik",
    "middleName": "Николаевна",
    "middleNameLat": "Nikolaevna",

```



```
    "lastName": "Манохина",
    "lastNameLat": "Манохина",
    "birthDate": "0001-01-01T00:00:00"
  }
],
"errors": [],
"success": true
}
```

## Сервис promotions, метод POST

### Промокод на перелёт

### Запрос /api/v1.1/promotions

```
{
  "number": "9jKBaQ56",
  "origin": "LED",
  "destination": "SVX",
  "departureDate": "2018-06-20",
  "returnDate": null,
  "passengersCount": 1
}
```

### Ответ:

```
{
  "data": {
    "number": "9jKBaQ56",
    "amount": 6.0,
    "promotionStatus": 0,
    "promotionType": 0,
    "description": "Скидка на несколько тарифов.",
    "isUnique": false,
    "exclude": true,
    "campaignId": "1-1PNGECV",
    "directions": [],
    "services": [],
    "families": [{
      "code": "ECOLIGHT"
    }, {
      "code": "ECOFLEX"
    }, {
      "code": "BUSFLEX"
    }
  ],
  "excludedFlights": [{
    "flightNumber": "6006",
    "departureAirport": "AER",
    "arrivalAirport": "LED"
  }
}
```

```
    ],
    "hasConflict": false,
    "isValid": true
  },
  "errors": [],
  "success": true
}
```

### Промокод на дополнительные услуги

#### Запрос /api/v1.1/promotions

```
{
  "number": "oGVy099w",
  "origin": "LED",
  "destination": "SVX",
  "departureDate": "2018-06-20",
  "returnDate": null,
  "passengersCount": 1
}
```

#### Ответ:

```
{
  "data": {
    "number": "oGVy099w",
    "amount": 12.0,
    "promotionStatus": 0,
    "promotionType": 1,
    "description": "Скидка на все дополнительные услуги",
    "startDate": "2018-01-01T00:00:00",
    "endDate": "2019-01-01T00:00:00",
    "isUnique": false,
    "exclude": true,
    "campaignId": "1-1PSAKF4",
    "directions": [],
    "services": [{
      "code": "PETC"
    }, {
      "code": "BBAG"
    }, {
      "code": "CBAG"
    }, {
      "code": "HBAG"
    }, {
      "code": "SEAT"
    }, {
      "code": "VGML"
    }, {
      "code": "CHML"
    }, {

```

```

        "code": "MOML"
    }, {
        "code": "SNML"
    }, {
        "code": "HTML"
    }, {
        "code": "SFML"
    }, {
        "code": "ABAG"
    }
],
"families": [],
"excludedFlights": [{
    "flightNumber": "6006",
    "departureAirport": "AER",
    "arrivalAirport": "LED"
}
],
"hasConflict": false,
"isValid": true
},
"errors": [],
"success": true
}

```

Пример запроса схемы мест в самолете без предварительно созданного бронирования для 1 взрослого с младенцем: POST api/SeatmapStandalone

### Запрос:

```

{
  "passengers": [
    {
      "id": 2,
      "firstName": "Ivan",
      "surname": "Petrov",
      "passengerType": "0",
      "sex": "0",
      "birthDate": "1989-10-25",
      "hasInfant": true
    }
  ],
  "origin": "LED",
  "destination": "DME",
  "departureDate": "2020-06-22",
  "flightNumber": "60",
  "classOfService": "X",
  "commercialFamily": "U6ECONOMY"
}

```

### Ответ:

```

{

```

```

"data": {
  "compartmentDetails": [{
    "number": 3,
    "seats": [{
      "seat": "A",
      "number": 3,
      "description": "W",
      "characteristics": ["CH", "L", "W"],
      "occupation": "F",
      "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ]
  }, {
    "classDesignator": "ECONOMY"
  }, {
    "seat": "B",
    "number": 3,
    "description": "9",
    "characteristics": ["CH", "I", "L"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ], {
    "classDesignator": "ECONOMY"
  }, {
    "seat": "C",
    "description": "A",
    "characteristics": ["A", "CH", "L"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ], {
    "classDesignator": "ECONOMY"
  }, {
    "seat": "D",
    "description": "A",
    "characteristics": ["A", "CH", "L"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",

```

```

        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}, {
    "seat": "E",
    "number": 3,
    "description": "9",
    "characteristics": ["CH", "I", "L"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}, {
    "seat": "F",
    "number": 3,
    "description": "W",
    "characteristics": ["CH", "L", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 4,
    "seats": [{
        "seat": "A",
        "number": 4,
        "description": "W",
        "characteristics": ["C", "CH", "W"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 4,
  "description": "9",
  "characteristics": ["C", "CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["A", "C", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["A", "C", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 4,
  "description": "9",
  "characteristics": ["C", "CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",

```

```

        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 4,
  "description": "W",
  "characteristics": ["C", "CH", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 5,
  "seats": [{
    "seat": "A",
    "number": 5,
    "description": "W",
    "characteristics": ["CH", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "B",
    "number": 5,
    "description": "9",
    "characteristics": ["CH", "I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ]
}
]
}

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 5,
  "description": "9",
  "characteristics": ["CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 5,
  "description": "W",
  "characteristics": ["CH", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",

```



```

        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 6,
    "seats": [{
        "seat": "A",
        "number": 6,
        "description": "W",
        "characteristics": ["CH", "W"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
},
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 6,
    "description": "9",
    "characteristics": ["CH", "I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": ["A", "CH"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
}
}

```

```

    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "D",
    "description": "A",
    "characteristics": ["A", "CH"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 6,
  "description": "9",
  "characteristics": ["CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 6,
  "description": "W",
  "characteristics": ["CH", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
]
}, {
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 7,
  "seats": [{
    "seat": "A",
    "number": 7,
    "description": "W",

```

```

"characteristics": ["CH", "W"],
"occupation": "F",
"passengerRelatedPrice": [{
  "name": "IVAN",
  "type": "ADT",
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 299.0,
  "isSeatUnavailable": true,
  "hasInfant": true
}
],
"classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 7,
  "description": "9",
  "characteristics": ["CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],

```

```

    "classDesignator": "ECONOMY"
  }, {
    "seat": "E",
    "number": 7,
    "description": "9",
    "characteristics": ["CH", "I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 7,
  "description": "W",
  "characteristics": ["CH", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 8,
  "seats": [{
    "seat": "A",
    "number": 8,
    "description": "W",
    "characteristics": ["CH", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 8,
  "description": "9",

```

```

"characteristics": ["CH", "I"],
"occupation": "F",
"passengerRelatedPrice": [{
  "name": "IVAN",
  "type": "ADT",
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 299.0,
  "isSeatUnavailable": false,
  "hasInfant": true
}
],
"classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 8,
  "description": "9",
  "characteristics": ["CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],

```

```

    "classDesignator": "ECONOMY"
  }, {
    "seat": "F",
    "number": 8,
    "description": "W",
    "characteristics": ["CH", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 9,
  "seats": [{
    "seat": "A",
    "number": 9,
    "description": "W",
    "characteristics": ["CH", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 9,
  "description": "9",
  "characteristics": ["CH", "I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["A", "CH"],

```



```

        "classDesignator": "ECONOMY"
    }
]
}, {
    "number": 10,
    "seats": [{
        "seat": "A",
        "number": 10,
        "description": "W",
        "characteristics": ["1D", "W"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
}, {
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 10,
    "description": "9",
    "characteristics": ["1D", "I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": ["1D", "A"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": ["1D", "A"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
}
}

```



```

    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "E",
    "number": 10,
    "description": "9",
    "characteristics": ["1D", "I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 10,
  "description": "W",
  "characteristics": ["1D", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}
]
}, {
  "number": 11,
  "seats": [{
    "seat": "A",
    "number": 11,
    "description": "W",
    "characteristics": ["1A", "1D", "CH", "E", "IE", "L",
"V", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 11,
  "description": "9",
  "characteristics": ["1A", "1D", "CH", "E", "IE", "L",
"V"],

```

```

"occupation": "F",
"passengerRelatedPrice": [{
  "name": "IVAN",
  "type": "ADT",
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 999.0,
  "isSeatUnavailable": true,
  "hasInfant": true
}
],
"classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["1A", "1D", "A", "CH", "E", "IE",
"occupation": "F",
"passengerRelatedPrice": [{
  "name": "IVAN",
  "type": "ADT",
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 999.0,
  "isSeatUnavailable": true,
  "hasInfant": true
}
],
"classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["1A", "1D", "A", "CH", "E", "IE",
"occupation": "F",
"passengerRelatedPrice": [{
  "name": "IVAN",
  "type": "ADT",
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 999.0,
  "isSeatUnavailable": true,
  "hasInfant": true
}
],
"classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 11,
  "description": "9",
  "characteristics": ["1A", "1D", "CH", "E", "IE", "L",
"occupation": "F",
"passengerRelatedPrice": [{
  "name": "IVAN",
  "type": "ADT",
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 999.0,
  "isSeatUnavailable": true,
  "hasInfant": true

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 11,
  "description": "W",
  "characteristics": ["1A", "1D", "CH", "E", "IE", "L",
"V", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 999.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 12,
  "seats": [{
    "seat": "A",
    "number": 12,
    "description": "W",
    "characteristics": ["1A", "CH", "E", "IE", "L", "V",
"W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "B",
    "number": 12,
    "description": "9",
    "characteristics": ["1A", "CH", "E", "IE", "L", "V"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }
]

```

```

    }, {
      "seat": "C",
      "description": "A",
      "characteristics": ["1A", "A", "CH", "E", "IE", "L",
"V"],
      "occupation": "F",
      "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "D",
    "description": "A",
    "characteristics": ["1A", "A", "CH", "E", "IE", "L",
"V"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 999.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 12,
  "description": "9",
  "characteristics": ["1A", "CH", "E", "IE", "L", "V"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 999.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 12,
  "description": "W",
  "characteristics": ["1A", "CH", "E", "IE", "L", "V",
"W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",

```

```

        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 999.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 13,
    "seats": [{
        "seat": "A",
        "number": 13,
        "description": "W",
        "characteristics": ["CH", "W"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 13,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ]
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": ["A", "CH"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ]
    ],
    "classDesignator": "ECONOMY"
}

```

```

    }, {
      "seat": "D",
      "description": "A",
      "characteristics": ["A", "CH"],
      "occupation": "F",
      "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
      }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "E",
    "number": 13,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 13,
  "description": "W",
  "characteristics": ["CH", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}
]
}, {
  "number": 14,
  "seats": [{
    "seat": "A",
    "number": 14,
    "description": "W",
    "characteristics": ["CH", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",

```

```

        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 14,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": ["A", "CH"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": ["A", "CH"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}, {
    "seat": "E",
    "number": 14,
    "description": "9",
    "characteristics": ["I"],

```

```

        "occupation": "F",
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "F",
    "number": 14,
    "description": "W",
    "characteristics": ["CH", "W"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "name": "IVAN",
        "type": "ADT",
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 299.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 15,
    "seats": [{
        "seat": "A",
        "number": 15,
        "description": "W",
        "characteristics": ["CH", "W"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "name": "IVAN",
            "type": "ADT",
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 299.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 15,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
}

```



```

    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "C",
    "description": "A",
    "characteristics": ["A", "CH"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "name": "IVAN",
      "type": "ADT",
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 299.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["A", "CH"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 15,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "number": 15,
  "description": "W",
  "characteristics": ["CH", "W"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "name": "IVAN",
    "type": "ADT",
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 299.0,

```

```

        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 16,
    "seats": [{
        "seat": "A",
        "description": "W",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 16,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ]
}

```

```

    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "E",
    "number": 16,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}
]
}, {
  "number": 17,
  "seats": [{
    "seat": "A",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 17,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],

```

```

    "classDesignator": "ECONOMY"
  }, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 17,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}
]
}, {
  "number": 18,
  "seats": [{
    "seat": "A",

```

```

        "description": "W",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 18,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "E",
    "number": 18,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,

```

```

        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 19,
    "seats": [{
        "seat": "A",
        "description": "W",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
        ],
        "classDesignator": "ECONOMY"
    }, {
        "seat": "B",
        "number": 19,
        "description": "9",
        "characteristics": ["I"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
        ],
        "classDesignator": "ECONOMY"
    }, {
        "seat": "C",
        "description": "A",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
        ]
    }
}
]
}

```

```

    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 19,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}
]
}, {
  "number": 20,
  "seats": [{
    "seat": "A",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}, {
  "classDesignator": "ECONOMY"
}, {

```

```

        "seat": "B",
        "number": 20,
        "description": "9",
        "characteristics": ["I"],
        "occupation": "F",
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": false,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "E",
    "number": 20,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",

```



```

        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 21,
    "seats": [{
        "seat": "A",
        "description": "W",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 21,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 21,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }]
}, {
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }]
}, {
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 22,
  "seats": [{
    "seat": "A",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }]
  }],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 22,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }]
}
}

```

```

    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 22,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}
]
}, {
  "number": 23,
  "seats": [{

```

```

    "seat": "A",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 23,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 23,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 24,
  "seats": [{
    "seat": "A",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "B",
    "number": 24,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
}
]
}

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 24,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}
],
  "number": 25,
  "seats": [{
    "seat": "A",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
  ]
},
  "classDesignator": "ECONOMY"

```

```

}, {
  "seat": "B",
  "number": 25,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 25,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{

```

```

        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
]
}, {
    "number": 26,
    "seats": [{
        "seat": "A",
        "description": "W",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 26,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "C",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,

```



```

        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}, {
    "seat": "E",
    "number": 26,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}, {
    "seat": "F",
    "description": "W",
    "characteristics": [],
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
],
"classDesignator": "ECONOMY"
}
]
}, {
    "number": 27,
    "seats": [{
        "seat": "A",
        "description": "W",
        "characteristics": [],
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ]
}, {
    "classDesignator": "ECONOMY"
}, {
    "seat": "B",
    "number": 27,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "F",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }

```

```

    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 27,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}
],
}, {
  "number": 28,

```

```

"seats": [{
  "seat": "A",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 28,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
  "classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 28,
  "description": "9",
  "characteristics": ["I"],
  "occupation": "F",
  "passengerRelatedPrice": [{

```

```

        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
  ],
  "classDesignator": "ECONOMY"
}, {
  "seat": "F",
  "description": "W",
  "characteristics": [],
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
  ],
  "classDesignator": "ECONOMY"
}
]
}, {
  "number": 29,
  "seats": [{
    "seat": "A",
    "number": 29,
    "description": "W",
    "characteristics": ["W"],
    "occupation": "O",
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": true,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "B",
    "number": 29,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "O",
    "passengerRelatedPrice": [{
      "uniqueCustomerIdentifier": "2",
      "seatPrice": 0.0,
      "isSeatUnavailable": false,
      "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
  }, {
    "seat": "C",
    "description": "A",
    "characteristics": ["A"],
    "occupation": "O",
    "passengerRelatedPrice": [{

```

```

        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
}, {
    "classDesignator": "ECONOMY"
}, {
    "seat": "D",
    "description": "A",
    "characteristics": ["A"],
    "occupation": "O",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "E",
    "number": 29,
    "description": "9",
    "characteristics": ["I"],
    "occupation": "O",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}, {
    "seat": "F",
    "number": 29,
    "description": "W",
    "characteristics": ["W"],
    "occupation": "O",
    "passengerRelatedPrice": [{
        "uniqueCustomerIdentifier": "2",
        "seatPrice": 0.0,
        "isSeatUnavailable": true,
        "hasInfant": true
    }
    ],
    "classDesignator": "ECONOMY"
}
],
}, {
    "number": 30,
    "seats": [{
        "seat": "A",
        "number": 30,
        "description": "W",
        "characteristics": ["1D", "7A", "W"],

```

```

"occupation": "O",
"passengerRelatedPrice": [{
  "uniqueCustomerIdentifier": "2",
  "seatPrice": 0.0,
  "isSeatUnavailable": true,
  "hasInfant": true
}
],
"classDesignator": "ECONOMY"
}, {
  "seat": "B",
  "number": 30,
  "description": "9",
  "characteristics": ["1D", "7A", "I"],
  "occupation": "O",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": false,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "C",
  "description": "A",
  "characteristics": ["1D", "7A", "A"],
  "occupation": "O",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "D",
  "description": "A",
  "characteristics": ["1D", "7A", "A"],
  "occupation": "O",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",
    "seatPrice": 0.0,
    "isSeatUnavailable": true,
    "hasInfant": true
  }
],
"classDesignator": "ECONOMY"
}, {
  "seat": "E",
  "number": 30,
  "description": "9",
  "characteristics": ["1D", "7A", "I"],
  "occupation": "O",
  "passengerRelatedPrice": [{
    "uniqueCustomerIdentifier": "2",

```

```

        "seatPrice": 0.0,
        "isSeatUnavailable": false,
        "hasInfant": true
    }, {
        "classDesignator": "ECONOMY"
    }, {
        "seat": "F",
        "number": 30,
        "description": "W",
        "characteristics": ["1D", "7A", "W"],
        "occupation": "O",
        "passengerRelatedPrice": [{
            "uniqueCustomerIdentifier": "2",
            "seatPrice": 0.0,
            "isSeatUnavailable": true,
            "hasInfant": true
        }
    ],
        "classDesignator": "ECONOMY"
    }
]
}
],
"isEmpty": false,
"passengersTotalData": [{
    "passengerReference": "2",
    "passengerPriceLevels": [{
        "price": 299.0,
        "priceLevel": 1
    }, {
        "price": 999.0,
        "priceLevel": 2
    }
]
}
]
},
"errors": [],
"success": true
}

```

**Пример запроса каталога услуг без предварительно созданного бронирования для 1 взрослого с ребенком и младенцем: POST api/CatalogueStandalone**

**Запрос:**

```

{
  "flights": {
    "outboundFlights": [{
      "origin": "LED",
      "destination": "DME",
      "departureDate": "2020-07-03T16:05:00",
      "arrivalDate": "2020-07-03T17:30:00",
    }
  ]
}

```

```

        "flightNumber": "60",
        "classOfService": "P",
        "marketingCarrier": "U6",
        "operatingCarrier": "U6",
        "flightIndex": 1
    }
]
},
"fares": {
    "outboundFareFamily": {
        "code": "ECOPREMURU",
        "category": "ECOPREM"
    },
    "fares": [{
        "passengerType": 0,
        "flightIndex": 1,
        "fareBasis": "PFLOW",
        "isBreakpoint": true
    }, {
        "passengerType": 5,
        "flightIndex": 1,
        "fareBasis": "PFLOW",
        "isBreakpoint": true
    }, {
        "passengerType": 4,
        "flightIndex": 1,
        "fareBasis": "PFLOW",
        "isBreakpoint": true
    }
]
},
"passengers": [{
    "id": 3,
    "firstName": "Sofia",
    "middleName": "Evgenievna",
    "surname": "Bodrova",
    "passengerType": 0,
    "birthDate": "1995-07-20",
    "sex": "Female",
    "hasInfant": true
}, {
    "id": 2,
    "firstName": "Rebenok",
    "surname": "Bodrova",
    "passengerType": 5,
    "birthDate": "2015-07-20",
    "sex": "Male"
}, {
    "id": 4,
    "firstName": "Mladenets",
    "surname": "Bodrova",
    "passengerType": 4,
    "birthDate": "2019-07-20",
    "sex": "Male"
}
]
}

```



**ОТВЕТ:**

```
{
  "data": {
    "services": [{
      "code": "HBAG",
      "flightReferences": [{
        "referenceNumber": "1",
        "passengerPrices": [{
          "referenceNumber": "3",
          "amount": 2000.0
        }, {
          "referenceNumber": "2",
          "amount": 2000.0
        }, {
          "referenceNumber": "4",
          "amount": 2000.0
        }
      ]
    }
  ]
}, {
  "code": "BGML",
  "flightReferences": [{
    "referenceNumber": "1",
    "passengerPrices": [{
      "referenceNumber": "3",
      "amount": 500.0
    }, {
      "referenceNumber": "2",
      "amount": 500.0
    }, {
      "referenceNumber": "4",
      "amount": 500.0
    }
  ]
}
]
}, {
  "code": "ABAG",
  "flightReferences": [{
    "referenceNumber": "1",
    "passengerPrices": [{
      "referenceNumber": "3",
      "amount": 2000.0
    }, {
      "referenceNumber": "2",
      "amount": 2000.0
    }, {
      "referenceNumber": "4",
      "amount": 2000.0
    }
  ]
}
]
}, {
```

```

"code": "SFML",
"flightReferences": [{
  "referenceNumber": "1",
  "passengerPrices": [{
    "referenceNumber": "3",
    "amount": 600.0
  }, {
    "referenceNumber": "2",
    "amount": 600.0
  }, {
    "referenceNumber": "4",
    "amount": 600.0
  }
]
}
], {
"code": "CHML",
"flightReferences": [{
  "referenceNumber": "1",
  "passengerPrices": [{
    "referenceNumber": "3",
    "amount": 600.0
  }, {
    "referenceNumber": "2",
    "amount": 600.0
  }, {
    "referenceNumber": "4",
    "amount": 600.0
  }
]
}
], {
"code": "HTML",
"flightReferences": [{
  "referenceNumber": "1",
  "passengerPrices": [{
    "referenceNumber": "3",
    "amount": 600.0
  }, {
    "referenceNumber": "2",
    "amount": 600.0
  }, {
    "referenceNumber": "4",
    "amount": 600.0
  }
]
}
], {
"code": "CAML",
"flightReferences": [{
  "referenceNumber": "1",
  "passengerPrices": [{
    "referenceNumber": "3",
    "amount": 500.0
  }
]
}
]

```

```

        }, {
            "referenceNumber": "2",
            "amount": 500.0
        }, {
            "referenceNumber": "4",
            "amount": 500.0
        }
    ]
}
]
}, {
    "code": "CBAG",
    "flightReferences": [{
        "referenceNumber": "1",
        "passengerPrices": [{
            "referenceNumber": "3",
            "amount": 2000.0
        }, {
            "referenceNumber": "2",
            "amount": 2000.0
        }, {
            "referenceNumber": "4",
            "amount": 2000.0
        }
    ]
}
]
}, {
    "code": "PETC",
    "flightReferences": [{
        "referenceNumber": "1",
        "passengerPrices": []
    }
}
]
}, {
    "code": "SNML",
    "flightReferences": [{
        "referenceNumber": "1",
        "passengerPrices": [{
            "referenceNumber": "3",
            "amount": 350.0
        }, {
            "referenceNumber": "2",
            "amount": 350.0
        }, {
            "referenceNumber": "4",
            "amount": 350.0
        }
    ]
}
]
}, {
    "code": "VGML",
    "flightReferences": [{
        "referenceNumber": "1",
        "passengerPrices": [{
            "referenceNumber": "3",

```

```

        "amount": 600.0
      }, {
        "referenceNumber": "2",
        "amount": 600.0
      }, {
        "referenceNumber": "4",
        "amount": 600.0
      }
    ]
  }
]
}, {
  "code": "BBAG",
  "flightReferences": [{
    "referenceNumber": "1",
    "passengerPrices": [{
      "referenceNumber": "3",
      "amount": 2000.0
    }, {
      "referenceNumber": "2",
      "amount": 2000.0
    }, {
      "referenceNumber": "4",
      "amount": 2000.0
    }
  ]
}
]
}
],
"packs": [],
"seatServices": [{
  "seatCharacteristics": [],
  "code": "SEAT",
  "flightReferences": [{
    "referenceNumber": "1",
    "passengerPrices": [{
      "referenceNumber": "3",
      "amount": 0.0
    }, {
      "referenceNumber": "2",
      "amount": 0.0
    }, {
      "referenceNumber": "4",
      "amount": 0.0
    }
  ]
}
]
}
],
"petsServiceConfiguration": [{
  "petsList": ["cat", "dog", "canary", "parrot"],
  "flightReferences": ["1"]
}
],
"portions": [{

```

```

        "type": "BTU",
        "number": "1",
        "flightReferenceNumbers": ["1"]
    }
],
"serviceCategories": [{
    "code": "URFL",
    "category": "insurance",
    "order": 4,
    "fullDirection": false
}, {
    "code": "URTR",
    "category": "insurance",
    "order": 4,
    "fullDirection": false
}, {
    "code": "SEAT",
    "category": "seat",
    "order": 1,
    "fullDirection": false
}, {
    "code": "ABAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
}, {
    "code": "BBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
}, {
    "code": "CBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
}, {
    "code": "HBAG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
}, {
    "code": "PETC",
    "category": "pets",
    "order": 4,
    "fullDirection": true
}, {
    "code": "CHML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
}, {
    "code": "HTML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
}, {
    "code": "MOML",

```

```

    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "SFML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "SNML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "SPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "VGML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "KSML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "CAML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "BGML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "AVML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "HNML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "VJML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "TCML",
    "category": "meal",
    "order": 2,

```

```

    "fullDirection": false
  }, {
    "code": "GFML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "BBML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "CPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "MPML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "GRML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "CBBG",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  }, {
    "code": "RVML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "VLML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "VOML",
    "category": "meal",
    "order": 2,
    "fullDirection": false
  }, {
    "code": "BIKE",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  }, {
    "code": "SKI",
    "category": "luggage",
    "order": 3,
    "fullDirection": true
  }, {

```

```

        "code": "DBML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "BLML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "EXST",
        "category": "seat",
        "order": 1,
        "fullDirection": false
    }, {
        "code": "AVIH",
        "category": "pets",
        "order": 4,
        "fullDirection": true
    }, {
        "code": "SVAN",
        "category": "pets",
        "order": 4,
        "fullDirection": true
    }, {
        "code": "CHNM",
        "category": "data",
        "order": 6,
        "fullDirection": true
    }, {
        "code": "NAMM",
        "category": "data",
        "order": 6,
        "fullDirection": true
    }, {
        "code": "NAME",
        "category": "data",
        "order": 6,
        "fullDirection": true
    }, {
        "code": "MEQT",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "GPST",
        "category": "seat",
        "order": 1,
        "fullDirection": false
    }, {
        "code": "BULK",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "LCML",
        "category": "meal",

```



```

        "order": 2,
        "fullDirection": false
    }, {
        "code": "WEAP",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "FPML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "LFML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "NLML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "LSML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "NFML",
        "category": "meal",
        "order": 2,
        "fullDirection": false
    }, {
        "code": "XBAG",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "HBGS",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "HBGT",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "GOLF",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "SPEQ",
        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }

```

```

}, {
  "code": "BUSS",
  "category": "other",
  "order": 7,
  "fullDirection": true
}, {
  "code": "DBGS",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "DBGT",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "XBLS",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "TIME",
  "category": "other",
  "order": 7,
  "fullDirection": true
}, {
  "code": "FRAG",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "JPML",
  "category": "meal",
  "order": 2,
  "fullDirection": false
}, {
  "code": "WBAG",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "PFEE",
  "category": "data",
  "order": 6,
  "fullDirection": true
}, {
  "code": "SRXBLL",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "SRXBLS",
  "category": "luggage",
  "order": 3,
  "fullDirection": true
}, {
  "code": "XBLL",

```

```

        "category": "luggage",
        "order": 3,
        "fullDirection": true
    }, {
        "code": "UMNR",
        "category": "other",
        "order": 7,
        "fullDirection": true
    }, {
        "code": "PAKA",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PAKB",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PAKC",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PAKD",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PAKE",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PAKF",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PKDP",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }, {
        "code": "PKEP",
        "category": "pack",
        "order": 0,
        "fullDirection": true
    }
    ],
    "serviceOptions": [{
        "code": "HTML",
        "preselect": true,
        "flightReferences": ["1"],
        "category": "meal",
        "priority": 1
    }, {

```

```
        "code": "SNML",
        "preselect": true,
        "flightReferences": ["1"],
        "category": "meal",
        "priority": 0
    }, {
        "code": "URFL",
        "preselect": true,
        "flightReferences": ["1"],
        "category": "insurance",
        "priority": 0
    }
],
    "moneyWallCoefficient": 1.21
},
"errors": [],
"success": true
}
```