iikoNET. POS API v2 (extended)

[Введение](#h.c662isxbc4js)

[Форматы](#h.fqygrq4ia46s)

[Методы](#h.vkup352hjw93)

[Пакетные вызовы (батчинг)](#h.cdwbfjfvf57h)

[Проверка соединения](#h.yheh3cuydo06)

[Авторизация](#h.jakjww31ym6a)

[Настройки](#h.vogqj9o9v91h)

[Пользователи](#h.ooiku3cefbwn)

[Поиск пользователя](#h.otj5mxmmr7ko)

[Получить информацию о пользователе](#h.5fuz1qx29fk7)

[Получить фотографию пользователя](#h.b058qybb8zy7)

[Получить Url фотографии пользователя](#h.ipxna28oodoa)

[Привязать внешний идентификатор к пользователю](#h.hji5dkq5jeko)

[Заказы](#h.w59j65j13nqu)

[Создать или обновить заказ](#h.s57vv0jjuuu)

[Привязать заказ к пользователю](#h.ixlrij9v4fa4)

[Отвязать заказ от пользователя](#h.j8eegfewyv3x)

[Кошельки](#h.k3j7avprmgy2)

[Получить список кошельков пользователя](#h.68zoyy9n6e40)

[Получить баланс кошелька](#h.xsw24g2ai31z)

[Пополнить баланс кошелька](#h.5k8u6xt1khd2)

[Платежи](#h.7mmxehybrcuj)

[Оплата заказа](#h.ag1zz5ny9tg2)

[Отмена оплаты](#h.vu0p8e38xmvn)

[Получить оплаты iikoNet для заказа](#h.d7w82lc3g5w4)

[Получить максимальную сумму, которую можно оплатить заданным типом оплаты](#h.53aptpdxrhc3)

[Проверить возможность оплаты заданной суммы в заказе заданным типом оплаты](#h.bk9gwsxp9jmu)

[Возврат заказа (отмена всех оплат и бонусов)](#h.pbk19qoau20)

[Программы лояльности](#h.1qitc5q80lpx)

[Получить текст пречека в виде разметки](#h.pco2io5qwuax)

[Получить текст чека в виде разметки](#h.9xpkmfmyzjpb)

[Получить текст пречека в виде текста](#h.yr5221xbja8p)

[Получить текст чека в виде текста](#h.e6732uei6sl6)

[Привязать промокод к заказу](#h.5f5gz5ktexvz)

[Предварительный расчет бонусов (для добавления в заказ скидок и бесплатных блюд)](#h.z7r09seq01ox)

[Задать примененные к заказу программы лояльности (скидки)](#h.tamvza56lx30)

[Задать оплаты, с помощью которых закрыт заказ](#h.799ji0sovcv9)

[Задать сумму, на которую могут быть начислены бонусы по каждому из кошельков](#h.1xjtg3w3165a)

[Закрыть заказ и произвести окончательный расчет бонусов](#h.7z5xc6i7zk12)

[Типы данных](#h.nqmjynjbu3nv)

[CardMaskInfo](#h.57v2bnqox10d)

[CardMaskInfoType](#h.9jskk9b8xt69)

[ExternalIdType](#h.m6nh87nb023i)

[FrontSettings](#h.5n0tyhdvvjos)

[LoyaltyOperation](#h.7xj8ksdeklwr)

[LoyaltyProgramResult](#h.v0bvx84lwu98)

[LoyaltyResult](#h.mvgzrk57gewl)

[Order](#h.57v2bnqox10d)

[OrderItem](#h.nfct2zt1jz8g)

[OrderType](#h.l2naw7oabrk0)

[ParamType](#h.ofxjt8862rcs)

[UserData](#h.jfi5atjg1aq9)

[UserRefParam](#h.rd83gwatp10)

[UserSearchResult](#h.xu5u715hy5ae)

[UserSearchScope](#h.3gcc5l7ie49a)

[UserSex](#h.gr65upgfeb2a)

[Wallet](#h.wi0xepflzb3q)

[WalletPayment](#h.57v2bnqox10d)

[WalletSumForBonus](#h.iurfpjfly0xq)

[WalletType](#h.57fccwwaesl)

[Приложения](#h.gzth3b8rti9t)

[Разметка чеков](#h.q93alg6y1xfu)

[Примеры использования](#h.gpmuawaaf321)

[// привязка картой, начисление бонусов](#h.okc16erjqzxw)

[// оплата по карте будет выглядеть как-то так](#h.omfwlp6s6i46)

[а отмена совсем просто](#h.cs3mysvlkj3k)

# 

# 

# Введение

Данное Api предназначено для интеграции системы iikoNet с кассовыми системами (POS).

Взаимодействие осуществляется по протоколу https. Версия протокола http: 1.1. Сериализация - json.

Для всех вызовов, кроме проверки соединения и авторизации, используются batch-запросы ([RFC2045](http://tools.ietf.org/html/rfc2045), [RFC2046](http://tools.ietf.org/html/rfc2046)).

Пока что имеется возможность вызова всех методов без упаковки в batch. Но этот режим использовать не рекомендуется. В скором времени он может быть отключен.

Все примеры запросов в данном описании приведены для batch-запросов.

Поддерживаются следующие режимы сжатия трафика:

* Без сжатия
* Сжатие gzip
* Сжатие deflate (рекомендуемый режим)

Базовый адрес для всех http-запросов (кроме проверки соединения и авторизации): **https://iiko.net:9010/api/v2/front**

# 

# 

# Форматы

Формат для дат: **yyyy-MM-ddTHH:mm:ss.FFFFFFFK**([ISO8601](http://www.iso.org/iso/catalogue_detail?csnumber=40874))

Формат для Guid: **{00000000-0000-0000-0000-000000000000}**

Десятичный разделитель: **точка**

Сериализация перечислений (enum): по названию

# 

# 

# Методы

Ниже приведены описания методов, предоставляемых Api. Общие правила:

* Если в описании не указана секция “Параметры запроса” - запрос вызывается без параметров
* Если в описании не указана секция “Возвращаемое значение” - запрос не имеет возвращаемого значения (void)
* Успешным считается запрос, ответ на который содержит HttpStatusCode == 200 (OK)

## Пакетные вызовы (батчинг)

## 

## Проверка соединения

Может использоваться для проверки доступности сервиса Api. Работает по защищенному каналу, но без авторизации.

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/ |
| Http-метод | GET |

Пример запроса:

|  |
| --- |
| GET http://192.168.0.64:9010/api/ HTTP/1.1  Accept: application/json  Host: 192.168.0.64:9010  Connection: Keep-Alive |

Пример ответа:

|  |
| --- |
| HTTP/1.1 204 No Content  Content-Length: 0  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 15:09:45 GMT |

## 

## 

## 

## Авторизация

Метод используется для создания клиентской сессии на сервере iikoNet и получения ее идентификатора. Этот идентификатор должен использоваться в http-заголовке Session для всех методов, которые требуют авторизации.

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/session/ |
| Http-метод | GET |

Параметры запроса:

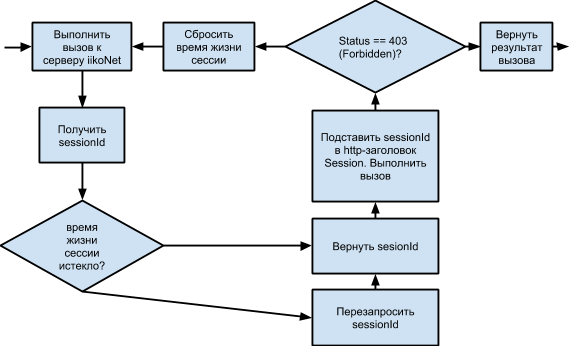
|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| userId | string | Логин точки продаж (настраивается в iikoBiz) |
| userSecret | string | Пароль точки продаж (настраивается в iikoBiz) |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| string | Уникальный ключ клиентской сессии для авторизованного подключения. Этот ключ должен передаваться в http-заголовке ‘Session’ всех остальных методов, кроме проверки связи |

Клиентская сессия имеет время жизни (по умолчанию 15 минут). При перезагрузке сервисов iikoNet клиентская сессия может быть сброшена. Таким образом, клиентская сторона должна поддерживать перезапрос идентификатора сессии.

Рекомендуется такой сценарий:



Пример запроса:

|  |
| --- |
| GET http://192.168.0.64:9010/api/session?userId=3685&userSecret=3685 HTTP/1.1  Accept: application/json  Host: 192.168.0.64:9010 |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 25  Content-Type: application/json; charset=utf-8  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 15:09:45 GMT  "N22rTZalMHs5VDe5yFqi9A2” |

## 

## 

## Настройки

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/settings |
| Http-метод | GET |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| [FrontSettings](#h.sulywle73a2d) | Параметры для кассы, определяемые на сервере iikoNet |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 141  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/settings HTTP/1.1  Host: 192.168.0.64:9009  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 347  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Thu, 05 Jun 2014 11:03:56 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"promoCodeLifeTime":"00:00:00","suggestReplenishForChange":false,"minDenominationForChange":0,"maxPaymentPercent":0.0,"requiredChequeSum":0.0,"minBalanceForPayment":0.0,"allowPaymentWithoutPhote":false}  --batch-1-- |

## 

## Пользователи

### Поиск пользователя

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| identifier | string | Значение идентификатора |
| searchScope | [UserSearchScope](#h.3gcc5l7ie49a) | Возможные типы идентификатора (область поиска) |
| batchId | int | Маркер результата поиска. На него могут ссылаться другие методы внутри батча. В текущей версии всегда должен быть равен 0. |

Возвращаемое значение:

|  |  |
| --- | --- |
| Тип | Назначение |
| [UserSearchResult](#h.xu5u715hy5ae) | Результат поиска |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 208  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%2c+barcode%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 211  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Thu, 05 Jun 2014 17:14:29 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"c77ae7b3-ecd4-11e3-8fc5-3085a99c8b42"}  --batch-1-- |

### Получить информацию о пользователе

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/data |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| [UserData](#h.jfi5atjg1aq9) | Информация о пользователе |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 414  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user/data HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 433  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 10:55:05 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"f878be66-ed68-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"fullName":"Петров Василий Андреевич","sex":"male","hasPhoto":false}  --batch-1-- |

### Получить фотографию пользователя

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/photo |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| byte[] | Фотография пользователя. Массив кодируется в base64 |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 415  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user/photo HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 1028  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 11:36:06 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"b7538db4-ed6e-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  ""  --batch-1-- |

### Получить Url фотографии пользователя

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/photoUrl |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| string | Url по которому доступна фотография пользователя |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 418  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user/photoUrl HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 375  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 11:40:14 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"28c638cf-ed6f-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  "http://iiko.net/images/test.png"  --batch-1-- |

### Привязать внешний идентификатор к пользователю

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/bind |
| Http-метод | PUT |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| externalIdType | [ExternalIdType](#h.m8u90idzy5q1) | Тип идентификатора |
| externalId | string | Значение идентификатора |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 467  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  PUT /api/v2/front/user/bind?externalIdType=%22phone%22&externalId=%2b79169059999 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 303  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 11:43:39 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"c6d1be08-ed6f-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1-- |

## 

## Заказы

### Создать или обновить заказ

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order |
| Http-метод | PUT |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| order | [Order](#h.cgbuodb2rqva) | Заказ |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| Guid | Идентификатор заказа, выданный сервером iikoNet. Если в исходном заказе Id задан - вернется то же самое значение. Если не задан - сервер iikoNet сгенерирует новый идентификатор. Используется в других вызовах, где требуется указать ссылку на заказ. |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 1028  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  PUT /api/v2/front/order HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"id":"a72f90ea-4e8d-4bc3-bb46-2e27a8a87547","fullSum":100.0,"items":[{"id":"8686e0e3-067c-4052-93cb-75b73ed40e8a","productCode":"11145","productName":"Чай с лимоном","amount":1.0,"sum":25.0,"sumAfterDiscount":25.0,"fixedPrice":25.0,"printTime":"2014-06-06T16:54:38.0942279+04:00","productCategory":"Напитки"},{"id":"98aadf62-2946-40de-b001-fda98bf17ac2","productCode":"22348","productName":"Пирожное с кремом","amount":1.0,"sum":75.0,"sumAfterDiscount":75.0,"fixedPrice":75.0,"printTime":"2014-06-06T16:54:38.0942279+04:00","productCategory":"Выпечка"}],"restaurantSectionName":"Веранда","tableNumber":0,"number":34,"guestCount":1,"cashierName":"Петр Смирнов","openTime":"2014-06-06T16:54:38.095228+04:00","prechequeTime":"2014-06-06T16:54:58.095228+04:00","type":"tableServiceOrder"}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 182  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 12:55:08 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  "a72f90ea-4e8d-4bc3-bb46-2e27a8a87547"  --batch-1-- |

### Привязать заказ к пользователю

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/bind |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |
| userRef | [UserRefParam](https://docs.google.com/document/d/1TknFuaaOugz81Mwk7bJn5dvcRjcvmYugujG4OeDcaZg/edit#heading=h.rd83gwatp10) | Ссылка на пользователя |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 1478  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  PUT /api/v2/front/order HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"id":"a14386a8-d8bf-4116-8691-d93a01242430","fullSum":100.0,"items":[{"id":"7a1c72c5-a785-4e3b-8848-abe296375ab1","productCode":"11145","productName":"Чай с лимоном","amount":1.0,"sum":25.0,"sumAfterDiscount":25.0,"fixedPrice":25.0,"printTime":"2014-06-06T16:57:46.8701036+04:00","productCategory":"Напитки"},{"id":"437cab8e-6021-402a-9e01-b5f87b2797f6","productCode":"22348","productName":"Пирожное с кремом","amount":1.0,"sum":75.0,"sumAfterDiscount":75.0,"fixedPrice":75.0,"printTime":"2014-06-06T16:57:46.8701036+04:00","productCategory":"Выпечка"}],"restaurantSectionName":"Веранда","tableNumber":0,"number":34,"guestCount":1,"cashierName":"Петр Смирнов","openTime":"2014-06-06T16:57:46.8701036+04:00","prechequeTime":"2014-06-06T16:58:06.8701036+04:00","type":"tableServiceOrder"}  --batch-1  Content-Type: application/http; msgtype=request  POST /api/v2/front/order/bind?orderId=a14386a8-d8bf-4116-8691-d93a01242430 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 472  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 12:59:15 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"30c88092-ed7a-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  "a14386a8-d8bf-4116-8691-d93a01242430"  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1-- |

### 

### Отвязать заказ от пользователя

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/unbind |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 191  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  POST /api/v2/front/order/unbind?orderId=ec96a659-57e6-4b5a-9fce-5497a01d2d5b HTTP/1.1  Host: 192.168.0.64:9009  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 105  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 13:38:12 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1-- |

## 

## 

## Кошельки

Id кошелька по умолчанию (бонусы iikoNet): **{00000000-0000-0000-0000-000000000000}**

### Получить список кошельков пользователя

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/wallets |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| userRef | [UserRefParam](https://docs.google.com/document/d/1TknFuaaOugz81Mwk7bJn5dvcRjcvmYugujG4OeDcaZg/edit#heading=h.rd83gwatp10) | Ссылка на пользователя |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| [Wallet](#h.a8ppzpaa3un3)[] | Массив, в котором перечислены кошельки пользователя. |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 417  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user/wallets HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 669  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 13:46:39 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"ef8ff6df-ed80-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  [{"id":"00000000-0000-0000-0000-000000000000","type":{"id":"00000000-0000-0000-0000-000000000000","name":"р. iiko.NET","isActive":true},"balance":100.0},{"id":"ef8ff6e6-ed80-11e3-8fc5-3085a99c8b42","type":{"id":"de1125fd-d157-42e3-a65c-21937511b238","name":"Тестовый кошелек","isActive":false},"balance":200.0}]  --batch-1-- |

### Получить баланс кошелька

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/wallet/balance |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| walletId | Guid | Идентификатор кошелька |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| decimal | Баланс кошелька пользователя. |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 470  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user/wallet/balance?walletId=00000000-0000-0000-0000-000000000000 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 347  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 13:50:52 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"3b29dbd8-ed81-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  100.0  --batch-1-- |

### Пополнить баланс кошелька

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/user/wallet/refill |
| Http-метод | PUT |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| walletId | Guid | Идентификатор кошелька |
| sum | decimal | Сумма пополнения |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-2"  Host: 192.168.0.64:9009  Content-Length: 477  --batch-2  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-2  Content-Type: application/http; msgtype=request  PUT /api/v2/front/user/wallet/refill?walletId=00000000-0000-0000-0000-000000000000&sum=100 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-2-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 303  Content-Type: multipart/mixed; boundary="batch-2"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 13:58:21 GMT  --batch-2  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"88997fa5-ed82-11e3-8fc5-3085a99c8b42"}  --batch-2  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-2-- |

## 

## Платежи

### Оплата заказа

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/pay |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| sum | decimal | Сумма к оплате |
| paymentCode | string | Код типа оплаты |
| orderId | Guid | Идентификатор заказа |
| externalTransactionId | Guid | Идентификатор транзакции оплаты |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-2"  Host: 192.168.0.64:9009  Content-Length: 208  --batch-2  Content-Type: application/http; msgtype=request  POST /api/v2/front/order/pay?sum=70&paymentCode=&orderId=8951f86d-6450-4bb6-b3a5-49f962b5d527&externalTransactionId=8951f86d-6450-4bb6-b3a5-49f962b5d527 HTTP/1.1  Host: 192.168.0.64:9009  --batch-2-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 105  Content-Type: multipart/mixed; boundary="batch-2"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 14:13:57 GMT  --batch-2  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-2-- |

### 

### Отмена оплаты

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/cancelPay |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| externalId | decimal | Сумма к оплате |
| paymentCode | string | Код типа оплаты |
| orderId | Guid | Идентификатор заказа |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Получить оплаты iikoNet для заказа

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/payments |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| [WalletPayment](#h.w9ad3bv3ocbg)[] | Оплаты, пришедшие с сервера iikoNet |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 192  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/order/payments?orderId=90062f1c-7834-451a-9fb3-45461c4463f7 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 212  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 14:22:36 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  [{"walletTypeId":"95d5ec98-6003-42b4-9462-4088c74ea127","sum":10.0}]  --batch-1-- |

### 

### Получить максимальную сумму, которую можно оплатить заданным типом оплаты

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/maxPaymentSum |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| paymentCode | string | Код типа оплаты |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| decimal | Максимальная сумма, которую можно оплатить заданным типом оплаты |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 228  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/order/maxPaymentSum?paymentCode=iikoNetPaymentCode&orderId=00000000-0000-0000-0000-000000000000 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 149  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 14:19:47 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  100.0  --batch-1-- |

### Проверить возможность оплаты заданной суммы в заказе заданным типом оплаты

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/assertMaxPaymentSum |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| paymentCode | string | Код типа оплаты |
| fullSum | decimal | Полная сумма заказа, за вычетом скидок |
| sumToPay | decimal | Сумма, которую собираются оплатить |
| orderId | Guid | Идентификатор заказа |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Session: N22rTZalMHs5VDe5yFqi9A2  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 486  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  POST /api/v2/front/order/assertMaxPaymentSum?paymentCode=iikoNetPaymentCode&fullSum=100&sumToPay=50 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 303  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 14:15:45 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"048662c4-ed85-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1-- |

### 

### Возврат заказа (отмена всех оплат и бонусов)

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/abort |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

## 

## Программы лояльности

### Получить текст пречека в виде разметки

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/billChequeFooter |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| string | Текст для печати на пречеке в виде [разметки](#h.1ux1l49uk23m) |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### Получить текст чека в виде разметки

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/chequeFooter |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| string | Текст для печати на фискальном чеке в виде [разметки](#h.1ux1l49uk23m) |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Получить текст пречека в виде текста

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/billChequeFooterFormatted |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| tapeWidth | int | Ширина ленты печатающего устройства в символах |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| string | Текст для печати на пречеке, отформатированный по ширине ленты печатающего устройства и разбитый по строкам |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Получить текст чека в виде текста

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/chequeFooterFormatted |
| Http-метод | GET |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| tapeWidth | int | Ширина ленты печатающего устройства в символах |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| string | Текст для печати на фискальном чеке, отформатированный по ширине ленты печатающего устройства и разбитый по строкам |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Привязать промокод к заказу

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/bindPromocode |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| promocode | string | Промокод |
| orderId | Guid | Идентификатор заказа |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Предварительный расчет бонусов (для добавления в заказ скидок и бесплатных блюд)

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/loyaltyResult |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| [LoyaltyResult](#h.p4sydro921la) | Результат работы программ лояльности |

### 

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Задать примененные к заказу программы лояльности (скидки)

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/appliedDiscounts |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| Название | Тип | Назначение |
| orderId | Guid | Идентификатор заказа |
| discounts | AppliedDiscount[] | Набор примененных скидок |

Возвращаемое значение:

|  |  |
| --- | --- |
| Тип | Назначение |
| [LoyaltyResult](#h.p4sydro921la) | Результат работы программ лояльности |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### Задать оплаты, с помощью которых закрыт заказ

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/bindPayments |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |
| payments | [Payment](#h.tiskgwq4wa29)[] | Оплаты |

Пример запроса:

|  |
| --- |
|  |

Пример ответа:

|  |
| --- |
|  |

### 

### Задать сумму, на которую могут быть начислены бонусы по каждому из кошельков

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/sumForBonuses |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |
| sums | [WalletSumForBonus](#h.iurfpjfly0xq)[] | Суммы по кошелькам, на которые будут начислены бонусы |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 1329  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  PUT /api/v2/front/order/sumForBonuses?orderId=8f6e19d0-373c-4eee-80bb-263b6dbfaf04 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  [{"walletTypeId":"00000000-0000-0000-0000-000000000000","sum":50.0}]  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 274  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 14:26:24 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1-- |

### 

### Закрыть заказ и произвести окончательный расчет бонусов

|  |  |
| --- | --- |
| Uri | https://iiko.net:9010/api/v2/front/order/close |
| Http-метод | POST |

Параметры запроса:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| orderId | Guid | Идентификатор заказа |
| userRef | [UserRefParam](#h.rd83gwatp10) | Ссылка на пользователя |

Возвращаемое значение:

|  |  |
| --- | --- |
| **Тип** | **Назначение** |
| [LoyaltyResult](#h.p4sydro921la) | Результат работы программ лояльности |

Пример запроса:

|  |
| --- |
| POST http://192.168.0.64:9009/api/v2/front/batch HTTP/1.1  Accept: application/json  Content-Type: multipart/mixed; boundary="batch-1"  Host: 192.168.0.64:9009  Content-Length: 2042  Connection: Keep-Alive  --batch-1  Content-Type: application/http; msgtype=request  GET /api/v2/front/user?identifier=1234556789&searchScope=%22cardTrack%22&batchId=0 HTTP/1.1  Host: 192.168.0.64:9009  --batch-1  Content-Type: application/http; msgtype=request  PUT /api/v2/front/order HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"id":"98a37fce-a808-46c5-a0b9-7914bd0652a2","fullSum":100.0,"items":[{"id":"4ef56813-e4b8-43a6-a8fc-2ebc1272b76f","productCode":"11145","productName":"Чай с лимоном","amount":1.0,"sum":25.0,"sumAfterDiscount":25.0,"fixedPrice":25.0,"printTime":"2014-06-06T18:28:40.1254028+04:00","productCategory":"Напитки"},{"id":"6a128e53-db11-46ce-9d1b-b9f2365a7ae6","productCode":"22348","productName":"Пирожное с кремом","amount":1.0,"sum":75.0,"sumAfterDiscount":75.0,"fixedPrice":75.0,"printTime":"2014-06-06T18:28:40.1254028+04:00","productCategory":"Выпечка"}],"restaurantSectionName":"Веранда","tableNumber":0,"number":34,"guestCount":1,"cashierName":"Петр Смирнов","openTime":"2014-06-06T18:28:40.1264019+04:00","prechequeTime":"2014-06-06T18:29:00.1264019+04:00","type":"tableServiceOrder"}  --batch-1  Content-Type: application/http; msgtype=request  POST /api/v2/front/order/bind?orderId=98a37fce-a808-46c5-a0b9-7914bd0652a2 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1  Content-Type: application/http; msgtype=request  PUT /api/v2/front/order/sumForBonuses?orderId=98a37fce-a808-46c5-a0b9-7914bd0652a2 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  [{"walletTypeId":"00000000-0000-0000-0000-000000000000","sum":50.0}]  --batch-1  Content-Type: application/http; msgtype=request  POST /api/v2/front/order/close?orderId=98a37fce-a808-46c5-a0b9-7914bd0652a2 HTTP/1.1  Host: 192.168.0.64:9009  Content-Type: application/json; charset=utf-8  {"paramType":"batchIdRef","batchIndex":0}  --batch-1-- |

Пример ответа:

|  |
| --- |
| HTTP/1.1 200 OK  Content-Length: 734  Content-Type: multipart/mixed; boundary="batch-1"  Server: Microsoft-HTTPAPI/2.0  Date: Fri, 06 Jun 2014 14:29:10 GMT  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"foundBy":"cardTrack","id":"e3906dc6-ed86-11e3-8fc5-3085a99c8b42"}  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  "98a37fce-a808-46c5-a0b9-7914bd0652a2"  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 204 No Content  --batch-1  Content-Type: application/http; msgtype=response  HTTP/1.1 200 OK  Content-Type: application/json; charset=utf-8  {"totalDiscount":0.0,"totalBonus":3.50}  --batch-1-- |

# 

# 

# 

# Типы данных

### CardMaskInfo

Описатель маски карты iikoNet. Применяется в offline-сценариях, когда нужно определить принадлежность карты к iikoNet при отсутствии связи.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| type | [CardMaskInfoType](#h.au69iwjp3g01) | тип маски |
| mask | string | значение маски |

### 

### CardMaskInfoType

Перечисление возможных типов масок карт:

* **iikoNetDefaultFormat** - формат маски по умолчанию
* **plastek** - маска карт Пластек
* **cardNumberMask** - маска для номера карты
* **cardBarcodeMask** - маска для штрихкода

### 

### ExternalIdType

Перечисление возможных типов идентификатора пользователя. Возможные значения:

* **notFound** - пользователь не найден
* **external** - внешний пользователь. Пользователь принадлежит внешнему процессингу и в базе iikoNet не хранится (например, Связной).
* **phone** - телефон
* **cardNumber** - номер карты
* **cardTrack** - трек карты
* **id** - внутренний идентификатор пользователя iikoNet
* **barcode** - штрихкод
* **qrCode** - Qr-код

### 

### FrontSettings

Настройки кассовой системы, которые задаются на сервере iikoNet.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| promoCodeLifetime | TimeSpan | Время жизни промокода |
| suggestReplenishForChange | bool | Предлагать ли пополнить кошелек на сдачу |
| minDenominationForChange | int | Минимальная купюра. Остаток от сдачи меньше этой купюры будет предложен для пополнения кошелька. |
| cardMasks | [CardMaskInfo](#h.57v2bnqox10d)[] | Маски карт |
| maxPaymentPercent | decimal | Максимально допустимый процент оплаты бонусами |
| requiredChequeSum | decimal | Минимальная сумма чека для оплаты бонусами |
| minBalanceForPayment | decimal | Минимально допустимый баланс кошелька для оплаты боунсами |
| allowPaymentWithoutPhoto | bool | Разрещать ли оплату для гостя, у которого не задана фотография |

### 

### LoyaltyOperation

Действие (операция) программы лояльности, применямое к одному элементу заказа

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| code | string | Код типа операции |
| productCode | string | Код продукта (в кассовой системе, артикул) |
| productName | string | Название блюда |
| discountSum | decimal | Сумма скидки |
| comment | string | Комментарий к блюду от программы лояльности (происхождение скидки и т.п.) |

### LoyaltyProgramResult

Результат работы применяемой к заказу программы лояльности

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| programId | Guid | Идентификатор программы лояльности |
| name | string | Наименование программы |
| operations | [LoyaltyOperation](#h.7xj8ksdeklwr)[] | Операции, применяемые к отдельным элементам заказа |

### LoyaltyResult

Результат работы программ лояльности для заказа.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| totalBonus | decimal | Оценка Бонус для заказа (сумма бонусов которые будут начислены если заказ будет закрыт на типы оплаты на которые начисляется бонус) |
| programResults | [LoyaltyProgramResult](#h.v0bvx84lwu98)[] | Маркетинговые акции и прочие программы |

### Order

Заказ.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| id | Guid | Идентификатор заказа |
| fullSum | decimal | Полная сумма заказа за вычетом скидки |
| items | [OrderItem](#h.t49plg224bc1)[] | Элементы (позиции) заказа |
| restarauntSectionName | string | Название группы (секции ресторана), в которой открыт заказ |
| tableNumber | int | Номер стола, на котором открыт заказ |
| number | int | Порядковый номер заказа в кассовой смене |
| fiscalChequeNumber | int? | Номер фискального чека |
| guestsCount | int | Количество гостей |
| waiterName | string | Официант, открывший заказ |
| cashierName | string | Кассир, выполнивший оплату |
| openTime | DateTimeOffset | Время открытия заказа (локальное, с часовым поясом) |
| prechequeTime | DateTimeOffset? | Время пробития пречека |
| closeTime | DateTimeOffset? | Время закрытия заказа |
| type | [OrderType](#h.l2naw7oabrk0) | Тип заказа (TableService, FastFood и т.п.) |
| conception | string | Концепция |

### OrderItem

Элемент (позиция) заказа.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| id | Guid | Идентификатор заказа |
| productCode | string | Код продукта (артикул) |
| productName | string | Название продукта |
| amount | decimal | Количество единиц блюда |
| sum | decimal | Стоимость блюда |
| sumAfterDiscount | decimal | Стоимость блюда после применения скидок |
| paymentIsNotAvailableForCodes | string | [к обсуждению] Коды типов оплат, которыми блюдо не может быть оплачено |
| comment | string | Комментарий к блюду |
| fixedPrice | decimal | Цена за единицу блюда на момент печати |
| printTime | DateTimeOffset? | Время печати блюда |
| productCategory | string | Категория продукта |

### OrderType

Перечисление возможных типов заказа:

* **unknown** - тип неизвестен
* **fastFoodOrder** - заказ FastFood
* **tableServiceOrder** - заказ TableService
* **deliveryOrder** - доставки
* **selfServiceOrder** - самообслуживание
* **document** - документооборот
* **petroleumOrder** - заказ с автозаправочной станции
* **banquetOrder** - банкет

### ParamType

Перечисление возможных типов параметра-ссылки пакетного запроса:

* **BatchIdRef** - ссылка на результат запроса, выполненного в рамках текущего пакета
* **ExternalId** - идентификатор внешнего объекта. Такие объекты (например, пользователи внеших платежных систем) не хранятся в сервере IikoNet и, следовательно, не могут быть опеределены другими вызовами в рамках пакета

### UserData

Информация о пользователе

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| fullName | string | Имя, фамилия, отчество |
| sex | [UserSex](#h.gr65upgfeb2a) | Пол |
| birthday | DateTime? | День рождения |
| hasPhoto | bool | Указана ли фотография |

### UserRefParam

Параметра для задания пользователя в запросе. Используется для адресации к пользователям iikoNet (хранятся в базе, могут быть определены или закешированы другими вызовами батча) и внешним пользователям (в базе iikoNet не хранятся).

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| paramType | [ParamType](#h.ofxjt8862rcs) | Тип параметра (от значения зависит, какие поля являются значимыми - BatchIndex или ExternalIdType + ExternalId) |
| batchIndex | int | Индекс внутри батча |
| externalIdType | [ExternalIdType](#h.uli1wanljgvw) | Тип внешнего идентификатора |
| externalId | string | Значение внешнего идентификатора |

### UserSearchResult

Результат поиска пользователя

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| foundBy | [ExternalIdType](#h.uli1wanljgvw) | По какому типу идентификатора был найден |
| id | Guid | Идентификатор пользрователя в системе iikoNet. Для внешних пользователей значения не имеет и равен Guid.Empty |

### UserSearchScope

Mаска для поиска пользователя. Определяет типы идентификаторов, по которым будет осуществлен поиск. Может быть скомбинирована из флагов:

* **phone** - искать в телефонных номерах
* **cardNumber** - искать в номерах карт
* **cardTrack -** искать в треках карт
* **id** - искать по идентификатору
* **barcode** - искать по штрихкоду
* **qrCode** - искать по Qr-коду

### UserSex

Перечисление для указания пола:

* **notSpecified** - пол не задан
* **male** - мужской
* **female** - женский

### Wallet

Кошелек пользователя.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| id | Guid | Идентификатор кошелька |
| type | [WalletType](#h.57fccwwaesl) | Тип кошелька |
| balance | decimal | Баланс кошелька |

### WalletPayment

Оплата из кошелька.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| walletTypeId | Guid | Идентификатор типа кошелька |
| sum | decimal | Сумма оплаты |

### 

### WalletSumForBonus

Сумма для начисления бонусных баллов на кошелек.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| walletTypeId | Guid | Идентификатор типа кошелька |
| sum | decimal | Сумма, на которую будут начислены бонусы |

### 

### WalletType

Тип кошелька.

Поля:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Назначение** |
| id | Guid | Идентификатор типа кошелька |
| name | string | Название кошелька |
| isActive | bool | Активен ли кошелек |

# 

# Приложения

## Разметка чеков

## Примеры использования

### // привязка картой, начисление бонусов

private void button2\_Click(object sender, EventArgs e)

{

tbTrace.Clear();

var sw = Stopwatch.StartNew();

apiClient.Ping();

var batch = apiClient.NewBatch();

// наполнение батча вызовами

// 1. Поиск пользователя по карте

var userResult = batch.SearchUser("7777", UserSearchScope.CardNumber);

// 2. Обновление заказа

var orderItem1 = new OrderItem { Amount = 1, ProductName = "Чай с лимоном", FixedPrice = 25, Id = Guid.NewGuid(), Sum = 25 };

var orderItem2 = new OrderItem { Amount = 1, ProductName = "Булочка", FixedPrice = 75, Id = Guid.NewGuid(), Sum = 75 };

var orderInfo = new Order

{

Id = Guid.NewGuid(),

CashierName = "Петя",

FullSum = 100,

Items = new[] { orderItem1, orderItem2}

};

batch.CreateOrUpdateOrder(orderInfo);

// 3. Привязка пользователя к заказу

batch.BindOrderToUser(orderInfo.Id, userResult);

// 4. Получение пречека

var billChequeResult = batch.GetBillChequeFooterFormatted(40, orderInfo.Id);

// 5. Суммы для начисления бонусов

var sums = new[] { new WalletSumForBonus { WalletTypeId = Guid.Empty, Sum = 50 } };

batch.SetSumForBonuses(orderInfo.Id, sums);

batch.CloseOrder(orderInfo.Id, userResult);

// выполнение батча (все вызовы уходят в одном http-запросе)

batch.Execute();

// получение результатов

var lines = billChequeResult.Value;

sw.Stop();

Trace.TraceInformation(string.Join(Environment.NewLine, lines));

tbTrace.Text = @"Elapsed: " + sw.ElapsedMilliseconds + @" ms." + Environment.NewLine + Environment.NewLine + tbTrace.Text;

}

### // оплата по карте будет выглядеть как-то так

const string IikoNetPaymentTypeCode = "iikoNet PaymentTypeCode";

const string YandexPaymentTypeCode = "Yandex money";

tbTrace.Clear();

var sw = Stopwatch.StartNew();

// 1. Заказ готов к оплате.

var orderInfo = new Order

{

Id = Guid.NewGuid(), CashierName = "Петя", FullSum = 25

};

var orderItem = new OrderItem

{

Amount = 1, ProductName = "Чай с лимоном", FixedPrice = 25, Id = Guid.NewGuid(), Sum = 25

};

orderInfo.Items = new[] {orderItem};

// 2. Гость передает свою карту и говорит, что хочет оплачивать бонусами.

const string cardTrack = "7777";

// 3. Кассир прокатывает карту на кассе.

var batch = apiClient.NewBatch();

var userSearchResult = batch.SearchUser(cardTrack, UserSearchScope.CardNumber);

batch.CreateOrUpdateOrder(orderInfo);

batch.BindOrderToUser(orderInfo.Id, userSearchResult);

var userInfoResult = batch.GetUserData(userSearchResult);

var userPhotoResult = batch.GetUserPhoto(userSearchResult);

// максимально возможная сумма для оплаты бонусами

var sumToPayBonusesResult = batch.GetMaxPaymentSum(IikoNetPaymentTypeCode, orderInfo.Id);

// максимально возможная сумма для оплаты Yandex Money

var sumToPayYandexResult = batch.GetMaxPaymentSum(YandexPaymentTypeCode, orderInfo.Id);

var chequeFooterResult = batch.GetChequeFooterFormatted(30, orderInfo.Id);

// 4. Касса обращается на сервер системы лояльности, отправляет на него номер карты, состав заказа, суммы, которые можно оплатить заданными типами оплат,

// уже примененные оплаты (если есть).

try

{

batch.Execute();

}

catch (HttpException httpException)

{

if (httpException.GetHttpCode() == (int)HttpStatusCode.NotFound)

{

Trace.TraceInformation("User not found: " + httpException.Message);

return;

}

Trace.TraceInformation("Processing error: " + httpException.Message);

return;

}

// 5. Система лояльности отправляет на кассу сумму, доступную к оплате и данные гостя (ФИО и фото, например), данные для печати не чеке.

Trace.TraceInformation(userInfoResult.Value.ToString());

Trace.TraceInformation("Photo size: " + userPhotoResult.Value.Length);

Trace.TraceInformation("Max sum to pay bonueses: " + sumToPayBonusesResult.Value);

Trace.TraceInformation("Max sum to pay yandex: " + sumToPayYandexResult.Value);

Trace.TraceInformation("Cheque footer: " + chequeFooterResult.Value);

// 6. На данный момент у нас есть «привязанный» заказ (заказ у которого есть гость в системе лояльности).

// 7. Касса запрашивает доступные для заказа программы лояльности (скидки, маркетинговые акции)

batch = apiClient.NewBatch();

var loyaltyResult = batch.GetLoyaltyResult(orderInfo.Id);

try

{

batch.Execute();

}

catch (HttpException httpException)

{

Trace.TraceInformation("Processing error: " + httpException.Message);

return;

}

// 8. Касса применяет скидки к заказу

AppliedDiscount[] appliedDiscounts = null;

if (checkinResult.LoyaltyResult != null)

{

var appliedDiscounts = checkinResult.LoyaltyResult.ProgramResults

.SelectMany(program => program.Operations, (program, operation) => new { program, operation })

.Select(item => new AppliedDiscount

{

ProgramId = item.program.ProgramId,

OperationCode = item.operation.Code,

ProductCode = item.operation.ProductCode,

ProductName = item.operation.ProductName,

Sum = item.operation.DiscountSum

})

.ToArray();

batch = apiClient.NewBatch();

batch.SetAppliedDiscounts(appliedDiscounts);

try

{

batch.Execute();

}

catch (HttpException httpException)

{

Trace.TraceInformation("Processing error: " + httpException.Message);

return;

}

}

// 9. Кассир удостоверяется что перед ним действительно тот гость, кому принадлежит карта (по фото) и соглашается с оплатой.

// 10. Касса добавляет в заказ «Оплата бонусами iiko.net».

// 11. Кассир добавляет другие типы оплат и закрывает заказ

batch = apiClient.NewBatch();

if (IsOrderChanged(orderInfo))

batch.CreateOrUpdateOrder(orderInfo);

batch.Pay(10, IikoNetPaymentTypeCode, orderInfo.Id, null);

batch.Pay(50, YandexPaymentTypeCode, orderInfo.Id, null);

try

{

batch.Execute();

}

catch (HttpException httpException)

{

Trace.TraceInformation("Processing error: " + httpException.Message);

return;

}

// 12. Касса отправляет системе лояльности данные о заказе и типах оплат, на которые возможно начисление бонусов.

// 13. Система лояльности списывает оплаченные бонусы и начисляет положенные бонусы за данный заказ

// (при этом бонусы на бонусы не начисляются, а начисляются они только на те типы оплат, на которые начисление бонусов разрешено).

batch = apiClient.NewBatch();

if (IsOrderChanged(orderInfo))

batch.CreateOrUpdateOrder(orderInfo);

var iikoNetPayment = new Payment {PaymentTypeCode = IikoNetPaymentTypeCode, Sum = 10};

var yandexPayment = new Payment { PaymentTypeCode = YandexPaymentTypeCode, Sum = 50 };

var cashPayment = new Payment { PaymentTypeCode = "CashPaymentTypeCode", Sum = 15 };

batch.BindPaymentsToOrder(orderInfo.Id, new[] { iikoNetPayment, yandexPayment, cashPayment });

userSearchResult = batch.SearchUser(cardTrack, UserSearchScope.CardNumber);

loyaltyResult = batch.CloseOrder(orderInfo.Id, userSearchResult);

try

{

batch.Execute();

}

catch (HttpException httpException)

{

Trace.TraceInformation("Processing error: " + httpException.Message);

return;

}

// 14. Касса печатает чек, с информацией полученной от системы лояльности.

Trace.TraceInformation(loyaltyResult.Value.ToString());

sw.Stop();

tbTrace.Text = @"Elapsed: " + sw.ElapsedMilliseconds + @" ms." + Environment.NewLine + Environment.NewLine + tbTrace.Text;

### а отмена совсем просто

batch = client.NewBatch();

batch.Abort(orderDto.Id);

batch.Execute();

## 

## 

## 