



# **API системы "Колибри"**

**версия 1.0.0**

## Содержание

Общая информация.....	3
Концепции.....	3
Технические особенности.....	5
Сервисы.....	6
/AnnounceService.....	6
connectAct.....	6
connectShow.....	7
deleteAnnounce.....	7
getAgentAcceptedAnnounceReport.....	8
getAgentNewAnnounceReport.....	9
rejectShow.....	10
/AuthService.....	11
getSession.....	11
logout.....	11
/PartnerService.....	12
confirmPartner.....	12
deletePartner.....	12
deleteRequest.....	13
listPartners.....	13
listRequests.....	14
requestPartner.....	14
/PlaceService.....	16
listCities.....	16
listCountries.....	16
listPlaces.....	17
/TicketOfficeService.....	18
buyTicket.....	18
cancelReserve.....	19
getAgentFee.....	20
getOrder.....	21
getOrderStatus.....	22
getReturnSumm.....	23
getTicketsWoSeat.....	23
listShows.....	24
listTicket.....	25
listTicketTypes.....	26
reserveTicket.....	27
returnTickets.....	28
setPaid.....	28
Приложение.....	30
Пример работы с веб-сервисами на javascript.....	30

## Общая информация

API предназначено для интеграции билетных систем агентов с системой "Колибри". API предоставляет агенту возможность выполнения всех операций, которые ему необходимы при организации продаж билетов из системы "Колибри" через свою систему.

## Концепции

API системы "Колибри" позволяет агентам продавать билеты из доступных им билетных квот на эксклюзивных либо конкурентных условиях. Условия распространения билетов для каждой квоты задаёт организатор мероприятия, билеты на которое составляют квоту.

*Мероприятием* в системе считается некое организованное представление или зрелище, которое может проводиться один или несколько раз. В соответствии с этим, мероприятие может включать в себя один или несколько *сеансов*, характеризующихся *временем начала* сеанса. На каждый сеанс организатор мероприятия распространяет *билеты*. В рамках одного сеанса билеты объединяются в *билетные квоты*, для каждой из которых организатор задаёт условия распространения (размер агентского вознаграждения за продажу билетов из квоты). Билетная квота может быть *открытой* - и доступной для продажи в кассах всех агентов, присоединённых к системе, которых устроили условия присоединения; и *закрытой*: в этом случае она доступна только самому организатору, либо одному или нескольким выбранным организатором агентам-партнёрам.

Для организации партнёрства между организаторами мероприятий и агентами предназначен сервис *PartnerService*. С его помощью можно:

- послать заявку на партнёрство (функция *requestPartner*);
- принять (*confirmPartner*) или отклонить (*deleteRequest*) заявку, пришедшую от организатора мероприятий;
- получить список всех пришедших заявок (*listRequests*);
- посмотреть текущий список своих партнёров (*listPartners*);
- разорвать партнёрство с кем-либо из списка партнёров (*deletePartner*).

После того как агент станет партнёром какого-либо организатора мероприятий, этот организатор сможет устанавливать для этого агента персональные условия по агентскому вознаграждению за продажу билетов на свои мероприятия или отдельные сеансы. В случае если такие условия не заданы, агент может продавать билеты на условиях, определённых организатором "для всех". При этом ему доступны только открытые квоты билетов. Если же условия агентского вознаграждения определены, они действуют для агента при продаже билетов из любых квот: как открытых, так и выделенных персонально ему.

Информация о новых сеансах и условиях агентского вознаграждения посылается организатору мероприятия в виде *оферт*. Работу с офертами обеспечивает сервис *AnnounceService*. Он позволяет:

- получить список ожидающих принятия оферт (*getAgentNewAnnounceReport*);
- получить список уже принятых оферт (*getAgentAcceptedAnnounceReport*);
- принять оферту для конкретного сеанса (*connectShow*) либо всего мероприятия (*connectAct*);
- отклонить оферту (*rejectShow*) - при этом отклоняются все оферты на данное мероприятие;
- удалить оферту (*deleteAnnounce*).

При принятии оферт агент указывает, в каких из его касс будут продаваться билеты на включённые в оферты сеансы. В случае удаления принятой оферты подключение сеансов к кассам сохраняется.

Приняв пришедшие оферты, агент может продавать билеты из доступных ему квот. Для этого предназначен сервис обеспечения работы кассы *TicketOfficeService*. Он даёт возможность агенту:

- получить список доступных кассе сеансов мероприятий (*listShow*);
- получить список доступных *типов билетов* ("обычный", "детский" и т.п.);
- получить список доступных для продажи билетов на сеанс или на всё мероприятие (*getTicketsWoSeat* и *listTicket* для билетов без мест и с местами соответственно);
- забронировать билеты (*reserveTicket*) и отменить бронирование (*cancelReserve*);
- получить параметры заказа до его оплаты (*getOrder*);
- получить параметры своего агентского вознаграждения (*getAgentFee*);
- оформить заказ (*buyTicket*), сообщить о факте оплаты заказа (*setPaid*) и проверить статус оплаты (*getOrderStatus*);
- осуществить возврат проданных ранее билетов (*returnTickets*).

Кроме того, в системе предусмотрен информационный сервис *PlaceService*, при помощи которого можно получить список всех доступных агенту площадок для проведения мероприятий (*listPlaces*), а также списки городов (*listCities*) и стран (*listCountries*), в которых эти площадки располагаются.

## Технические особенности

Серверная часть предоставляет веб-сервисы, общение с которыми происходит посредством SOAP-протокола. В качестве транспортного уровня используется HTTPS. Поддерживаются методы GET и POST. Все примеры в настоящей документации приведены в расчёте на применение метода GET. При использовании метода POST в сокет передаётся часть запроса после подстроки "?query=". В этом случае эту часть запроса вместе с подстрокой нужно удалить из URL.

Результат обработки запроса может возвращаться в формате JSON, JSONP, либо XML. Для получения ответа в формате JSON, необходимо указать в заголовке тип сообщения "application/json". Для получения ответа в формате JSONP, в запрос кроме параметра query добавляется параметр callback="<имя функции>". В остальных случаях ответ будет в виде XML.

Если какое-либо поле в ответе имеет неопределённое значение, оно в ответ не включается. Если значение какого-либо поля в ответе является массивом, и этот массив пустой, то такое поле также опускается. В случае, когда такой массив содержит только один элемент, этот элемент возвращается как простая переменная соответствующего типа. Этих же принципов можно придерживаться и при формировании запросов. Однако корректной также будет передача во входных параметрах функций пустых массивов и неопределённых значений.

Некоторые поля в запросах являются обязательными и должны быть всегда заполнены. В описании формата параметров вызова функций такие поля выделяются шрифтом.

Вместе с каждым запросом передаётся идентификатор сессии @sessionid. Этот идентификатор получается до начала выполнения других операций с API функцией getSession сервиса AuthService. Запрос к этой функции должен включать в себя ключ доступа к API, который агент может получить в клиентском веб-интерфейсе системы "Колибри". Если сессия больше не нужна, её можно закрыть при помощи функции logout сервиса AuthService. В случае если агент запросил новый идентификатор сессии, его предыдущий идентификатор сессии становится недействительным. Кроме того, в случае запроса через веб-интерфейс нового ключа доступа к API текущая сессия будет закрыта.

Вместе с каждым ответом сервиса приходит код возврата @rc. В случае успешного выполнения запроса этот код будет равен "200". Любой другой код свидетельствует о возникшей во время выполнения запроса ошибки, текст которой также будет возвращён в ответе сервиса.

# Сервисы

Протокол: **https**

Адрес сервера: **service.biletcolibri.ru**, порт **443**

Адрес тестового сервера: **testservice.biletcolibri.ru**, порт **8443**

Базовый путь: **/data/services**

Вход: application/json

Выход: application/json | application/jsonp | application/xml

Далее приводится описание всех доступных агенту сервисов и их функций в алфавитном порядке.

## /AnnounceService

### connectAct

#### Описание

Присоединить мероприятие, для которого получена оферта, к кассам

#### Запрос

<b>connectAct</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>SetActQuery</b>	object	
<b>actId</b>	numeric	Идентификатор мероприятия
<b>replace</b>	boolean	Если true, то, в случае наличия существующих подключений, перезаписывать их
<b>offices</b>	object	Массив идентификаторов касс. Если отсутствует, подключаются все кассы
<b>Long</b>	numeric array	

#### Ответ

<b>connectActResponse</b>	object	
<b>@rc</b>	string	Код возврата

#### Пример вызова

```
/AnnounceService?query={"connectAct":{"@sessionId":"<sessionId>","SetActQuery":{"actId":3,"offices":{"Long":[62,64]}}}}
```

## connectShow

### Описание

Присоединить сеанс, для которого получена оферта, к кассам

### Запрос

<b>connectShow</b>	object	
<b>@sessionid</b>	string	Идентификатор сессии
<b>SetActQuery</b>	object	
<b>showId</b>	numeric	Идентификатор сеанса
<b>officeId</b>	numeric array	Массив идентификаторов касс. Если отсутствует, подключаются все кассы

### Ответ

<b>connectShowResponse</b>	object	
<b>@rc</b>	string	Код возврата

### Пример вызова

```
/AnnounceService?query={"connectShow":{"@sessionid":"<sessionid>","showId":7,"officeId":  
[62,64]}}
```

## deleteAnnounce

### Описание

Удалить оферты

### Запрос

<b>deleteAnnounce</b>	object	
<b>@sessionid</b>	string	Идентификатор сессии
<b>id</b>	numeric array	Массив идентификаторов удаляемых оферт

### Ответ

<b>deleteAnnounceResponse</b>	object	
<b>@rc</b>	string	Код возврата

### Пример вызова

```
/AnnounceService?query={"deleteAnnounce":{"@sessionid":"<sessionid>","id":[2,3]}}
```

## getAgentAcceptedAnnounceReport

### Описание

Получить список уже принятых оферт

### Запрос

**getAgentAcceptedAnnounceReport** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

### Ответ

**getAgentAcceptedAnnounceReportResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

AnnounceReport	object array
----------------	--------------

clientId	numeric	Идентификатор клиента
----------	---------	-----------------------

clientName	string	Название клиента
------------	--------	------------------

actorgId	numeric	Идентификатор организатора мероприятия
----------	---------	--

actorgName	string	Название организатора мероприятия
------------	--------	-----------------------------------

actId	numeric	Идентификатор мероприятия
-------	---------	---------------------------

actName	string	Название мероприятия
---------	--------	----------------------

actTypes	string	Список жанров мероприятия
----------	--------	---------------------------

showStartFirst	datetime	Начало первого сеанса из оферты
----------------	----------	---------------------------------

showStartLast	datetime	Начало последнего сеанса из оферты
---------------	----------	------------------------------------

showCount	numeric	Количество сеансов в оферте
-----------	---------	-----------------------------

actTickets	numeric	Суммарное количество билетов на все сеансы мероприятия
------------	---------	--

showTickets	numeric	Количество билетов на один сеанс
-------------	---------	----------------------------------

minPrice	numeric	Минимальная цена билета
----------	---------	-------------------------

maxPrice	numeric	Максимальная цена билета
----------	---------	--------------------------

agentFixedFee	numeric	Предлагаемая агенту фиксированная часть агентского вознаграждения
---------------	---------	---

agentCoefficient	numeric	Предлагаемая агенту часть агентского вознаграждения в виде коэффициента к цене билета
------------------	---------	---

placeCountry	string	Страна проведения мероприятия
--------------	--------	-------------------------------

placeCity	string	Город проведения мероприятия
-----------	--------	------------------------------

placeId	numeric	Идентификатор площадки
---------	---------	------------------------

placeName	string	Название площадки
-----------	--------	-------------------

### Пример вызова

```
/AnnounceService?query={"getAgentAcceptedAnnounceReport":{"@sessionid":"<sessionid>"}}
```



## getAgentNewAnnounceReport

### Описание

Получить список новых оферт

### Запрос

**getAgentNewAnnounceReport** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

### Ответ

**getAgentNewAnnounceReportResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

AnnounceReport	object array
----------------	--------------

clientId	numeric	Идентификатор клиента
----------	---------	-----------------------

clientName	string	Название клиента
------------	--------	------------------

actorgId	numeric	Идентификатор организатора мероприятия
----------	---------	--

actorgName	string	Название организатора мероприятия
------------	--------	-----------------------------------

actId	numeric	Идентификатор мероприятия
-------	---------	---------------------------

actName	string	Название мероприятия
---------	--------	----------------------

actTypes	string	Список жанров мероприятия
----------	--------	---------------------------

showStartFirst	datetime	Начало первого сеанса из оферты
----------------	----------	---------------------------------

showStartLast	datetime	Начало последнего сеанса из оферты
---------------	----------	------------------------------------

showCount	numeric	Количество сеансов в оферте
-----------	---------	-----------------------------

actTickets	numeric	Суммарное количество билетов на все сеансы мероприятия
------------	---------	--

showTickets	numeric	Количество билетов на один сеанс
-------------	---------	----------------------------------

minPrice	numeric	Минимальная цена билета
----------	---------	-------------------------

maxPrice	numeric	Максимальная цена билета
----------	---------	--------------------------

agentFixedFee	numeric	Предлагаемая агенту фиксированная часть агентского вознаграждения
---------------	---------	---

agentCoefficient	numeric	Предлагаемая агенту часть агентского вознаграждения в виде коэффициента к цене билета
------------------	---------	---

placeCountry	string	Страна проведения мероприятия
--------------	--------	-------------------------------

placeCity	string	Город проведения мероприятия
-----------	--------	------------------------------

placeId	numeric	Идентификатор площадки
---------	---------	------------------------

placeName	string	Название площадки
-----------	--------	-------------------

### Пример вызова

```
/AnnounceService?query={"getAgentNewAnnounceReport":{"@sessionid":"<sessionid>"}}
```

## rejectShow

### Описание

Отклонить oferty на подключение сеанса

### Запрос

**rejectShow** object

<b>@sessionId</b>	string	Идентификатор сессии
<b>actId</b>	numeric array	Массив идентификаторов мероприятий, на которые отклоняются oferty

### Ответ

**rejectShowResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

### Пример вызова

```
/AnnounceService?query={"rejectShow":{"@sessionId":"<sessionId>","actId":[2,3]}}
```

## /AuthService

### getSession

#### Описание

Открыть новую сессию и получить её идентификатор

#### Запрос

**getSession** object

<b>key</b>	string	Ключ доступа к API, генерируется в веб-интерфейсе клиента системы "Колибри"
------------	--------	---

#### Ответ

**getSessionResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

SessionInfo	object	
-------------	--------	--

<b>id</b>	string	Идентификатор сессии
-----------	--------	----------------------

#### Пример вызова

```
/AuthService?query={"getSession":{"key":"<ключ доступа>"}}
```

### logout

#### Описание

Закрыть текущую сессию

#### Запрос

**logout** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

#### Ответ

**logoutResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

#### Пример вызова

```
/PartnerService?query={"logout":{"@sessionid":"<sessionid>"}}
```

## /PartnerService

### confirmPartner

#### Описание

Принять заявку на партнёрство

#### Запрос

<b>confirmPartner</b>	object	
<b>@sessionid</b>	string	Идентификатор сессии
<b>id</b>	numeric	Идентификатор клиента, заявка которого принимается

#### Ответ

<b>confirmPartnerResponse</b>	object	
<b>@rc</b>	string	Код возврата

#### Пример вызова

```
/PartnerService?query={"confirmPartner":{"@sessionid":"<sessionid>","id":129}}
```

### deletePartner

#### Описание

Удалить партнёров

#### Запрос

<b>deletePartner</b>	object	
<b>@sessionid</b>	string	Идентификатор сессии
<b>id</b>	numeric array	Массив идентификаторов удаляемых партнёров

#### Ответ

<b>deletePartnerResponse</b>	object	
<b>@rc</b>	string	Код возврата

#### Пример вызова

```
/PartnerService?query={"deletePartner":{"@sessionid":"<sessionid>","id":129}}
```

## deleteRequest

### Описание

Удалить заявки на партнёрство

### Запрос

<b>deleteRequest</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>id</b>	numeric	Идентификатор клиента, заявка которого удаляется

### Ответ

<b>deleteRequestResponse</b>	object	
<b>@rc</b>	string	Код возврата

### Пример вызова

```
/PartnerService?query={"deleteRequest":{"@sessionId":"<sessionId>","id":129}}
```

## listPartners

### Описание

Получить список партнёров

### Запрос

<b>listPartners</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии

### Ответ

<b>listPartnersResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>ClientBasic</b>	object array	
<b>id</b>	numeric	Идентификатор партнёра
<b>name</b>	string	Название партнёра
<b>email</b>	string	E-mail
<b>phone</b>	string	Телефон
<b>postalAddress</b>	string	Почтовый адрес
<b>site</b>	string	URL сайта
<b>brand</b>	string	Торговая марка
<b>fbPage</b>	string	Страница в Facebook

vkPage	string	Страница в Vkontakte
--------	--------	----------------------

## Примеры вызова

```
/PartnerService?query={"listPartners":{"@sessionid":"<sessionid>"}}
```

## listRequests

### Описание

Получить список заявок на партнёрство

### Запрос

**listRequests** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

### Ответ

**listRequestsResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

ClientPartner	object array
---------------	--------------

client	numeric	Идентификатор клиента
--------	---------	-----------------------

clientName	string	Название клиента
------------	--------	------------------

partner	numeric	Идентификатор предложившего партнёрство клиента
---------	---------	---

partnerName	string	Название предложившего партнёрство клиента
-------------	--------	--

requesttime	datetime	Время запроса на партнёрство
-------------	----------	------------------------------

## Примеры вызова

```
/PartnerService?query={"listRequests":{"@sessionid":"<sessionid>"}}
```

## requestPartner

### Описание

Послать заявку на партнёрство

### Запрос

**requestPartner** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

<b>id</b>	numeric	Идентификатор клиента, к которому отправляется заявка на партнёрство
-----------	---------	--

## Ответ

**requestPartnerResponse** object

**@rc** string Код возврата

## Пример вызова

```
/PartnerService?query={"requestPartner":{"@sessionId":"<sessionId>","id":129}}
```

## /PlaceService

### listCities

#### Описание

Получить список городов

#### Запрос

**listCities** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

#### Ответ

**listCitiesResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

City	object array
------	--------------

id	numeric	Идентификатор города
----	---------	----------------------

name	string	Название города
------	--------	-----------------

country	string	Идентификатор страны
---------	--------	----------------------

tz	string	Временная зона
----	--------	----------------

actual	boolean	Если false, запись устарела и должна быть проигнорирована
--------	---------	---

#### Примеры вызова

```
/PlaceService?query={"listCities":{"@sessionid":"<sessionid>"}}
```

### listCountries

#### Описание

Получить список стран

#### Запрос

**listCountries** object

<b>@sessionid</b>	string	Идентификатор сессии
-------------------	--------	----------------------

#### Ответ

**listCountriesResponse** object

<b>@rc</b>	string	Код возврата
------------	--------	--------------

Country	object array
---------	--------------



id	numeric	Идентификатор страны
name	string	Название страны
actual	boolean	Если false, запись устарела и должна быть проигнорирована

## Примеры вызова

```
/PlaceService?query={"listCountries":{"@sessionId":"<sessionId>"}}
```

## listPlaces

### Описание

Получить список площадок с сеансами

### Запрос

**listPlaces** object

@sessionId	string	Идентификатор сессии
------------	--------	----------------------

### Ответ

**listPlacesResponse** object

@rc	string	Код возврата
-----	--------	--------------

PlaceBasic	object array
------------	--------------

id	numeric	Идентификатор площадки
name	string	Название площадки
placeOpen	boolean	Если false, то площадка доступна только владельцу; иначе доступна всем
cityId	numeric	Идентификатор города
cityName	string	Название города
ownerId	numeric	Идентификатор владельца площадки
phone	string	Телефон площадки
address	string	Адрес площадки
howtoreach	string	Как добраться

## Примеры вызова

```
/PlaceService?query={"listPlaces":{"@sessionId":"<sessionId>"}}
```

## /TicketOfficeService

### buyTicket

#### Описание

Купить билеты (оформить заказ со статусом "неоплачено")

#### Запрос

<b>buyTicket</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>BuyQuery</b>	object	
<b>tickets</b>	object	Массив билетов в заказе
<b>BuyTicketQuery</b>	object array	
<b>ticketId</b>	numeric	Идентификатор билета (для билета с местом)
<b>typeId</b>	numeric	Идентификатор типа билета
<b>showId</b>	numeric	Идентификатор сеанса (для билетов без мест)
<b>count</b>	numeric	Количество билетов (для билетов без мест)
<b>txProcId</b>	numeric	Идентификатор платёжной системы
<b>discountCodes</b>	numeric array	Массив промокодов
<b>name</b>	string	ФИО покупателя (для печати на билетах)
<b>email</b>	string	E-mail покупателя (для отсылки билетов)

#### Ответ

<b>buyTicketResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>OrderBasic</b>	object	
<b>id</b>	numeric	Идентификатор заказа
<b>agentId</b>	numeric	Идентификатор агента
<b>agentName</b>	string	Название агента
<b>agentEmail</b>	string	E-mail агента
<b>actOrgId</b>	numeric	Идентификатор организатора мероприятия
<b>buyer</b>	string	ФИО покупателя
<b>actId</b>	numeric	Идентификатор мероприятия
<b>actName</b>	string	Название мероприятия
<b>time</b>	datetime	Время заказа
<b>summ</b>	numeric	Сумма заказа
<b>summForOrganizer</b>	numeric	Сумма, которую получит организатор мероприятия

status	numeric	Статус оплаты заказа (0 - новый заказ, 1 - отправлен в платёжную систему, 2 - неудача, 3 - платёж отменен, 4 - заказ оплачен)
statusName	string	Название статуса оплаты заказа
ticketOffice	string	Название кассы
ticketOfficeAddress	string	Адрес кассы
agentFee	numeric	Сумма агентского вознаграждения
txFee	numeric	Комиссия платёжной системы
systemFee	numeric	Комиссия билетной системы
freeTickets	numeric	Количество в заказе бесплатных билетов
paidTickets	numeric	Количество в заказе билетов с оплатой
tickets	object	Массив билетов в заказе
TicketBasic		
id	numeric	Идентификатор билета
code	string	Код билета
passout	boolean	Если true, то билет является контрамаркой

## Примеры вызова

### Купить билет типа 306 с местом:

```
/TicketOfficeService?query={"buyTicket":{"@sessionId":"<sessionId>","BuyQuery":{"discountCodes":{"String":[123,456]},"tickets":{"BuyTicketQuery":{"typeId":306,"ticketId":140599}}}}
```

### Купить билеты типа 1 и 2 с местами по их идентификаторам (могут быть в любом сеансе); купить 3 билета типа 2 без мест на сеанс 100; учесть коды скидок 123 и 456:

```
/TicketOfficeService?query={"buyTicket":{"@sessionId":"<sessionId>","BuyQuery":{"discountCodes":{"String":[123,456]},"tickets":{"BuyTicketQuery":[{"typeId":1,"ticketId":3309}, {"typeId":2,"ticketId":3310}, {"showId":100,"typeId":2,"count":3}]}}}}
```

### Купить билеты с местом по идентификатору и билеты без мест в разных сеансах; учесть коды скидок:

```
/TicketOfficeService?query={"buyTicket":{"@sessionId":"<sessionId>","BuyQuery":{"discountCodes":{"String":[123,456]},"tickets":{"BuyTicketQuery":[{"typeId":306,"ticketId":140599}, {"showId":101,"typeId":305,"count":2}, {"showId":102,"typeId":306,"count":3}]}}}}
```

## cancelReserve

### Описание

Отменить бронирование билетов

### Запрос

cancelReserve object

@sessionId string Идентификатор сессии

CancelReserveQuery object

ticketsWSeat	numeric array	Массив идентификаторов билетов с местами для отмены брони
ticketsWOSeat	object array	Массив объектов для отмены брони билетов без мест
<b>BuyTicketQuery</b> object array		
ticketId	numeric	Идентификатор билета (для билета с местом)
typeId	numeric	Идентификатор типа билета
showId	numeric	Идентификатор сеанса (для билетов без мест)
count	numeric	Количество билетов (для билетов без мест)

## Ответ

**cancelReserveResponse** object

**@rc** string Код возврата

## Примеры вызова

```
/TicketOfficeService?query={"cancelReserve":{"@sessionId":"<sessionId>","ticketId":140598}}
```

```
/TicketOfficeService?query={"cancelReserve":{"@sessionId":"<sessionId>","showId":140598}}
```

## getAgentFee

### Описание

Получить параметры агентского вознаграждения при продаже билетов на мероприятие

### Запрос

**getAgentFee** object

**@sessionId** string Идентификатор сессии

**actId** numeric Идентификатор мероприятия

## Ответ

**getAgentFeeResponse** object

**@rc** string Код возврата

**ShowAgent** object array

id numeric Идентификатор записи

act numeric Идентификатор мероприятия

show numeric Идентификатор сеанса

agent numeric Идентификатор агента

fixedFee numeric Фиксированная часть агентского вознаграждения

coefficient numeric Часть агентского вознаграждения в виде коэффициента к цене билета

## Пример вызова

```
/TicketOfficeService?query={"getAgentFee":{"@sessionId":"<sessionId>","actId":123}}
```

## getOrder

### Описание

Получить предварительные параметры заказа до его реального формирования

### Запрос

<b>getOrder</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>BuyQuery</b>	object	
<b>tickets</b>	object	Массив билетов в заказе
<b>BuyTicketQuery</b>	object array	
<b>ticketId</b>	numeric	Идентификатор билета (для билета с местом)
<b>typeId</b>	numeric	Идентификатор типа билета
<b>showId</b>	numeric	Идентификатор сеанса (для билетов без мест)
<b>count</b>	numeric	Количество билетов (для билетов без мест)
<b>txProcId</b>	numeric	Идентификатор платёжной системы
<b>discountCodes</b>	numeric array	Массив промокодов

### Ответ

<b>getOrderResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>OrderBasic</b>	object	
<b>id</b>	numeric	Идентификатор заказа
<b>agentId</b>	numeric	Идентификатор агента
<b>agentName</b>	string	Название агента
<b>agentEmail</b>	string	Е-mail агента
<b>actOrgId</b>	numeric	Идентификатор организатора мероприятия
<b>buyer</b>	string	ФИО покупателя
<b>actId</b>	numeric	Идентификатор мероприятия
<b>actName</b>	string	Название мероприятия
<b>time</b>	datetime	Время заказа
<b>summ</b>	numeric	Сумма заказа
<b>summForOrganizer</b>	numeric	Сумма, которую получит организатор мероприятия

status	numeric	Статус оплаты заказа (0 - новый заказ, 1 - отправлен в платёжную систему, 2 - неудача, 3 - платёж отменен, 4 - заказ оплачен)
statusName	string	Название статуса оплаты заказа
ticketOffice	string	Название кассы
ticketOfficeAddress	string	Адрес кассы
agentFee	numeric	Сумма агентского вознаграждения
txFee	numeric	Комиссия платёжной системы
systemFee	numeric	Комиссия билетной системы
freeTickets	numeric	Количество в заказе бесплатных билетов
paidTickets	numeric	Количество в заказе билетов с оплатой
tickets	object	Массив билетов в заказе
TicketBasic	object array	
id	numeric	Идентификатор билета
code	string	Код билета
passout	boolean	Если true, то билет является контрамаркой

## Пример вызова

```
/TicketOfficeService?query={"getOrder":{"@sessionId":"<sessionId>","BuyQuery":{"discountCodes":{"String":[123,456]},"tickets":{"BuyTicketQuery":{"typeId":306,"ticketId":140599}}}}
```

## getOrderStatus

### Описание

Получить статус оплаты заказа

### Запрос

<b>getOrderStatus</b>	object	
@sessionId	string	Идентификатор сессии
orderId	numeric	Идентификатор заказа

### Ответ

<b>getOrderStatusResponse</b>	object	
@rc	string	Код возврата
status	numeric	Статус оплаты заказа (0 - новый заказ, 1 - отправлен в платёжную систему, 2 - неудача, 3 - платёж отменен, 4 - заказ оплачен)

## Пример вызова

```
/TicketOfficeService?query={"returnTickets":{"@sessionId":"<sessionId>","code":1011034865}}
```

## getReturnSumm

### Описание

Рассчитать суммы возврата для возвращаемых билетов

### Запрос

<b>getReturnSumm</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>code</b>	numeric array	Массив кодов возвращаемых билетов

### Ответ

<b>getReturnSummResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>BigDecimal</b>	numeric array	Массив сумм возврата в порядке следования переданных в запросе кодов билета

### Пример вызова

```
/TicketOfficeService?query={"getReturnSumm":{"@sessionId":"<sessionId>","code":1011034865}}
```

## getTicketsWoSeat

### Описание

Получить цены на билеты без мест

### Запрос

<b>getTicketsWoSeat</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>showId</b>	numeric	Идентификатор сеанса

### Ответ

<b>getTicketsWoSeatResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>TicketWOSeatPrice</b>	object array	
<b>show</b>	numeric	Идентификатор сеанса
<b>type</b>	numeric	Идентификатор типа билета
<b>typeName</b>	string	Название типа билета
<b>price</b>	numeric	Цена билета данного типа
<b>standart</b>	boolean	Если true, то тип является стандартным

## Пример вызова

```
/TicketOfficeService?query={"getTicketsWoSeat":{"@sessionId":"<sessionId>","showId":100}}
```

## listShows

### Описание

Получить список сеансов, которые подключены к кассе

### Запрос

**listShows** object

<b>@sessionId</b>	string	Идентификатор сессии
<b>ShowQuery</b>	object	
actId	numeric	Идентификатор мероприятия
showId	numeric	Идентификатор сеанса
start	datetime	Начало сеанса с
stop	datetime	Начало сеанса по

### Ответ

**listShowsResponse** object

<b>@rc</b>	string	Код возврата
Show	object array	
id	numeric	Идентификатор сеанса
actId	numeric	Идентификатор мероприятия
actName	string	Название мероприятия
start	datetime	Начало сеанса
place	numeric	Идентификатор площадки
map	numeric	Идентификатор карты мест
startTime	datetime	Время начала сеанса
stopTime	datetime	Время завершения сеанса
saleStart	datetime	Время начала продаж
saleStop	datetime	Время завершения продаж
checkStart	datetime	Время начала контроля билетов
checkStop	datetime	Время завершения контроля билетов
cancelled	boolean	Если true, то сеанс отменён, продажи невозможны
ticketsWoSeat	numeric	Количество билетов без мест
updatingTime	datetime	Время последнего обновления записи
txFeeInPrice	boolean	Если true, комиссия платёжной системы включается в цену билета



feesExcluded	boolean	Если true, комиссии агента и билетной системы добавляются к цене билета
minPrice	numeric	Минимальная цена билета
maxPrice	numeric	Максимальная цена билета

## Примеры вызова

### Список любых сеансов с ограничением по времени начала:

```
https://service.biletcolibri.ru/data/services/TicketOfficeService?query={"listShow": {"@sessionId": "<sessionId>", "ShowQuery": {"stop": "18.06.2012 00:00:00", "start": "18.01.2012 00:00:00"}}
```

### Список сеансов конкретного мероприятия с ограничением по времени начала:

```
https://service.biletcolibri.ru/data/services/TicketOfficeService?query={"listShow": {"@sessionId": "<sessionId>", "ShowQuery": {"stop": "18.06.2012 00:00:00", "start": "18.01.2012 00:00:00", "actId": "144"}}
```

## listTicket

### Описание

Получить список билетов на мероприятие/сеанс

### Запрос

**listTicket** object

**@sessionId** string Идентификатор сессии

**TicketQuery** object

actId numeric Идентификатор мероприятия

showId numeric Идентификатор сеанса

maxTicks datetime Максимальное количество билетов в ответе

seatIds object Массив идентификаторов мест

Long numeric array

### Ответ

**listTicketResponse** object

**@rc** string Код возврата

Ticket object array

id numeric Идентификатор билета

act numeric Идентификатор мероприятия

show numeric Идентификатор сеанса

map numeric Идентификатор карты мест

seat numeric Идентификатор места

block numeric Идентификатор билетной квоты

status	numeric	Статус билета (1 - в продаже, 2 - забронирован, 3 - продан, 4 - пройден контроль, 5 - временный выход)
code	string	Код билета

## Примеры вызова

```
/TicketOfficeService?query={"listTicket":{"@sessionId":"<sessionId>","TicketQuery":{"seatId":3318,"showId":100}}}
```

```
/TicketOfficeService?query={"listTicket":{"@sessionId":"<sessionId>","TicketQuery":{"maxTicks":1,"showId":100,"seatIds":{"Long":[111,222]}}}}
```

## listTicketTypes

### Описание

Получить список доступных для мероприятия типов билетов

### Запрос

**listTicketTypes** object

<b>@sessionId</b>	string	Идентификатор сессии
<b>actId</b>	numeric	Идентификатор мероприятия

### Ответ

**listTicketTypesResponse** object

<b>@rc</b>	string	Код возврата
<b>TicketType</b>	object array	Массив типов билета
id	numeric	Идентификатор типа билета
name	string	Название типа
act	numeric	Идентификатор мероприятия
defPrice	numeric	Цена по умолчанию
standart	boolean	Если true, то тип является стандартным

## Примеры вызова

```
/TicketOfficeService?query={"listTicketTypes":{"@sessionId":"<sessionId>","actId":144}}
```

## reserveTicket

### Описание

Забронировать билеты

### Запрос

<b>reserveTicket</b>	object	
<b>@sessionId</b>	string	Идентификатор сессии
<b>ReserveTicketQuery</b>	object	
showId	numeric	Идентификатор сеанса
ticketsWOSeat	numeric	Количество билетов без мест
tickets	object	Массив идентификаторов билетов с местом для бронирования
Long	numeric array	
ticketWoSeatQuery	object	Массив параметров для бронирования билетов без мест
TicketTypeCount	object array	
type	numeric	Идентификатор типа билета
count	numeric	Количество билетов
ticketQuery	object	Массив параметров для бронирования билетов с местами
TicketTypeCount	object array	
id	numeric	Идентификатор билета
type	numeric	Идентификатор типа билета

### Ответ

<b>cancelReserveResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>id</b>	numeric array	Массив идентификаторов забронированных билетов

### Пример вызова

**Забронировать 3 билета на сеанс 100 без мест и 2 билета с местами:**

```
/TicketOfficeService?query={"reserveTicket":{"@sessionId":"<sessionId>","ReserveTicketQuery":{"showId":100,"ticketsWOSeat":3,"tickets":{"Long":[3318,3309]}}}}
```

## returnTickets

### Описание

Провести возврат билетов

### Запрос

<b>returnTickets</b>	object	
<b>@sessionid</b>	string	Идентификатор сессии
<b>code</b>	numeric array	Массив кодов возвращаемых билетов

### Ответ

<b>returnTicketsResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>BigDecimal</b>	numeric	Сумма возврата

### Пример вызова

```
/TicketOfficeService?query={"returnTickets":{"@sessionid":"<sessionid>","code":1011034865}}
```

## setPaid

### Описание

Установить заказу статус "оплачен"

### Запрос

<b>setPaid</b>	object	
<b>@sessionid</b>	string	Идентификатор сессии
<b>orderId</b>	numeric	Идентификатор заказа

### Ответ

<b>setPaidResponse</b>	object	
<b>@rc</b>	string	Код возврата
<b>OrderBasic</b>	object	

id	numeric	Идентификатор заказа
agentId	numeric	Идентификатор агента
agentName	string	Название агента
agentEmail	string	E-mail агента
actOrgId	numeric	Идентификатор организатора мероприятия
buyer	string	ФИО покупателя
actId	numeric	Идентификатор мероприятия
actName	string	Название мероприятия
time	datetime	Время заказа
summ	numeric	Сумма заказа
summForOrganizer	numeric	Сумма, которую получит организатор мероприятия
status	numeric	Статус оплаты заказа (0 - новый заказ, 1 - отправлен в платёжную систему, 2 - неудача, 3 - платёж отменен, 4 - заказ оплачен)
statusName	string	Название статуса оплаты заказа
ticketOffice	string	Название кассы
ticketOfficeAddress	string	Адрес кассы
agentFee	numeric	Сумма агентского вознаграждения
txFee	numeric	Комиссия платёжной системы
systemFee	numeric	Комиссия билетной системы
freeTickets	numeric	Количество в заказе бесплатных билетов
paidTickets	numeric	Количество в заказе билетов с оплатой
tickets	object	Массив билетов в заказе
TicketBasic	object array	
id	numeric	Идентификатор билета
code	string	Код билета
passout	boolean	Если true, то билет является контрамаркой

## Пример вызова

```
/TicketOfficeService?query={"setPaid":{"@sessionId":"<sessionId>","orderId":1880}}
```

# Приложение

## Пример работы с веб-сервисами на javascript

Ниже приводится фрагмент библиотеки для работы с веб-сервисами на javascript. В коде используются функции библиотеки jQuery. Обращение происходит с использованием метода POST, результат запрашивается в формате JSON. Для упрощения примера из кода удалена функция обработки ошибок сервиса по умолчанию.

```
var service = {
  //url : 'https://testservice.biletcolibri.ru:8443/data/services/',
  url : 'https://service.biletcolibri.ru/data/services/',
  sessionId : undefined,

  res2arr : function (res) {
    return res !== undefined ? $.isArray(res) ? res : [ res ] : [ ];
  },

  sendRequest : function (serviceName, query, doneHandler, failHandler) {
    $.ajax({
      url : this.url + serviceName,
      type : 'POST', // 'GET',
      data : JSON.stringify(query), // 'query=' + JSON.stringify(query),
      processData : false,
      cache : false,
      contentType : 'application/json; charset=UTF-8'
    }).done(function (msg, textStatus) {
      if (doneHandler !== undefined && $.isFunction(doneHandler)) {
        doneHandler(msg);
      }
    }).fail(function (jqXHR, textStatus) {
      debug(arguments);
      debug(jqXHR);
      debug(textStatus);
      if (failHandler !== undefined && $.isFunction(failHandler)) {
        failHandler(jqXHR, serviceName, query, doneHandler);
      }
    });
  },

  authService : function () {
    [].unshift.call(arguments, 'AuthService');
    this.sendRequest.apply(this, arguments);
  },

  announceService : function () {
    [].unshift.call(arguments, 'AnnounceService');
    this.sendRequest.apply(this, arguments);
  },

  [ ... ]

  getSession : function (accessKey, doneHandler, failHandler, authErrorHandler) {
    var self = this;
    this.authService({
      getSession : {
        key : accessKey
      }
    }, function (msg) {
      var res = msg.getSessionResponse;
      switch (res['@rc']) {
        case '200':
          self.sessionId = self.res.SessionInfo.id;
          doneHandler();
          break;
        case '500':
          if ($.isFunction(authErrorHandler)) { authErrorHandler(); }
          else if ($.isFunction(this.defaultAuthHandler)) { this.defaultAuthHandler(); }
          break;
      }
    });
  }
};
```

```
        default:
console.log('unknown response: ', msg.authResponse);
    }
    }, failHandler);
},

getAgentAcceptedAnnounceReport : function (doneHandler) {
    var self = this;
    this.authService({
        getSelf : {
            '@sessionId' : self.sessionId
        }
    }, function (msg) {
        var res = msg.getAgentAcceptedAnnounceReport;
        switch (res['@rc']) {
            case '200':
                doneHandler(self.res2arr(res.AnnounceReport));
                break;
        }
    });
},

[ ... ]
```