**Интеграция систем Gmoji и iiko Front**

Редакция от 5 декабря 2017

**Используемые термины:**

**TBD**: to be discussed. Элементы системы, которые необходимо обсудить и согласовать для успешной разработки.

Для работы Gmoji решение Front должно обеспечивать следующий функционал:

* Возможность чтения штрих-кодов и qr-кодов формата Gmoji
* Возможность оплаты заказа полностью или частично купонами Gmoji

**Системные требования:**

Iiko версии 4.3.8 или выше.

Интернет соединение

.net framework 4.0 или выше

Windows XP или выше (возможны сложности с шифрованием запросов, но плагин будет стартовать в win xp)

**[TBD]** Лицензия на плагин в системе iiko. На время разработки будет использоваться SaaS лицензия за 500р в месяц (предоставляется 9 коннектов, т.е. плагин можно ставить на 9 касс в заведении)

**Общие принципы**

1. **Формат купонов Gmoji**

Купоны Gmoji представляют из себя штрих-коды формата Code128.

Формат купона фиксированный, длина n символов.

**TBD**: описать формат штрих кода и qr-кода, фиксированная ли длина, есть ли стартовые и конечные символы, какая часть используется для отправки на бэкенд. Предоставить примеры штрихкодов и qr-кодов (можно сгенерировать картинки, сканеры читают с экрана).

 Примеры номеров: «**150M-701J-701H-6945»**

1. **Передача данных**

Передача данных между POS-терминалом и Gmoji осуществляется посредством HTTP-запросов к web api Gmoji, формат данных: JSON

Каждому **заведению выдаются уникальный логин и пароль для авторизации**. Так же **выдается store\_id для гашения джипонов**. При начале работы с системой Gmoji система iikoFront должна произвести авторизацию на сервере Gmoji для получения токена, которым сопровождаются все дальнейшие запросы.

 Для сокращения времени расчета заказа, на любой запрос к системе должен быть установлен тайм-аут 20 секунд. По истечении этого времени, если не был получен ответ от сервера, запрос считается несостоявшимся и пользователь iikoFront должен получить сообщение об ошибке.

**TBD:** токен живет 1 день, описать процедуру обновления токена (отдельный запрос или просто повторная авторизация). **Я провел запрос на получение токена** и там есть поля "expires\_in" и "refresh\_token", но метод для обновления токена не был предоставлен. Expires\_in – время жизни токена в секундах(видно из значения).

Пример авторизации:

*Headers:*

**X-Api-Factory-Application-Id**: 587640c995ed3c0c59b14600 (не меняется)

**Authorization**: Basic NTg3NjQwYzk5NWVkM2MwYzU5YjE0NjAwOjg4YmYxY2Q3MGQ=

**Content-Type**: application/json

*Body:*

@param - {string} - username - логин

@param - {string} - password – пароль

Пример:

**POST - http://gmoji02.simbirsoft1.com/api/script/login\_subcontragent**

@return - data - информация из профиля представителя

@return - token - объект с токеном авторизации (access\_token)

1. **Сопоставление номенклатуры**

Концепция Gmoji подразумевает, что ресторан создает купоны во внешней административной панели Gmoji (без участия iikoRms и iikoBackOffice). Для корректного взаимодействия с iikoFront, в информации о джипоне будет передаваться артикул блюда, сопоставленный с данным джипоном. Автоматическая выгрузка номенклатуры не подразумевается.В дальнейшем, при расчете купоном Gmoji, данный артикул может использоваться для подстановки в заказ соответствующего блюда из системы iikoFront.

**TBD**: данный функционал потребует изучения, т.к. есть некоторые сложности с блокировками заказов на редактирование, поэтому было предложено просто добавлять стоимость джипона в оплату (или нескольких джипонов), а добавление товара вынести в зону ответственности кассира. Так же следует учесть такие моменты, как модификаторы к блюдам и процессинг стоп-листов (блюдо закончилось на кухне).

1. **Работа с джипонами**

При работе с джипонами используются следующие методы:

* Поиск джипона по полному коду или его фрагменту
* Гашение джипона по его id, pin и идентификатору

Пример запроса для поиска джипона:

*Headers:*

**X-Api-Factory-Application-Id:** 587640c995ed3c0c59b14600 (не меняется)

**Authorization**: Bearer + access\_token (Пример: Bearer e10f156221b6474d70f5fc3710542d19d75d3b18)

**Content-Type**: application/json

*Body:*

@param - {string} - code - уникальный код(или часть кода) джипона (Пример: 150M-701J-701H-6945)

@param - {string} - store\_id - идентификатор торговой точки

Пример:

**GET - http://gmoji02.simbirsoft1.com/api/script/find\_gpon\_for\_cash?code=150B**

@return - массив джипонов

**TBD**: Структуру получил, но там на данный момент отсутствует артикул. При добавлении артикула просьба уведомить.

Пример запроса для гашения джипона:

*Headers:*

**X-Api-Factory-Application-Id:** 587640c995ed3c0c59b14600 (не меняется)

**Authorization:** Bearer + access\_token (Пример: Bearer e10f156221b6474d70f5fc3710542d19d75d3b18)

**Content-Type:** application/json

*Body:*

@param - {string} - certificate\_id - id выбранного джипона (Пример: 59d3400c7e38e128d2a53981)

@param - {string} - pin - пин выбранного джипона (Пример: 7647)

@param - {string} - store\_id - id магазина (Пример: 598aaa7e47217f28ba69b652)

Пример:

**POST - http://gmoji02.simbirsoft1.com/api/script/cash\_certificate\_wrap**

@return - погашенный джипон

**TBD**: Согласовать способ получения store\_id для ресторана.

**TBD**: Вопрос по полю PIN джипона. Может ли он начинаться с 0? Если да, то нужно будет рисовать свое окно для ввода пин кода с кнопки (в айке есть хороший контрол, который можно вызвать через апи для ввода числа, но он для ввода чисел, поэтому пин формата 0123 там ввести не представляется возможным)

Может возникнуть необходимость добавить метод для гашения массива джипонов в рамках единой транзакции

1. **Процедура расчета**
* Кассир открывает заказ
* Пользователь уведомляет кассира о желании погасить джипон
* Кассир добавляет товар, который соответствует джипону.
* Кассир переходит на кассовый экран, выбирает безналичный расчет для оплаты, тип оплаты Gmoji
* iikoFront запрашивает сканирование джипона с таймаутом 20 секунд на чтение
* После сканирования запрашивается информация о джипоне(запрос на сервер Gmoji). В случае невалидности джипона должно быть отображено сообщение об ошибке и расчет должен быть прерван
* Сумма оплаты автоматически заменяется на полученную от сервера. В случае попытки изменить сумму кассир не сможет закрыть заказ.
* Нажав на кнопку оплатить происходит гашение джипона в данном заказе (запрос на сервер Gmoji). В случае успеха, заказ закрывается.
1. **Возвраты по заказам**

Возвраты заказов, оплаченные полностью или частично джипонами запрещены в системе iikoFront(при попытке возврата такого чека, кассир получит сообщение о невозможности возврата джипона).

**TBD**: реализация возвратов возможна, если будут предоставлены соответствующие методы API.