Specification

of integration module
for POS platforms

Техническое задание

на модуль интеграции для
POS-систем

Версия документа 1.19

Версия API 20130701

Москва, 2019

Содержание

Содержание 2
Лист согласований. 4

1. Введение **7**

Общие принципы **8**
Примечание 8
Заголовки HTTP 8
Ответы API на запросы 11
Таблица 1. Коды состояния протокола HTTP 11

Авторизация **12**
Примечание 12
Объект Token 12
Объект ApplicationToken 13
Внимание 13

Спецификация интерфейса **14**
Ресурс программ лояльности (/loyalties/) 14
Программа лояльности (/loyalties/ 15
Ресурс пользователей (/users/) 17
Таблица 8. Параметры метода GET ресурса пользователей. 18
Таблица 9. Параметры метода POST ресурса пользователей. 19
Пользователь (/users/{user_id}) 20
Купоны пользователя (/users/{user_id}/ coupons/) 24
Таблица 11. Параметры метода GET ресурса купонов пользователя. 24
Купон пользователя (/users/{user_id}/ coupons/{id}) 26
Покупки пользователя (/users/{user_id}/ purchases/) 27
Таблица 12. Параметры метода GET ресурса покупок пользователя. 28
Таблица 14. Поля покупки. 31
Таблица 15. Поля номенклатурной позиции покупки. 33
Таблица 16. Параметры метода POST ресурса покупок пользователя. 33
Таблица 17. Атрибуты номенклатурной позиции покупки. 35
Покупка пользователя (/users/{user_id}/ purchases/{id}) 40
Номенклатура покупки (/users/{user_id}/purchases/ 41
Номенклатурная позиция покупки (/users/{user_id}/ purchases/{id}/items/{item_id}) 42

Приём-передаточные испытания **43**
Общие требования к приёмке работы 43
Приёмно-передаточные испытания перечень 43

Алгоритм предоставления скидки по акции **47**
Купон на скидку 47
Таблица Приложение 2.1. Условные обозначения 47
Таблица Приложение 2.2. Ограничение суммы, точная (параметр A). 47
Таблица Приложение 2.4. Ограничение суммы, максимальная (параметр C). 47
Купон на сумму 48

Таблица Приложение 2.6. Условные обозначения 48
Таблица Приложение 2.7. Без ограничений (параметры не заданы). 48
Таблица Приложение 2.9. Ограничение суммы, минимальная (параметр B). 48

Протокол тестирования интеграционного решения 49

Лист согласований.

№	Должность, ФИО	Дата	Замечания	Подпись
1	**Ген. директор Михайлишин А.Ю.**	**15.05.2019**		

Лист регистрации изменений

Издание 1.0 01.10.2012 Бадяев А.М.

Начальная редакция документа.

Издание 1.1 05.07.2013 Бадяев А.М.

1. В Раздел 2.3, «Ответы *API* на запросы» [8] добавлена ошибка HTTP с кодом 409.

2. Во всём документе уточнено использование терминов *токен авторизации* и *токен авторизации приложения*.

3. В Раздел 4.4, «Ресурс программ лояльности (`/loyalties/`)» [12] отмечено, что может быть ситуация, когда *Участники* не обладает программой лояльности - или нулевая программа лояльности.

4. В Таблица 11, «Параметры метода **GET** ресурса купонов пользователя.» [28] добавлены параметры: *`curr_iso_code`*, *`curr_iso_name`*, *`items`*.

5. Изменилось определение параметра *`items`* (товары покупки): добавлено поле *`gtin`*, изменились имена полей.

6. Глоссарий перенесен в конец документа.

Издание 1.2 08.07.2013 Бадяев А.М.

1. Длина параметра с номером карты пользователя увеличена до `25`.

2. Обновлены примеры ответов для методов ресурсов купонов в Раздел 4.12, «Купоны поль- зователя (`/users/{user_id}/coupons/`)» [28].

3. Обновлены примеры ответов для методов ресурсов купонов в Раздел 4.14, «Покупки поль- зователя (`/users/{user_id}/purchases/`)» [31]. Исправлены примеры допустимых значений параметра *`items`*.

Издание 1.3 15.07.2013 Бадяев А.М.

1. В описании метода **POST** ресурса покупок пользователя добавлено объяснение обработки параметра *`commit`*.

Издание 1.4 18.07.2013 Бадяев А.М.

Добавлено описание числовых параметров в параграфе: Числовые параметры [7] и даны ссылки на него в тексте документа.

Издание 1.5 13.08.2013 Бадяев А.М.

Добавлено описание параметров даты и времени в параграфе: Параметры даты и времени [7] и даны ссылки на него в тексте документа.

Издание 1.6 02.09.2013 Бадяев А.М.

В Таблица 14, «Поля покупки.» [36] добавлено описание параметра *`url`*.

Издание 1.7 05.09.2013 Бадяев А.М.

В Таблица 16, «Параметры метода **POST** ресурса покупок пользователя.» [38] добавлено описание параметра *`override`*.

Издание 1.8 09.10.2013 Бадяев А.М.

В Таблица 8, «Параметры метода **GET** ресурса пользователей.» [21] добавлено описание параметра *`foreigncard`*.

Издание 1.8.1 11.10.2013 Бадяев А.М.

В Таблица 11, «Параметры метода **GET** ресурса купонов пользователя.» [28] добавлено описание параметров *redeem_auto* и *sum_total*. Выполнена общая редакция документа без из- менений в определении программного интерфейса.

Издание 1.9 15.02.2014 Бадяев А.М.

В Раздел 2, «Общие принципы» [5] добавлено предупреждение о проблеме ограни- чения длины URL [http://ru.wikipedia.org/wiki/URL] в Microsoft [http://ru.wikipedia.org/wiki/Microsoft]™ Internet Explorer [http://ru.wikipedia.org/wiki/Internet_Explorer]™.

Издание 1.9.1 15.02.2014 Бадяев А.М.

В Таблица 16, «Параметры метода **POST** ресурса покупок пользователя.» [38] Исправлено описание типа параметра *item_N_{attr}*.

Издание 1.10 17.02.2014 Бадяев А.М.

В Таблица 8, «Параметры метода **GET** ресурса пользователей.» [21] добавлено описание параметра *auto*.

Издание 1.11 17.03.2016 Бадяев А.М.

В Раздел 4.10, «Ресурс пользователей (`/users/`)» [21] добавлено описание нового метода POST [23].

Издание 1.12 25.03.2016 Бадяев А.М.

Добавлены разделы Раздел 4.8, «Сторонние карты (`/cards/`)» [17] и Раздел 4.9, «Сторон- няя карта пользователя (`/cards/{card_id}`)» [20].

Издание 1.13 08.07.2016 Бадяев А.М.

Внесены дополнения в разделы Раздел 4.8, «Сторонние карты (`/cards/`)» [17] и Раздел 4.9, «Сторонняя карта пользователя (`/cards/{card_id}`)» [20].

Издание 1.14 27.12.2016 Бадяев А.М.

В метод `GET` ресурса пользователей в разделе Раздел 4.10, «Ресурс пользователей (`/users/`)» [21] добавлен параметр *email*.

Издание 1.15 19.12.2017 Бадяев А.М.

В описании ресурса программы лояльности добавлен раздел Параметры бонусной программы лояльности [14].

Издание 1.16 05.12.2018 Бадяев А.М.

Добавлен раздел *Алгоритм предоставления скидки по акции* [51].

Издание 1.17 06.12.2018 Бадяев А.М.

Добавлены разделы *Приём-передаточные испытания* [47] и *Протокол тестирования интеграционного решения* [53].

Издание 1.18 06.12.2018 Бадяев А.М.

Дано подробное описание параметров ответа на запрос проведения покупки с примером одновременного начисления и списания бонусных баллов.

Издание 1.19 15.05.2019 Бадяев А.М.

Добавлено описание новых функций управления справочником номенклатуры товаров в Раздел 4.6, «Справочник номенклатуры (`/products/`)» [16] и Раздел 4.7, «Элемент справочника номенклатуры (`/products/{product_id}`)» [17].

1. Введение

Этот документ описывает открытый программный интерфейс (*API*) к Системе версии 20130701. Архитектура программного интерфейса построена в стиле REST [http://ru.wikipedia.org/wiki/REST] и содержит следующие корневые ресурсы:

1. **tokens.** Интерфейс к подсистеме управления *токенами*. Клиентское приложение имеет возможность запросить новый *токен*, проверить текущее состояние полученного *токе-на*, и отозвать (отключить) полученный *токен*. Этот ресурс подробно описан в Раздел 4.2, «Ресурс авторизации (`/tokens/`)» [11].

2. **loyalties.** Программы лояльности *участников Системы*.

3. **users.** Этот ресурс является корневым для всех функций *Системы*, за исключением управления авторизацией, и подробно рассмотрен в Раздел 4.10, «Ресурс пользователей (`/users/`)» [21].

4. **logon.** Служебный ресурс для авторизации в *Системе* при работе с *API* через интернет-обозреватель (например, Internet Explorer).

2. Общие принципы

Система предоставляет для доступа к своим функциям *API*, представляющее собой Web-интерфейс. Все сообщения между пользователями и *системой* выполняются через протокол HTTPS [http://ru.wikipedia.org/wiki/HTTPS]. Запросы клиентского приложения, в соответствии со стилем архитектуры REST [http://ru.wikipedia.org/wiki/REST], могут содержать методы:

- **GET.** Чтение данных ресурса (например, получить полную анкету пользователя).
- **PUT.** Запись данных в ресурс (например, изменить значение предоставляемой пользователю скидки).
- **POST.** Создание нового элемента данных ресурса (например, добавить продажу пользователя).
- **DELETE.** Удаление элемента данных ресурса (например, возврат продажи пользователя).

Примечание

При использования для доступа к функциям API обозревателя (или его программ- ного компонента) Microsoft [http://ru.wikipedia.org/wiki/Microsoft]™ Internet Explorer [http://ru.wikipedia.org/wiki/Internet_Explorer]™ может возникнуть проблема ограничения в `2083` символов на длину URL [http://ru.wikipedia.org/wiki/URL], описан- ная в статье [http://support.microsoft.com/kb/208427/ru]. Для избежания этой проблемы следует отказаться от использования программных компонент Microsoft [http://ru.wikipedia.org/wiki/Microsoft]™ Internet Explorer [http://ru.wikipedia.org/wiki/Internet_Explorer]™ при реализации интеграционного решения доступа к функциям API в пользу какого-либо открытого программного компонента (например, cURL [http://curl.haxx.se/]), который свободен от указанного в статье ограничения.

2.1. Заголовки HTTP

Ниже перечислены заголовки HTTP, которые указываются в запросах к *API*. В случае, когда клиентское приложение не имеет возможности управлять заголовками отправляемых запросов, допустимо указывать соответствующую информацию в *`GET`*-параметрах и (в отдельных случа- ях) в параметрах cookie [http://ru.wikipedia.org/wiki/HTTP_cookie]. Полное описание всех HTTP заголовков содержится в документации консорциума W3C [http://www.w3.org/] `RFC 2616 - 14 Header Field Definitions [http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html]`.

DM-Authorization. Значением этого параметра является *токен авторизации приложения* (интеграционного решения). Тип авторизации при этом должен быть *`dmapptoken`*. Пример заголовка:

```
DM-Authorization : dmapptoken 550e8400-e29b-41d4-a716-446655440000
```

Если приложение не может сформировать параметр заголовка, то допускается передача *токена приложения* в **GET**-параметре запроса с именем *`_dmapptoken`*.

1. **Authorization.** Значением этого параметра является *токен авторизации* кассового терминала (POS-системы). Тип авторизации при этом должен быть *`dmtoken`*. Пример

заголовка:

```
Authorization : dmtoken 550e8400-e29b-41d4-a716-446655440000
```

Если приложение не может сформировать параметр заголовка, то допускается передача *токена* в **GET**-параметре запроса с именем *_dmtoken*.

2. **Accept.** Этот заголовок описывает формат ответа от *API*, ожидаемый клиентским приложением. Если параметр не указан в запросе, то по умолчанию ответ отправляется в фор- мате JSON [http://ru.wikipedia.org/wiki/JSON]. Рекомендуется указывать этот параметр в заголовках всех запросов к *API*. Пример заголовка, запрашивающий ответ в формате XML:

```
Accept : application/xml
```

Если указан неподдерживаемый формат, то будет возвращен код ошибки 406 Not Acceptable и подробное описание в формате JSON [http://ru.wikipedia.org/wiki/JSON]. Пример ответа Системы на запрос с неверным параметром заголовка **Accept**:

```
{"available_types": ["application/json", "text/html",
   "application/xml"], "detail": "Could not satisfy the client's Accept
   header"}
```

Если параметр не указан в заголовке запроса, то по умолчанию *Система* возвращает ответ (если он не пустой) в формате JSON [http://ru.wikipedia.org/wiki/JSON]. Если приложение не может сформировать параметр заголовка, то допускается передача языка в **GET**-параметре запроса с именем *format*.

3. **Accept-Language.** Значением этого параметра является язык (локализация), на котором приложение предпочитает получать ответы от *Системы*. Пример заголовка (для русского языка, Российского варианта):

```
Accept-Language : ru-RU
```

Если параметр не указан в заголовке запроса, то по умолчанию *Система* возвращает ответ в русской «локали» ru. Если приложение не может сформировать параметр заголовка, то допускается передача языка в **GET**-параметре запроса с именем *lang*.

4. **User-Agent.** Строка, полностью идентифицирующая платформу и клиентское приложение, которое отправило запрос. Принятый в компании формат этой строки совместим с общепринятым в мире форматом http://en.wikipedia.org/wiki/User_agent, однако есть ряд специфических особенностей. Рассмотрим пример допустимой идентифицирую- щей строки:

```
User-Agent: 1C/8.2 (Trade/1.2.310) Rubin-Kem/1.0.13
```

Ниже перечислены все поля строки и дано их подробное описание:

- Клиентское приложение.

    1. **1C/8.2.** Программной платформой клиентского приложения служит продукт 1С: Предприятие [http://v8.1c.ru/overview/Platform.htm] 8.2.
    2. **Trade.** Наименование клиентского приложения - в данном случае это 1С:Управление торговлей [http://v8.1c.ru/trade/].
    3. **1.2.310.** Версия клиентского приложения.

- Интеграционное решение, взаимодействующее с *API*.

    1. **Rubin-Kem.** Наименование интеграционного решения, обеспечивающего

взаимодействие клиентского приложения с *Системой* через *API*.

2. **`1.0.13.`** Версия интеграционного решения.

Этот параметр является обязательным. Допускается передача параметра через **GET**-параметр *`_useragent`*.

2.2. Запросы к *API*

Формат сообщения с запросом к *API* зависит от того, какой метод HTML (**GET**, **PUT**, **POST**, **DELETE**) вызывается. Для вызова одной из перечисленных выше команд ресурсов *API* необходимо выполнить соответствующий запрос HTTP. То есть для вызова команды **GET** необходимо отправить HTTP запрос типа **GET**, и так далее. При использовании команды **GET** параметры запроса передаются в URL запроса, при остальных методах параметры кодируются в теле сообщения. Для передачи параметров команды в теле HTTP запроса могут быть использованы следующие значения заголовка **Content-Type**:

- **`application/x-www-form-urlencoded.`** Параметры метода *API* кодируются в теле HTML запроса по правилам кодирования параметров в URL.

Числовые параметры. Все числовые параметры передаются в виде строки десятичных цифр. Если для параметра указано количество знаков после запятой, то дробная часть числа всегда (независимо от региональных настроек) отделяется от целой части символом «точка» (`.`), после которой должны следовать десятичные цифры в указанном количестве знаков после запятой. Например, если тип параметра описан как число с тремя знаками после запятой, то допустимым значением параметра будет являться строка вида: «`156.000`», при этом следует указать все цифры дробной части, даже если они нулевые.

Параметры даты и времени. Все параметры даты и времени передаются строкой вида «`YYYY-MM-DD HH:MM:SS {+|-}hhmm`», где:

- `YYYY` - год, например `2013`;
- `MM` - номер месяца в году, начиная с `1`, например `07`;
- `DD` - номер дня в месяце, начиная с `1`, например `03`;
- `HH` - номер часа в сутках, начиная с `0`, например `09`;
- `MM` - номер минуты в часе, начиная с `0`, например `05`;
- `SS` - номер секунды в минуте, начиная с `0`, например `01`;

- `{+|-}hhmm` - часы и секунды смещения локального времени от времени по Гринвичу, например +0700.

Примеры допустимого значения параметра даты и времени: «`2013-07-03 09:05:01 +0700`», «`2013-12-31 23:59:59 -0330`».

2.3. Ответы *API* на запросы

Поскольку транспортом сообщений служит протокол HTTP, то код успешности выполнения запроса возвращается в статусе ответа на HTTP запрос (полный перечень кодов статуса ответа содержится в RFC 2616 [http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html]). Вся возвращаемая в ответ на запрос информация содержится в теле ответа в виде объекта, закодированного в ожидаемом клиентским приложением формате, переданном в заголовке HTTP **Accept** (3 [6]). При получении клиентским приложением определенных кодов статуса ответа, тело ответа может быть пустым (204 No Content) или содержать данные, не соответствующие запрошенному формату (например, если код ошибки был возвращен серверным *ПО*, до получения запроса *API*). Ниже приведён перечень кодов и имен статусов ответа API на запрос.

Таблица 1. Коды состояния протокола HTTP

200 OK	Запрос выполнен успешно.
201 Created	В результате успешного выполнения запроса был создан новый ресурс.
204 No Content	Запрос успешно выполнен, но в ответе были переданы только заголовки без тела сообщения.
400 Bad Request	Сервер обнаружил в запросе клиента синтаксическую ошибку.
401 Unauthorized	Для доступа к запрашиваемому ресурсу требуется аутентификация.
403 Forbidden	Клиент ограничен в доступе к запрашиваемому ресурсу.
404 Not Found	Запрашиваемая информация не найдена.
405 Method Not Allowed	Указанный клиентом метод нельзя применить к текущему ресурсу.
406 Not Acceptable	запрошенный URL не может удовлетворить переданным в заголовке характеристикам.
407 Proxy Authentication Required	Для доступа к запрашиваемому ресурсу требуется аутентификация на *proxy*-сервере.
408 Request Timeout	Время ожидания сервером передачи от клиента истекло.
409 Conflict	Запрос не может быть выполнен из-за конфликтного обращения к ресурсу.
5XX Internal Server Error	Внутренняя ошибка сервера.

Вообще, не определено содержимое тела ответа с любым статусом, не документированный настоящим руководством.

3. Авторизация

Подсистема авторизации предназначена для обеспечения такого доступа к программным интерфейсам программных продуктов компании, при котором выполняются только авторизованные запросы только авторизованных в системе (совокупность программного обеспечения, доступного через внешний программный интерфейс) пользователей (в том числе *ПО* кассовых терминалов - POS [http://marketch.ru/marketing_dictionary/marketing_terms_p/point_of_sale/]).

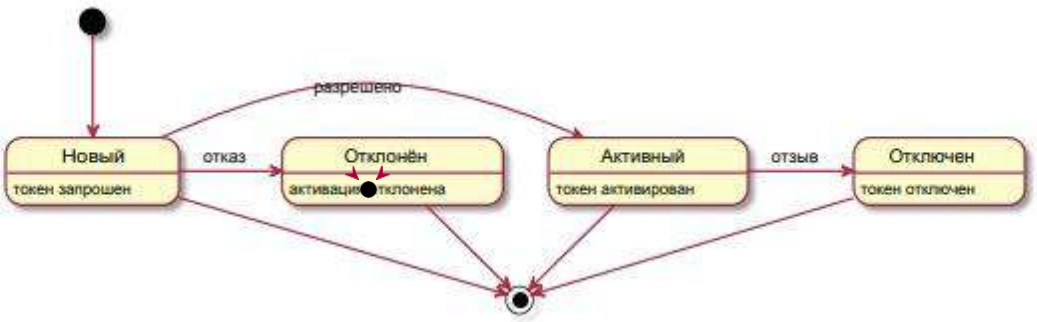
Примечание

Для определения принадлежности запроса определённому пользователю (кассовому терминалу) используется значение *токена*. Для определения *участника Системы*, которому принадлежит POS-терминал, с которого был отправлен запрос, используется атрибут *токена* `merchant_shop`.

Каждый *токен авторизации* уникален в системе и однозначно идентифицирует каждого пользователя или кассовый терминал. *ПО* кассовых терминалов получает *токены* в результате процесса активации (см. Активация [10]).

3.1. Объект `Token`

Объект типа `Token` в течение своей жизни в системе обладает следующими состояниями:



Также объект типа `Token` должен содержать следующие атрибуты:

- идентификатор (ссылку) на *участника* системы - владельца терминала (*DIN* или GLN);
- идентификатор *ПО* кассового терминала (POS-системы) - до `25` символов;
- описание терминала (касса) - строка до `100` символов, не обязательная для заполнения;
- дата и время запроса активации *токена*;
- дата и время активации *токена*;
- дата и время автоматического отзыва *токена* (по умолчанию `20` лет с момента активации);
- дата и время отзыва (отключения) *токена*;
- дата и время удаления *токена*.

В качестве значения *токена* используется UUID [http://ru.wikipedia.org/wiki/UUID], созданный системой и переданный пользователю (приложению терминала) в результате выполнения

процесса активации. В запросе к API системы *токен авторизации* передается в виде строкового параметра длиной не более 40 символов, и состоящего из символов множества:

```
[0-9,a-f,'-']
```

(шестнадцатеричные цифры и дефис).

Активация. В результате процесса активации все POS [http://marketch.ru/marketing_dictionary/marketing_terms_p/point_of_sale/]-терминалы *участника* системы получают уникальный идентификатор в системе (*токен*). Запрос на активацию принимается только от авторизованного (именем и паролем) пользователя системы, обладающего соответствующими полномочиями (правами доступа). В запросе на активацию передается вся необходимая для однозначной идентификации терминала информация. Результатом успешно выполненного запроса на активацию является объект системы `ActivationRequest`. Пользователь системы, обладающий правом активации терминала, может просматривать и управлять объектами `ActivationRequest`, в том числе выполнять операции разрешения активации, отклонения (с указанием причины) активации. Для всех объектов `ActivationRequest`, для которых была выполнена операция разрешения активации, система автоматически создает уникальный *токен*.

Использование. Значение объекта типа `Token` должно передаваться в каждом запросе к ресурсам с корнем `users` (см. Раздел 4.10, «Ресурс пользователей (`/users/`)» [21]) или `coupons`. При этом возможна передача *токена* в заголовке HTTP-запроса (параметр *`Authorization`*, тип авторизации `dmtoken`) или в **GET**-параметре *`token`* вызова *API*. При- мер параметра в заголовке HTTP-запроса:

```
Authorization : dmtoken 550e8400e29b41d4a716446655440000
```

Пример URL [http://ru.wikipedia.org/wiki/URL] запроса с *токеном* в **GET**-параметре:

```
https://dm.api/20130701/users/?_dmtoken=550e8400e29b41d4a716446655440000
```

При выполнении тестовых вызовов *API* через Web-интерфейс допускается передача *токена* в виде значения cookie [http://ru.wikipedia.org/wiki/HTTP_cookie] с именем *`_dmtoken`*.

3.2. Объект `ApplicationToken`

Токен авторизации приложения `ApplicationToken` представляет собой ключ авторизации приложения (интеграционного решения) в *Системе*. Этот ключ создаётся и выдаётся разработ- чикам приложений, использующим *API*, сотрудниками компании.

Внимание

В отличие от объекта `Token`, *токен авторизации приложения* должен быть указан в каждом запросе к *API*.

Значение *токена приложения* должно передаваться в заголовке HTTP-запроса (параметр *`DM-Authorization`*, тип авторизации *`dmapptoken`*). Пример параметра в заголовке HTTP-запроса:

```
DM-Authorization : dmapptoken 550e8400-e29b-41d4-a716-446655440000
```

Пример URL [http://ru.wikipedia.org/wiki/URL] запроса с *токеном приложения* в **GET**-параметре:

```
https://dm.api/20130701/users/?_dmapptoken=550e8400e29b41d4a716446655440000
```

При выполнении тестовых вызовов *API* через Web-интерфейс допускается передача *токе- на приложения* в виде значения cookie [http://ru.wikipedia.org/wiki/HTTP_cookie] с именем *`_dmapptoken`*.

4. Спецификация интерфейса

Программный интерфейс *Системы* представляет собой набор методов ресурсов, доступ к которым осуществляется через свой URL, который далее в документации указан относительно корневого URL сайта *API*, который публикуется компанией при выпуске каждой версии *Системы*. Этот раздел содержит подробное описание всех ресурсов *Системы*, доступные через описываемый настоящим документом *API*.

4.1. *Токен авторизации* (`/tokens/{token}`)

Этот ресурс представляет собой конкретный *токен авторизации*, полученный от *Системы*. Пример ссылки на ресурс *токена*:

```
https://dm.api/20130701/tokens/550e8400e29b41d4a716446655440000
```

Для ресурса доступны следующие методы:

- **GET.** Возвращает статус активации *токена*. В случае если *токен авторизации* активирован будет возвращен статус HTTP 200 OK, в ином случае 404 Not Found. Пример ответа *API* в формате JSON при успешном выполнении метода на ресурсе активного *токена*:

  ```
  {"active": true}
  ```

  Пример ответа в формате XML:

  ```
  <?xml version="1.0" encoding="utf-8"?>
  <root><active>True</active></root>
  ```

- **DELETE.** Отзыв (отключение) полученного ранее *токена*. После успешного выполнения возвращается статус HTTP 204 No Content, в ином случае 404 Not Found.

4.2. Ресурс программ лояльности (`/loyalties/`)

Программы лояльности *участников Системы* представлены корневым ресурсом *API* `loyalties/`. Пример ссылки на ресурс:

```
https://dm.api/20130701/loyalties/
```

Ресурс поддерживает следующие методы:

- **GET.** Возвращает список программ лояльности *участника Системы* (который определяется по *токену* авторизации). Пример ответа от *API* в формате JSON при успешном выполнении метода (идентификатор *участника* `2001`):

```
[{"url": "http://dm.api/20130701/loyalties/2001",
  "currency_code": 643,
  "currency_name": "RUB",
  "thresholds": [[0, 1], [10000, 3], [50000, 5]],
  "type": "amount"}]
```

Приведенный выше ответ содержит список из одной программы лояльности (скидка предоставляется покупателю на основании накопленной суммы покупок - именно это означает значение `amount` параметра *`type`*) *участника Системы* с идентификатором `2001`. Параметр *`url`* ответа содержит ссылку на конкретную программу лояльности.

Пример того же ответа в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <list-item>
    <url>http://dm.api/20130701/loyalties/2001</url>
    <currency_code>643</currency_code>
```

```
      <currency_name>RUB</currency_name>
      <thresholds>
        <list-item>
          <list-item>0</list-item>
          <list-item>1</list-item>
        </list-item>
        <list-item>
          <list-item>10000</list-item>
          <list-item>3</list-item>
        </list-item>
        <list-item>
          <list-item>50000</list-item>
          <list-item>5</list-item>
        </list-item>
      </thresholds>
      <type>amount</type>
    </list-item>
  </root>
```

4.3. Программа лояльности (`/loyalties/{merchant_id}`)

Программа лояльности *участника* Системы представлена ресурсом, ссылка на который имеет вид:

```
http://dm.api/20130701/loyalties/2001
```

Методы. Ресурс предоставляет следующие методы:

- **GET.** Получение полной информации о ПЛ (Программы Лояльности) *участника Системы*. Программа лояльности может отсутствовать, а также содержать нулевые значения порогов и скидок.

  Пример ответа на запрос в формате JSON при успешном выполнении метода:

  ```
  {"currency_code": 643,
   "currency_name": "RUB",
   "thresholds": [[0, 1], [10000, 3], [50000, 5]],
   "type": "amount"}
  ```

  Приведенный выше ответ содержит данные программы лояльности (скидка предоставляется покупателю на основании накопительной суммы покупок - именно это означает значение `amount` параметра `type`) *участника Системы* с идентификатором `2001`. Параметр `url` ответа содержит ссылку на конкретную программу лояльности. Параметр `thresholds` пред- ставляет собой список граничных условий и значений предоставляемой скидки в рамках указанного типа программы лояльности. Условия из примера ответа означают, что покупателю сразу (при первой покупке) предоставляется скидка `1%` от суммы покупки. Если общая сумма всех покупок покупателя превысила `10000` рублей, то покупателю предоставляется скидка `3%` от суммы покупки. И наконец, если общая сумма всех покупок пользователя у этого *участника Системы* превысила `50000` рублей, то покупатель получает скидку `5%` от суммы покупки.

  Пример того же ответа в формате XML:

  ```
  <?xml version="1.0" encoding="utf-8"?>
  <root>
    <currency_code>643</currency_code>
    <currency_name>RUB</currency_name>
    <thresholds>
      <list-item>
        <list-item>0</list-item>
        <list-item>1</list-item>
      </list-item>
  ```

```
      <list-item>
        <list-item>10000</list-item>
        <list-item>3</list-item>
      </list-item>
      <list-item>
        <list-item>50000</list-item>
        <list-item>5</list-item>
      </list-item>
    </thresholds>
    <type>amount</type>
  </root>
```

Параметры ответа. Тип ПЛ (Программы Лояльности) *участника Системы* содержится в параметре *type*. В зависимости от типа и настроек ПЛ метод GET возвращает различные данные. Ниже перечислены типы ПЛ и описаны возвращаемые методом GET параметры:

- **amount.** Накопительная скидка в процентах от суммы покупки с порогами по сумме покупок в магазинах *Участника*.

currency_code	Целое положительное число.	Цифровой код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
currency_name	Строка символов длиной 3.	Алфавитный код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
thresholds	Список, элементами которого являются списки из 2-х положительных целых чисел.	Каждый элемент списка представляет порог изменения скидки. Первый элемент порога представляет собой сумму в основной валюте *Участника*, а второй - скидку *Пользователя* в процентах, которую он получит при накоплении им требуемой суммы покупок.

- **count.** Накопительная скидка в процентах от суммы покупки с порогами по количеству покупок в магазинах *Участника*.

currency_code	Целое положительное число.	Цифровой код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
currency_name	Строка символов длиной 3.	Алфавитный код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
thresholds	Список, элементами которого являются списки из 2-х элементов.	Каждый элемент списка представляет порог изменения скидки. Первый элемент порога представляет собой количество покупок в точках продаж *Участника*, а второй - скидку *Пользователя* в процентах, которую он получит при накоплении им требуемого количе- ства покупок.

- **bonus.** Бонусная программа лояльности с накоплением бонусных баллов за стоимость покупки и с возможностью оплаты части покупки накопленными ранее в магазинах *Участника* бонусными баллами.

currency_code	Целое положительное число.	Цифровой код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
currency_name	Строка символов длиной 3.	Алфавитный код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
min_purchase_amount	Положительное целое число.	Минимальная сумма покупки для начисления бонусных баллов.
amount_to_bonus	*Список* из двух элемен- тов: 1. Положительное целое число. 2. Число с фиксированной точкой (точность до 4-х знаков после точки) в форме строки.	Элементы списка содержат данные: 1. Сумма денежных средств в основной валюте *Участника*. 2. Количество бонусных баллов, начисляемых за указанную в п.1 этого списка сумму денежных средств.
bonus_to_amount	Список из двух элемен- тов: 1. Положительное целое число. 2. Положительное целое число.	Элементы списка содержат данные: 1. Количество бонусных баллов. 2. Сумма скидки в основной валюте *Участника*, которая будет получена *Пользователем* при оплате части покупки бонусными баллами в количестве из п.1 этого списка.
max_purchase_percentage	Положительное целое число в диапазоне от 0 до 100.	Максимальная часть стоимости покупки в процентах, которая может быть оплачена бонусными баллами.
expiration	Положительное целое число.	Количество дней, спустя которые начисленные бонусные баллы сгорят.

- **slave.** Это так называемая «Ведомая программа лояльности», у которой нет параметров, кроме типа.

- **nothing.** Это так называемая «Нулевая программа лояльности».

4.4. Ресурс пользователей (`/users/`)

Ресурс пользователей (базовый URL которого `users/`) является корневым для всех ресурсов *Системы*, за исключением ресурса авторизации, и предназначен для поиска пользователей, зарегистрированных в Системе. Пример ссылки на ресурс пользователей Системы:

```
https://dm.api/20130701/users/
```

Ресурс предоставляет следующие методы:

- **GET.** Поиск пользователя системы по одному или нескольким фильтрам. Обязательно должен быть указан хотя бы один фильтр. Если ни один фильтр не указан, метод возвращает

ошибку HTTP с кодом 400 Bad Request. Параметры (фильтры поиска) метода:

Таблица 8. Параметры метода GET ресурса пользователей.

Параметр	Тип данных	Описание
auto	Строка цифр длиной до 100.	Номер карты (в *Системе* или в сторонней дисконтной системе) покупателя или номер купона, принадлежащего покупателю.
card	Строка цифр длиной до 25.	Номер карты покупателя.
phone	Строка цифр длиной до 15.	Номер телефона покупателя.
email	Строка цифр длиной до 100.	Адрес электронной почты покупателя.
gsrn	Строка цифр длиной до 25.	Номер карты покупателя.
foreigncard	Строка цифр длиной до 100.	Номер карты покупателя в сторонней дисконтной системе.
coupon	Строка цифр длиной до 25.	Номер купона, принадлежащего покупателю.

Если по предоставленным фильтрам найден один или несколько пользователей, то метод возвращает их перечень, который содержит идентификатор пользователя, его *ФИО*, и ссылку (URL) на ресурс конкретного пользователя. Пример ссылки на ресурс пользователя с идентификатором 1003:

```
https://dm.api/20130701/users/1003
```

При вызове метода следует указать только один из возможных фильтров поиска. Метод всегда возвращает статус HTTP 200 OK.

При поиске пользователя по номеру карты возвращается объект, представляющий найденного пользователя, или пустой объект, если пользователь с данным номером карты не найден. Пример ответа *API* в формате JSON при успешном поиске пользователя с идентификатором 1003:

```
[{"purchases": 3,
  "first_name": "Иван Петрович
  Сидоров", "last_name": "",
  "middle_name":
  "", "bonus": 70,
  "purchases_url": "http://dm.api/20130701/users/1003/purchases/",
  "discount": 3,
  "amount": "7000.00",
  "url": "http://dm.api/20130701/users/1003",
  "loyalty_url":
  "http://dm.api/20130701/loyalties/2001",
  "photo_urls": {"100x125":
                 "http://dm.api/img/default/avatar-100x100.png", "50x62":
                 "http://dm.api/img/default/avatar-50x50.png"},
  "coupons_url": "http://dm.api/20130701/users/1003/coupons/",
  "id": 1003,
  "card": "46200111390168093545331l6"}]
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <list-item>
    <purchases>3</purchases>
    <first_name>Иван Петрович Сидоров</first_name>
    <last_name></last_name>
    <middle_name></middle_name>
    <bonus>70</bonus>
    <purchases_url>http://dm.api/20130701/users/1003/purchases/
    </purchases_url>
    <discount>3</discount>
```

```
    <amount>7000.00</amount>
    <url>http://dm.api/20130701/users/1003</url>
    <loyalty_url>http://dm.api/20130701/loyalties/2001</loyalty_url>
    <photo_urls>
      <_100x125>http://dm.api/static/img/default/avatar-100x100.png</_100x100>
      <_50x62>http://dm.api/static/img/default/avatar-50x50.png</_50x50>
    </photo_urls>
    <coupons_url>http://dm.api/20130701/users/1003/coupons/
    </coupons_url>
    <id>1003</id>
    <card>46200111390168093545331l6</card>
  </list-item>
</root>
```

Пример ответа на запрос поиска в случае, когда по предоставленным фильтрам не удалось найти ни одного пользователя:

```
[]
```

- **POST.** Регистрация нового пользователя без создания сторонней карты.

Таблица 9. Параметры метода POST ресурса пользователей.

Параметр	Тип данных	Описание
short_name	Строка длиной до 100.	Краткое наименование нового пользователя.
full_name	Строка длиной до 255.	Полное наименование нового пользователя.
gender	Целое число: 1 или 2.	Пол нового пользователя (1 - мужской, 2 - женский).
phone	Строка цифр длиной до 15.	Номер телефона нового пользователя.
email	Строка цифр длиной до 100.	Адрес электронной почты нового пользователя.
password	Строка: “True” или “False”.	Если истина (“True”), то вернуть пароль нового пользователя.
discount	Целое положительное число.	Скидка нового пользователя.
bonus	Положительное число с точностью до двух знаков после запятой.	Накопления бонусных баллов нового пользователя. Формат числовых параметров см. в Числовые параметры [7].
sum	Положительное число с точностью до двух знаков после запятой.	Сумма покупок нового пользователя. Формат числовых параметров см. в Числовые параметры [7].
num	Целое положительное число.	Количество покупок нового пользователя.

После успешного выполнения запроса возвращается статус HTTP 201 CREATED. Если создать нового пользователя невозможно по причине того, что в *Системе* уже есть пользователь с таким номером телефона или адресом электронной почты, какие указаны в запросе, то будет возвращен статус HTTP 409 CONFLICT.

Пример ответа *API* в формате JSON при успешной регистрации нового пользователя без запроса пароля (параметр *password* не задан или его значение “False”):

```
{"DIN": 1008,
 "ID": "46200111390165623214295l6"}
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <DIN>1008</DIN>
  <ID>46200111390168027138257l9</ID>
</root>
```

Пример ответа *API* в формате JSON при успешной регистрации нового пользователя с запросом пароля (передан параметр `password` в значении “True”):

```
{"DIN": 1008,
 "password": "747702",
 "ID": "46200111390160240140010 37"}
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
<DIN>1008</DIN>
  <password>747702</password>
  <ID>46200111390168027138257 09</ID>
</root>
```

4.5. Пользователь (`/users/{user_id}`)

Ресурс пользователя (покупателя) предоставляет доступ к функциям *Системы*, предназначенным для управления данными, связанными с покупателем. Пример ссылки на ресурс пользователя с идентификатором `1003`:

```
https://dm.api/20130701/users/1003
```

Для этого ресурса доступны методы:

- **GET.** Ответ на вызов этого метода содержит полную и подробную информацию о пользователе, а также ссылки на ресурсы `purchases` (покупки пользователя у *участника Системы*, которому принадлежит *токен авторизации*) и `coupons` (не погашенные купоны пользователя по активным акциям или предложениям). Пример ответа *API* в формате JSON на успешный вызов метода:

```
{"purchases": 3,
 "first_name": "Иван Петрович
 Сидоров", "last_name": "",
 "middle_name":
 "", "bonus": 70,
 "purchases_url": "http://dm.api/20130701/users/1003/purchases/",
 "discount": 3,
 "amount": "7000.00",
 "loyalty_url": "http://dm.api/20130701/loyalties/2001",
 "photo_urls": {"100x125":
                "http://dm.api/img/default/avatar-100x100.png", "50x62":
                "http://dm.api/img/default/avatar-50x50.png"},
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/",
 "id": 1003,
 "card": "8460939940"}
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <purchases>3</purchases>
  <first_name>Иван Петрович Сидоров</first_name>
  <last_name></last_name>
  <middle_name></middle_name>
  <bonus>70</bonus>
  <purchases_url>http://dm.api/20130701/users/1003/purchases/
  </purchases_url>
  <discount>3</discount>
  <amount>7000.00</amount>
  <loyalty_url>http://dm.api/20130701/loyalties/2001</loyalty_url>
  <photo_urls>
    <_100x125>http://dm.api/static/img/default/avatar-100x100.png</_100x100>
    <_50x62>http://dm.api/static/img/default/avatar-50x50.png</_50x50>
  </photo_urls>
  <coupons_url>http://dm.api/20130701/users/1003/coupons/</coupons_url>
  <id>1003</id>
  <card>8460939940</card>
</root>
```

- **PUT.** Установить текущую скидку и/или счётчики накоплений пользователя в магазинах данного *участника* (определяется по *токену* авторизации). При успешном выполнении метод возвращает код HTTP ответа 200 OK и подробную информацию о новой дисконтной (виртуальной) карте покупателя. При вызове метода необходимо указать параметры для установки покупателю новой скидки (новая скидка будет рассчитана по счетчикам накоплений согласно действующей программе лояльности):

Таблица 10. Параметры метода PUT ресурса пользователя.

Параметр	Тип данных	Описание
sum	Неотрицательное число с точностью до 2 знаков после запятой.	Сумма всех покупок, совершенных пользователем у данного *участника* Системы. Формат числовых параметров см. в Числовые параметры [7].
num	Целое неотрицательное число.	Количество покупок, совершенных пользователем у данного *участника* Системы.
bonus	Неотрицательное число с точностью до 2 знаков после запятой.	Количество бонусных баллов пользователя у данного *участника* Системы, которые можно потратить. Формат числовых параметров см. в Числовые параметры [7].
percent	Целое неотрицательное число.	Размер новой скидки в процентах. Операция завершится успешно только в том случае, если этот размер скидки есть в текущей программе лояльности *участника* Системы.

Среди перечисленные выше параметров *percent* являются взаимно исключающим со всеми остальными параметрами. При передаче корректного значения параметра *percent* после успешного выполнения метода будут изменены счётчики накоплений пользователя в соот- ветствии с текущей программой лояльности данного *участника*.

Пример ответа *API* в формате JSON на успешную установку скидки по сумме всех покупок (параметр *sum*) пользователю с идентификатором 1003 (до вызова сумма покупок *amount* была 7000.00):

```
{"purchases": 3,
 "first_name": "Иван Петрович
 Сидоров", "last_name": "",
 "middle_name":
 "", "bonus": 70,
 "purchases_url": "http://dm.api/20130701/users/1003/purchases/",
 "discount": 3,
 "amount": "10000.00",
 "loyalty_url": "http://dm.api/20130701/loyalties/2001",
 "photo_urls": {"100x125":
                "http://dm.api/img/default/avatar-100x100.png", "50x62":
                "http://dm.api/img/default/avatar-50x50.png"},
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/",
 "id": 1003,
 "card": "8460939940"}
```

Тот же самый ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
```

```
  <purchases>3</purchases>
  <first_name>Иван Петрович Сидоров</first_name>
  <last_name></last_name>
  <middle_name></middle_name>
  <bonus>70</bonus>
  <purchases_url>http://dm.api/20130701/users/1003/purchases/
  </purchases_url>
  <discount>3</discount>
  <amount>10000.00</amount>
  <loyalty_url>http://dm.api/20130701/loyalties/2001</loyalty_url>
  <photo_urls>
    <_100x125>http://dm.api/static/img/default/avatar-100x100.png</_100x100>
    <_50x62>http://dm.api/static/img/default/avatar-50x50.png</_50x50>
  </photo_urls>
  <coupons_url>http://dm.api/20130701/users/1003/coupons/</coupons_url>
  <id>1003</id>
  <card>8460939940</card>
</root>
```

Пример ответа *API* в формате JSON на успешную установку скидки по количеству всех покупок (параметр *num*) пользователю с идентификатором 1003 (до вызова количество покупок *purchases* была 3):

```
{"purchases": 200,
 "first_name": "Иван Петрович
 Сидоров", "last_name": "",
 "middle_name":
 "", "bonus": 0,
 "purchases_url": "http://dm.api/20130701/users/1003/purchases/",
 "discount": 5,
 "amount": "0.00",
 "loyalty_url": "http://dm.api/20130701/loyalties/2002",
 "photo_urls": {"100x125":
                "http://dm.api/img/default/avatar-100x100.png", "50x62":
                "http://dm.api/img/default/avatar-50x50.png"},
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/",
 "id": 1003,
 "card": "8460939940"}
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <purchases>200</purchases>
  <first_name>Иван Петрович Сидоров</first_name>
  <last_name></last_name>
  <middle_name></middle_name>
  <bonus>0</bonus>
  <purchases_url>http://dm.api/20130701/users/1003/purchases/
  </purchases_url>
  <discount>5</discount>
  <amount>0.00</amount>
  <loyalty_url>http://dm.api/20130701/loyalties/2002</loyalty_url>
  <photo_urls>
    <_100x125>http://dm.api/static/img/default/avatar-100x100.png</_100x100>
    <_50x62>http://dm.api/static/img/default/avatar-50x50.png</_50x50>
  </photo_urls>
  <coupons_url>http://dm.api/20130701/users/1003/coupons/</coupons_url>
  <id>1003</id>
  <card>8460939940</card>
</root>
```

Пример ответа *API* в формате JSON на успешную установку количества бонусов (параметр *bonus*) пользователю с идентификатором 1003 (до вызова количество бонусов *bonus* было 70):

```
{"purchases": 0,
 "first_name": "Иван Петрович
 Сидоров", "last_name": "",
 "middle_name":
 "", "bonus": 400,
 "purchases_url": "http://dm.api/20130701/users/1003/purchases/",
 "discount": 0,
 "amount": "0.00",
 "loyalty_url": "http://dm.api/20130701/loyalties/2003",
 "photo_urls": {"100x125":
```

```
                "http://dm.api/img/default/avatar-100x100.png", "50x62":
                "http://dm.api/img/default/avatar-50x50.png"},
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/",
 "id": 1003,
 "card": "8460939940"}
```


Тот же самый ответ в формате XML:


```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <purchases>0</purchases>
  <first_name>Иван Петрович Сидоров</first_name>
  <last_name></last_name>
  <middle_name></middle_name>
  <bonus>400</bonus>
  <purchases_url>http://dm.api/20130701/users/1003/purchases/
  </purchases_url>
  <discount>0</discount>
  <amount>0.00</amount>
  <loyalty_url>http://dm.api/20130701/loyalties/2003</loyalty_url>
  <photo_urls>
    <_100x125>http://dm.api/static/img/default/avatar-100x100.png</_100x100>
    <_50x62>http://dm.api/static/img/default/avatar-50x50.png</_50x50>
  </photo_urls>
  <coupons_url>http://dm.api/20130701/users/1003/coupons/</coupons_url>
  <id>1003</id>
  <card>8460939940</card>
</root>
```


Пример ответа *API* в формате JSON на успешную установку процента скидки (параметр `percent`) пользователю с идентификатором `1003` (до вызова процент скидки `discount` был равен `3`):


```
{"purchases": 3,
 "first_name": "Иван Петрович
 Сидоров", "last_name": "",
 "middle_name":
 "", "bonus": 70,
 "purchases_url": "http://dm.api/20130701/users/1003/purchases/",
 "discount": 5,
 "amount": "50000.00",
 "loyalty_url": "http://dm.api/20130701/loyalties/2001",
 "photo_urls": {"100x125":
                "http://dm.api/img/default/avatar-100x100.png", "50x62":
                "http://dm.api/img/default/avatar-50x50.png"},
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/",
 "id": 1003,
 "card": "8460939940"}
```


Этот же ответ в формате XML:


```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <purchases>3</purchases>
  <first_name>Иван Петрович Сидоров</first_name>
  <last_name></last_name>
  <middle_name></middle_name>
  <bonus>70</bonus>
  <purchases_url>http://dm.api/20130701/users/1003/purchases/
  </purchases_url>
  <discount>5</discount>
  <amount>50000.00</amount>
  <loyalty_url>http://dm.api/20130701/loyalties/2001</loyalty_url>
  <photo_urls>
    <_100x125>http://dm.api/static/img/default/avatar-100x100.png</_100x100>
    <_50x62>http://dm.api/static/img/default/avatar-50x50.png</_50x50>
  </photo_urls>
  <coupons_url>http://dm.api/20130701/users/1003/coupons/</coupons_url>
  <id>1003</id>
  <card>8460939940</card>
</root>
```


Ресурс учётной записи предоставляет доступ к функциям *Системы*, предназначенным для управления данными, связанными с покупателем. Пример ссылки на ресурс пользователя с идентификатором `1003`:

```
https://dm.api/20130701/accounts/1003
```

Методы и параметры этого ресурса идентичны методам и параметрам ресурса пользователей, однако ответ на запрос информации об учётной записи отличается. Пример ответа *API* в формате JSON при наличии в системе учетной записи пользователя с идентификатором `1003`:

```
{"purchases": 3,
  "first_name": "Иван Петрович
  Сидоров", "last_name": "",
  "middle_name":
  "", "bonus": 70,
  "purchases_url":
  "http://dm.api/20130701/users/1003/purchases/", "discount": 3,
  "amount": "7000.00",
  "url": "http://dm.api/20130701/users/1003",
  "loyalty": {"currency_name": "RUB",
              "thresholds": [[0, 1], [10000, 3], [50000, 5]],
              "type": "amount",
              "currency_code":
              643},
  "photo_urls": {"100x125":
                 "http://dm.api/img/default/avatar-100x100.png", "50x62":
                 "http://dm.api/img/default/avatar-50x50.png"},
  "coupons": "1,2,3,4,5,6,7,8,9,10,11",
  "id": 1003,
  "card": "462001113901680935453311 6"}
```

4.6. Купоны пользователя (`/users/{user_id}/coupons/`)

Ресурс содержит не погашенные купоны (сертификаты) пользователя. Пример ссылки на контейнер купонов пользователя:

```
https://dm.api/20130701/users/2001/coupons/
```

Этот ресурс поддерживает метод:

- **GET.** Получение списка (с возможностью постраничной выборки) купонов пользователя. При этом возможно указание параметров фильтрации купонов:

Таблица 11. Параметры метода GET ресурса купонов пользователя.

Параметр	Тип данных	Описание
id	Целое положительное число.	Идентификатор покупки в *Системе*.
status	Строка вида: «ACTIVE», «USED» или «EXPIRED».	Состояние купона.
curr_iso_code	Целое положительное число.	Цифровой код валюты в стандар- те ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
curr_iso_name	Строка длиной 3.	Алфавитный код валюты в стандарте ISO 4217.
item_N_{attr}	Тип зависит от имени атрибута.	Атрибуты (реквизиты) номенклатурной позиции с номером *N* (начиная с 0) по-купки. Имена и типы атрибутов см. в Таблица 17, «Атрибуты номенклатурной позиции покупки.» [37]. Этот параметр обязателен при указании параметра *redeem_auto* в значении «true».

redeem_auto	Строка вида: «true» или «false».	Если значение «true», то купоны пользователя будут подобраны к покупке автоматически.
sum_total	Неотрицательное число с точностью до 2 знаков после запятой.	Сумма покупки без скидки. Формат числовых параметров см. в Числовые параметры [7]. Этот параметр обязателен при указании параметра *redeem_auto* в значении «true».

Строка списка купонов содержит идентификатор купона (используется в методе **POST** ресурса Раздел 4.14, «Покупки пользователя (/users/{user_id}/purchases/)» [31]) и ссылку на конкретный купон пользователя. Пример ссылки на купон с идентификатором 714:

```
https://dm.api/20130701/users/1003/coupons/714
```

Пример ответа API в формате JSON со списком купонов на вызов метода:

```
{"results":
 [{"status": "ACTIVE",
   "redeem_auto": false,
   "date_bought": "2012-09-15 20:00:00 +0700",
   "redeem_scope": "RS_ITEMS_IN_PURCHASE",
   "coupon_condition": "Условия акции: дата начала
   2012-09-01,
дата окончания 2013-11-01; скидка 40 %;
Платный купон позволяет получить скидку без
ограничений на сумму покупки",
   "curr_iso_code": 810,
   "number": "65489878",
   "curr_iso_name": "RUR",
   "url": "http://testserver/20130701/users/1003/coupons/1",
   "date_expiration": "2013-11-01 00:00:00 +0700",
   "award_type": "AWARD_DISCOUNT",
   "date_used": null,
   "award_value": "40.00",
   "offer_name":
   "unit-test", "id": 1,
   "redeem_condition": "RC_NOT_DEFINED"},
  {"status": "ACTIVE",
   "redeem_auto": false,
   "date_bought": "2012-09-15 20:15:00 +0700",
   "redeem_scope": "RS_ITEMS_IN_PURCHASE",
   "coupon_condition": "Условия акции: дата начала
   2012-09-01,
дата окончания 2013-11-01; скидка 40 %;
Платный купон позволяет получить скидку без
ограничений на сумму покупки",
   "curr_iso_code": 810,
   "number": "65489879",
   "curr_iso_name": "RUR",
   "url": "http://testserver/20130701/users/1003/coupons/2",
   "date_expiration": "2013-11-01 00:00:00 +0700",
   "award_type": "AWARD_DISCOUNT",
   "date_used": null,
   "award_value": "40.00",
   "offer_name":
   "unit-test", "id": 2,
   "redeem_condition": "RC_NOT_DEFINED"}
 ],
 "page": 1,
 "next": null,
 "per_page":
 20,
 "total": 2,
 "pages": 1,
 "previous":
 null}
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<root>
  <results>
    <list-item>
      <status>ACTIVE</status>
      <redeem_auto>False</redeem_auto>
      <date_bought>2012-09-15 20:00:00 +0700</date_bought>
      <redeem_scope>RS_ITEMS_IN_PURCHASE</redeem_scope>
      <coupon_condition>Условия акции: дата начала
2012-09-01, дата окончания 2013-11-01; скидка 40 %;
Платный купон позволяет получить скидку без
ограничений на сумму покупки</coupon_condition>
      <curr_iso_code>810</curr_iso_code>
      <number>65489878</number>
      <curr_iso_name>RUR</curr_iso_name>
      <url>http://testserver/20130701/users/1003/coupons/1</url>
      <date_expiration>2013-11-01 00:00:00 +0700</date_expiration>
      <award_type>AWARD_DISCOUNT</award_type>
      <date_used></date_used>
      <award_value>40.00</award_value>
      <offer_name>Мега-акция!</offer_name>
      <id>1</id>
      <redeem_condition>RC_NOT_DEFINED</redeem_condition>
    </list-item>
    <list-item>
      <status>ACTIVE</status>
      <redeem_auto>False</redeem_auto>
      <date_bought>2012-09-15 20:15:00 +0700</date_bought>
      <redeem_scope>RS_ITEMS_IN_PURCHASE</redeem_scope>
      <coupon_condition>Условия акции: дата начала
2012-09-01, дата окончания 2013-11-01; скидка 40 %;
Платный купон позволяет получить скидку без
ограничений на сумму покупки</coupon_condition>
      <curr_iso_code>810</curr_iso_code>
      <number>65489879</number>
      <curr_iso_name>RUR</curr_iso_name>
      <url>http://testserver/20130701/users/1003/coupons/2</url>
      <date_expiration>2013-11-01 00:00:00 +0700</date_expiration>
      <award_type>AWARD_DISCOUNT</award_type>
      <date_used></date_used>
      <award_value>40.00</award_value>
      <offer_name>Мега-акция!</offer_name>
      <id>2</id>
      <redeem_condition>RC_NOT_DEFINED</redeem_condition>
    </list-item>
  </results>
  <page>1</page>
  <next></next>
  <per_page>20</per_page>
  <total>2</total>
  <pages>1</pages>
  <previous></previous>
</root>
```

4.7. Купон пользователя (`/users/{user_id}/coupons/{id}`)

Ресурс представляет отдельный купон пользователя. Пример ссылки на покупку с идентификатором `714`:

```
https://dm.api/20130701/users/1003/coupons/714
```

Для ресурса доступны следующие методы:

- **GET.** Возвращает полную информацию о купоне (сертификате).

  Пример ответа API в формате JSON после успешного выполнения метода:

```
{"status": "ACTIVE",
 "redeem_auto":
 false,
 "curr_iso_code":
 810,
 "date_bought": "2012-09-15 20:00:00 +0700",
 "coupon_condition": "Условия акции: дата начала
```

```
2012-09-01, дата окончания 2013-11-01; скидка 40 %;
Платный купон позволяет получить скидку без ограничений
на сумму покупки",
 "purchase_url":
 null, "number":
 "65489878",
 "curr_iso_name": "RUR",
 "redeem_scope": "RS_ITEMS_IN_PURCHASE",
 "date_expiration": "2013-11-01 00:00:00
 +0700", "award_type": "AWARD_DISCOUNT",
 "date_used": null,
 "award_value": "40.00",
 "offer_name":
 "Мега-акция!", "id": 1,
 "redeem_condition": "RC_NOT_DEFINED"}
```

Тот же самый ответ в формате XML:

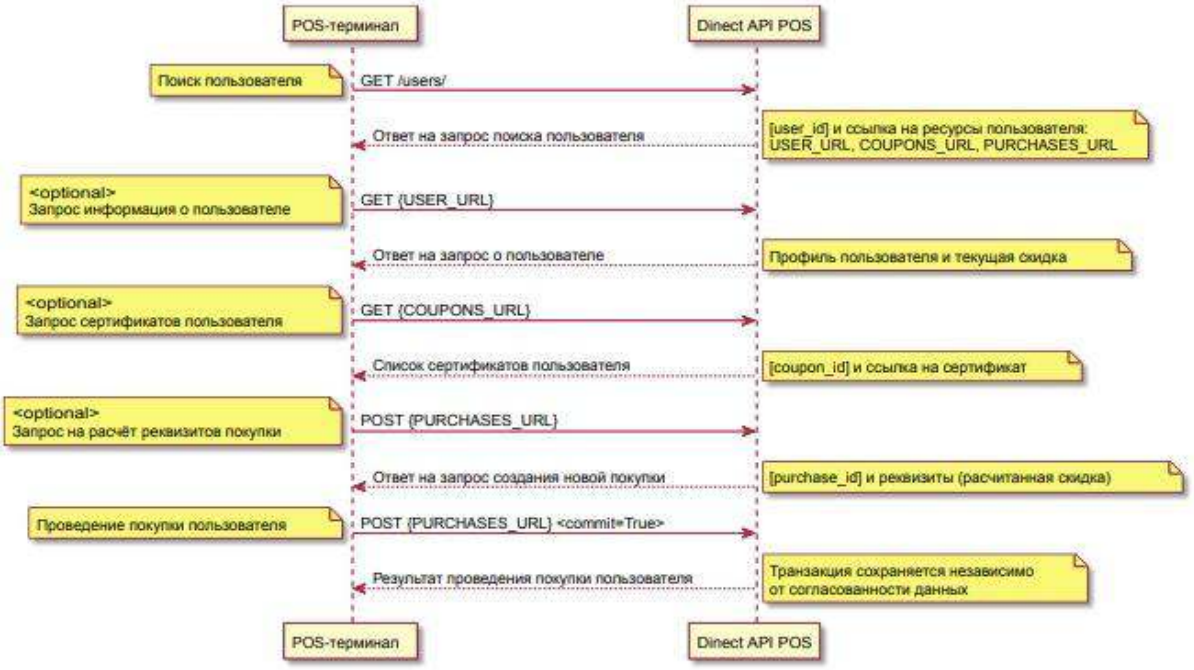
```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <status>ACTIVE</status>
  <redeem_auto>False</redeem_auto>
  <curr_iso_code>810</curr_iso_code>
  <date_bought>2012-09-15 20:00:00 +0700</date_bought>
  <coupon_condition>Условия акции: дата начала
2012-09-01, дата окончания 2013-11-01; скидка 40 %;
Платный купон позволяет получить скидку без
ограничений на сумму покупки</coupon_condition>
  <purchase_url></purchase_url>
  <number>65489878</number>
  <curr_iso_name>RUR</curr_iso_name>
  <redeem_scope>RS_ITEMS_IN_PURCHASE</redeem_scope>
  <date_expiration>2013-11-01 00:00:00 +0700</date_expiration>
  <award_type>AWARD_DISCOUNT</award_type>
  <date_used></date_used>
  <award_value>40.00</award_value>
  <offer_name>Мега-акция!</offer_name>
  <id>1</id>
  <redeem_condition>RC_NOT_DEFINED</redeem_condition>
</root>
```

4.8. Покупки пользователя (`/users/{user_id}/purchases/`)

Ресурс содержит покупки пользователя у данного *участника Системы* (*участник* находится по *токену* авторизации) и является основным для поддержки бизнес-процессов *участника Системы* компаниис. Пример ссылки на контейнер покупок пользователя с идентификатором `1003`:

```
https://dm.api/20130701/users/1003/purchases/
```

Ниже приведен пример типичного процесса обмена сообщениями между *ПО* интеграционного решения POS-системы и *API*:


POS-терминал
Direct API POS
Поиск пользователя
GET /users/
Ответ на запрос поиска пользователя
[user_id] и ссылка на ресурсы пользователя:
USER_URL, COUPONS_URL, PURCHASES_URL
<optional>
Запрос информация о пользователе
GET {USER_URL}
Ответ на запрос о пользователе
Профиль пользователя и текущая скидка
<optional>
Запрос сертификатов пользователя
GET {COUPONS_URL}
Список сертификатов пользователя
[coupon_id] и ссылка на сертификат
<optional>
Запрос на расчёт реквизитов покупки
POST {PURCHASES_URL}
Ответ на запрос создания новой покупки
[purchase_id] и реквизиты (расчитанная скидка)
Проведение покупки пользователя
POST {PURCHASES_URL} <commit=True>
Результат проведения покупки пользователя
Транзакция сохраняется независимо
от согласованности данных
POS-терминал
Direct API POS

Ресурс покупок пользователя поддерживает следующие методы:

- **GET.** Получение списка (с возможностью постраничной выборки) покупок пользователя у данного *участника Системы*. При этом возможно указание параметров фильтрации покупок:

Таблица 12. Параметры метода GET ресурса покупок пользователя.

Параметр	Тип данных	Описание
id	Целое положительное число.	Идентификатор покупки в *Системе*.
doc_id	Строка длиной до 50.	Идентификатор покупки в учетной системе *участника Системы*.
begin_date	Строка вида: «YYYY-MM-DD HH:MM:SS +hhmm».	Начало интервала фильтра покупок по дате и времени. Формат параметров даты и времени см. в Параметры даты и времени [7].
end_date	Строка вида: «YYYY-MM-DD HH:MM:SS +hhmm».	Окончание интервала фильтра покупок по дате и времени. Формат параметров даты и времени см. в Параметры даты и времени [7].
sum_total	Неотрицательное число с точностью до 2 знаков после запятой.	Сумма покупки без скидки. Формат числовых параметров см. в Числовые параметры [7].
sum_with_discount	Неотрицательное число с точностью до 2 знаков после запятой.	Сумма покупки со скидкой. Формат числовых параметров см. в Числовые параметры [7].
returned	Строка «true» или «false».	Фильтрует покупки по признаку - возвращена или нет.
sort	Строка «asc» или «desc».	Сортировка покупок по дате. Если передана строка «asc», то покупки сортируются по возрастанию даты. Если передана строка «desc» - по убыванию. Если параметр не указан - покупки выдаются в произвольном порядке.

Список покупок содержит ссылки на конкретные покупки пользователя в точках продаж *участника*. Пример ссылки на покупку с идентификатором 1325:

```
https://dm.api/20130701/users/2001/purchases/1325
```

Пример ответа метода в формате JSON со списком покупок пользователя:

```
{"results":
 [{"sum_total": "3000.00",
   "date": "2012-09-15 20:00:00 +0700",
```

```
    "url": "http://dm.api/20130701/users/1003/purchases/1",
    "sum_discount": "30.00",
    "curr_iso_code": 810,
    "curr_iso_name":
    "RUR", "pos": "1",
    "discount": 1,
    "coupons":
    null,
    "coupons_url": "http://dm.api/20130701/users/1003/coupons/?id=1",
    "items": [{"group_code": "8974126380001",
               "sum_total": "2000.00",
               "item_code": "8974126385215",
               "sum_with_discount":
               "1980.00", "item_gtin": "",
               "quantity": "2.000"},
              {"group_code": "8974126380001",
               "sum_total": "1000.00",
               "item_code": "8974126385216",
               "sum_with_discount":
               "990.00", "item_gtin": "",
               "quantity": "1.000"}],
    "sum_bonus": 0,
    "doc_id": "101",
    "id": 1,
    "items_url": "http://dm.api/20130701/users/1003/purchases/1/items/"},
   {"sum_total": "2070.00",
    "date": "2012-09-16 18:00:00 +0700",
    "url": "http://dm.api/20130701/users/1003/purchases/2",
    "sum_discount": "41.40",
    "curr_iso_code": 810,
    "curr_iso_name":
    "RUR", "pos": "2",
    "discount": 2,
    "coupons":
    null,
    "coupons_url": "http://dm.api/20130701/users/1003/coupons/?id=2",
    "items": [{"group_code": "8974126380002",
               "sum_total": "2070.00",
               "item_code": "8974126385217",
               "sum_with_discount": "2028.60",
               "item_gtin": "1693574465687",
               "quantity": "1.000"}],
    "sum_bonus": 0,
    "doc_id": "102",
    "id": 2,
    "items_url": "http://dm.api/20130701/users/1003/purchases/2/items/"},
   {"sum_total": "12000.00",
    "date": "2012-09-18 23:00:00 +0700",
    "url": "http://dm.api/20130701/users/1003/purchases/3",
    "sum_discount": "4800.00",
    "curr_iso_code": 810,
    "curr_iso_name": "RUR",
    "pos": "3",
    "discount": 40,
    "coupons":
    null,
    "coupons_url": "http://dm.api/20130701/users/1003/coupons/?id=3",
    "items": [{"group_code": "8974126380001",
               "sum_total": "10000.00",
               "item_code": "8974126385215",
               "sum_with_discount": "6000.00",
               "item_gtin": "1693574465685",
               "quantity": "10.000"},
              {"group_code": "8974126380001",
               "sum_total": "2000.00",
               "item_code": "8974126385216",
               "sum_with_discount": "1200.00",
               "item_gtin": "1693574465686",
               "quantity": "2.000"}],
    "sum_bonus": 0,
    "doc_id": "103",
    "id": 3,
    "items_url": "http://dm.api/20130701/users/1003/purchases/3/items/"}
  ],
  "page": 1,
  "next": null,
  "per_page":
  20,
  "total": 3,
  "pages": 1,
  "previous":
```

```
 null}
```

Этот же ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <results>
    <list-item>
      <sum_total>3000.00</sum_total>
      <date>2012-09-15 20:00:00 +0700</date>
      <url>http://dm.api/20130701/users/1003/purchases/1</url>
      <sum_discount>30.00</sum_discount>
      <curr_iso_code>810</curr_iso_code>
      <curr_iso_name>RUR</curr_iso_name>
      <pos>1</pos>
      <discount>1</discount>
      <coupons></coupons>
      <coupons_url>
http://dm.api/20130701/users/1003/coupons/?id=1</coupons_url>
      <items>
        <list-item>
          <group_code>8974126380001</group_code>
          <sum_total>2000.00</sum_total>
          <item_code>8974126385215</item_code>
          <sum_with_discount>1980.00</sum_with_discount>
          <item_gtin>1693574465685</item_gtin>
          <quantity>2.000</quantity>
        </list-item>
        <list-item>
          <group_code>8974126380001</group_code>
          <sum_total>1000.00</sum_total>
          <item_code>8974126385216</item_code>
          <sum_with_discount>990.00</sum_with_discount>
          <item_gtin>1693574465686</item_gtin>
          <quantity>1.000</quantity>
        </list-item>
      </items>
      <sum_bonus>0</sum_bonus>
      <doc_id>101</doc_id>
      <id>1</id>
      <items_url>
http://dm.api/20130701/users/1003/purchases/1/items/</items_url>
    </list-item>
    <list-item>
      <sum_total>2070.00</sum_total>
      <date>2012-09-16 18:00:00 +0700</date>
      <url>http://dm.api/20130701/users/1003/purchases/2</url>
      <sum_discount>41.40</sum_discount>
      <curr_iso_code>810</curr_iso_code>
      <curr_iso_name>RUR</curr_iso_name>
      <pos>2</pos>
      <discount>2</discount>
      <coupons></coupons>
      <coupons_url>
http://dm.api/20130701/users/1003/coupons/?id=2</coupons_url>
      <items>
        <list-item>
          <group_code>8974126380002</group_code>
          <sum_total>2070.00</sum_total>
          <item_code>8974126385217</item_code>
          <sum_with_discount>2028.60</sum_with_discount>
          <item_gtin>1693574465687</item_gtin>
          <quantity>1.000</quantity>
        </list-item>
      </items>
      <sum_bonus>0</sum_bonus>
      <doc_id>102</doc_id>
      <id>2</id>
      <items_url>
http://dm.api/20130701/users/1003/purchases/2/items/</items_url>
    </list-item>
    <list-item>
      <sum_total>12000.00</sum_total>
      <date>2012-09-18 23:00:00 +0700</date>
      <url>http://dm.api/20130701/users/1003/purchases/3</url>
      <sum_discount>4800.00</sum_discount>
      <curr_iso_code>810</curr_iso_code>
      <curr_iso_name>RUR</curr_iso_name>
      <pos>3</pos>
      <discount>40</discount>
```

```
      <coupons></coupons>
      <coupons_url>
http://dm.api/20130701/users/1003/coupons/?id=3</coupons_url>
      <items>
        <list-item>
          <group_code>8974126380001</group_code>
          <sum_total>10000.00</sum_total>
          <item_code>8974126385215</item_code>
          <sum_with_discount>6000.00</sum_with_discount>
          <item_gtin>1693574465685</item_gtin>
          <quantity>10.000</quantity>
        </list-item>
        <list-item>
          <group_code>8974126380001</group_code>
          <sum_total>2000.00</sum_total>
          <item_code>8974126385216</item_code>
          <sum_with_discount>1200.00</sum_with_discount>
          <item_gtin>1693574465686</item_gtin>
          <quantity>2.000</quantity>
        </list-item>
      </items>
      <sum_bonus>0</sum_bonus>
      <doc_id>103</doc_id>
      <id>3</id>
      <items_url>
http://dm.api/20130701/users/1003/purchases/3/items/</items_url>
    </list-item>
  </results>
  <page>1</page>
  <next></next>
  <per_page>20</per_page>
  <total>3</total>
  <pages>1</pages>
  <previous></previous>
</root>
```

В таблице ниже приведено полное описание полей ответа:
Таблица 13. Параметры ответа на метод GET ресурса покупок пользователя.

Параметр	Тип данных	Описание
results	Массив (JSON) или список (XML).	Контейнер (массив или список) покупок (данные о покупке см. в Таблица 14, «Поля покупки.» [36]).
page	Целое положительное число.	Номер страницы ответа.
next	URL (HTTP-ссылка).	Ссылка на следующую страницу ответа (`null` или пусто - нет следующей страницы).
per_page	Целое положительное число.	Максимальное количество покупок на одной странице.
total	Целое неотрицательное число.	Общее количество покупок в ответе (на всех страницах).
pages	Целое положительное число.	Количество страниц в ответе.
previous	URL (HTTP-ссылка).	Ссылка на предыдущую страницу ответа (`null` или пусто - нет предыдущей страницы).

В следующей таблице содержится описание полей покупки:

Таблица 14. Поля покупки.

Параметр	Тип данных	Описание
id	Целое положительное число.	Идентификатор покупки (уникален в *системе*).

date	YYYY-MM-DD HH:MM:SS +hhmm	Дата и время регистрации покупки в системе с указанием часового пояса. Формат параметров даты и времени см. в Параметры даты и времени [7].
url	Строка символов.	Единый указатель ресурсов - URL [http:// ru.wikipedia.org/wiki/URL].
doc_id	Строка длиной до 50.	Идентификатор покупки в учетной системе *участника*.
pos	Строка длиной до 25.	Идентификатор кассы, с которой была зарегистрирована эта покупка.
curr_iso_code	Целое положительное число.	Цифровой код валюты в стандар- те ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
curr_iso_name	Строка символов длиной 3.	Алфавитный код валюты в стандарте ISO 4217.
sum_total	Неотрицательное число с точностью 2 знака после запятой.	Сумма покупки без скидки. Формат числовых параметров см. в Числовые параметры [7].
discount	Целое неотрицательное число.	Фактическая скидка на покупку в процентах (округленно).
sum_discount	Неотрицательное число с точностью 2 знака после запятой.	Сумма скидки на покупку. Формат числовых параметров см. в Числовые параметры [7].
sum_bonus	Целое число с точностью 2 знака после запятой.	Количество бонусов, начисленных или списанных за покупку. Формат числовых параметров см. в Числовые параметры [7].
coupons	Список алфавитно-цифровых строк без знаков препинания, разделенных запятыми.	Список идентификаторов купонов, прикрепления к документу покупки.
coupons_url	URL (HTTP-ссылка).	Ссылка на купоны, погашенные по этой покупке.
items	Массив структур.	Список номенклатурных позиций покупки (см. Таблица 15, «Поля номенклатурной позиции покупки.» [37]).
items_url	URL (HTTP-ссылка).	Ссылка на номенклатурные позиции покупки

Набор параметров с префиксом *item_* задаёт реквизиты номенклатурных позиций покупки. Компонент *N* имени параметра задает номер строки номенклатурной позиции (начиная с 0). Например, для кода товара (в учетной системе *участника*) в 3-й строке покупки имя параметра будет: *item_2_id*.

В следующей таблице содержится описание полей номенклатурной позиции покупки:

Таблица 15. Поля номенклатурной позиции покупки.

Параметр	Тип данных	Описание
item_gtin	Строка цифр длиной до 14.	Идентификатор товара формата GTIN [http://ru.wikipedia.org/wiki/GTIN] в стандарте GS1 [http://www.gs1ru.org/].
item_code	Строка длиной до 100.	Идентификатор товара в учетной системе *участника*.
group_code	Строка длиной до 100.	Идентификатор группы товара в учетной системе *участника*.
quantity	Неотрицательное число с точностью 3 знака после запятой.	Количество товара в номенклатурной позиции покупки. Формат числовых параметров см. в Числовые параметры [7].
sum_total	Неотрицательное число с точностью 2 знака после запятой.	Стоимость номенклатурной позиции покупки без скидки. Формат числовых параметров см. в Числовые параметры [7].
sum_with_discount	Неотрицательное число с точностью 2 знака после запятой.	Стоимость номенклатурной позиции покупки со скидкой. Формат числовых параметров см. в Числовые параметры [7].

- **POST.** Проведение покупки. Метод принимает следующие параметры:

Таблица 16. Параметры метода POST ресурса покупок пользователя.

Параметр	Тип данных	Описание
doc_id	Строка длиной до 50.	Идентификатор покупки в учетной системе *участника*.
curr_iso_code	Целое положительное число.	Цифровой код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].
curr_iso_name	Строка символов длиной 3.	Алфавитный код валюты в стандарте ISO 4217.
sum_total	Неотрицательное число с точностью 2 знака после запятой.	Сумма покупки без скидки. Формат числовых параметров см. в Числовые параметры [7].
sum_with_discount	Неотрицательное число с точностью 2 знака после запятой.	Сумма покупки со скидкой. Формат числовых параметров см. в Числовые параметры [7].
bonus_amount	Неотрицательное число с точностью 2 знака после запятой.	Количество бонусов, начисленных пользователю за покупку. Формат числовых параметров см. в Числовые параметры [7].

bonus_payment	Неотрицательное число с точностью 2 знака после запятой.	Количество бонусов, потраченных пользователем на покупку. Формат числовых параметров см. в Числовые параметры [7].
coupons	Список алфавитно-цифровых строк без знаков препинания, разделенных запятыми.	Список идентификаторов купонов (сертификатов) пользователя, которые следует погасить в этой покупке.
date	Дата и время в формате ISO.	Дата и время покупки. Формат см. в Параметры даты и времени [7]. Этот параметр принимается только для отдельных клиентов.
item_N_{attr}	Тип зависит от имени атрибута.	Атрибуты (реквизиты) номенклатурной позиции с номером *N* (начиная с 0) покупки. Имена и типы атрибутов см. в Таблица 17, «Атрибуты номенклатурной позиции покупки.» [37].
redeem_auto	Строка вида «true» или «false». По умолчанию «false».	Если значение «true», то купоны пользователя будут подобраны к покупке автоматически.
override	Строка вида «true» или «false». По умолчанию «false».	Если значение «true», то сохранить покупку с реквизитами, полученными в запросе, не выполняя при этом каких-либо расчётов (по программам лояльности и купонам).
gift	Строка вида «true» или «false». По умолчанию «false».	Если значение «true», то включить расширение подарочных накоплений. При включенном расширении ответ от API содержит дополнительный блок данных.
commit	Строка вида «true» или «false».	Если значение «true», то покупка проводится (сохраняется). В противном случае (False или параметр не указан) производится автоматический расчет не указанных в запросе реквизитов покупки.

Ниже приведено описание атрибутов номенклатурной позиции покупки.

Таблица 17. Атрибуты номенклатурной позиции покупки.

Параметр	Тип данных	Описание
gtin	Строка цифр длиной 13.	Идентификатор товара формата GTIN [http://ru.wikipedia.org/wiki/GTIN] в стандарте GS1 [http://www.gs1ru.org/].
id	Строка длиной до 100.	Идентификатор товара в учетной системе *участника*.
gid	Строка длиной до 100.	Идентификатор группы товара в учетной системе *участника*.
q	Неотрицательное число с точностью 3 знака после запятой.	Количество товара в номенклатурной позиции покупки. Формат числовых параметров см. в Числовые параметры [7].
sum	Неотрицательное число с точностью 2 знака после запятой.	Стоимость номенклатурной позиции покупки без скидки. Формат числовых параметров см. в Числовые параметры [7].
sumwd	Неотрицательное число с точностью 2 знака после запятой.	Стоимость номенклатурной позиции покупки со скидкой. Формат числовых параметров см. в Числовые параметры [7].

Метод всегда возвращает статус HTTP 201 Created при успешном проведении (200 OK в случае, если параметр *commit* не указан или имеет значение False) и полную информацию о проведённой покупке.

Таблица 18. Параметры ответа метода POST ресурса покупок пользователя.

Параметр	Тип данных	Описание
id	Целое положительное число.	Идентификатор покупки в *Системе*.
url	URL	Ссылка на покупку в API.
doc_id	Строка длиной до 50.	Идентификатор покупки в учетной системе *участника*.
date	Дата и время в формате ISO.	Дата и время покупки. Формат см. в Параметры даты и времени [7]. Этот параметр принимается только для отдельных клиентов.
sum_total	Неотрицательное число с точностью 2 знака после запятой.	Сумма покупки без скидки. Формат числовых параметров см. в Числовые параметры [7].
sum_discount	Неотрицательное число с точностью 2 знака после запятой.	Сумма скидки. Формат числовых параметров см. в Числовые параметры [7].
curr_iso_code	Целое положительное число.	Цифровой код валюты в стандарте ISO 4217 [http://ru.wikipedia.org/wiki/ISO_4217].

curr_iso_name	Строка символов длиной 3.	Алфавитный код валюты в стандарте ISO 4217.
pos	Строка символов.	Идентификатор POS в учетной системе участника.
discount	Целое неотрицательное число.	Общая скидка на покупку в процентах (округлено до целого).
sum_bonus	Неотрицательное число с точностью 2 знака после запятой.	Количество бонусов, начисленных пользователю за покупку или списанных за скидку. Формат числовых параметров см. в Числовые параметры [7]. При одновременном начислении и списании бонусных баллов это строка вида «N начислено, M списано», где M - целое неотрицательное число списанных в покупке бонусных баллов, а N - целое неотрицательное число начисленных в покупке бонусных баллов.
coupons	Список алфавитно-цифровых строк без знаков препинания, разделенных запятыми.	Список идентификаторов купонов (сертификатов) пользователя, которые следует погасить в этой покупке.
coupons_url	URL	Ссылка на купоны, погашенные в этой покупке.
items	Массив объектов.	Атрибуты (реквизиты) номенклатурных позиций покупки. Имена и типы атрибутов см. в Таблица 17, «Атрибуты номенклатурной позиции покупки.» [37].
items_url	URL	Ссылка на ресурс товарных позиций покупки.
override	Строка вида «true» или «false». По умолчанию «false».	Если значение «true», то сохранить покупку с реквизитами, полученными в запросе, не выполняя при этом каких-либо расчётов (по программам лояльности и купонам).

Пример ответ от API в формате JSON:

```
{"sum_total": "1000.00",
 "date": "2013-02-01 17:05:12 +0700",
 "url": "http://dm.api/20130701/users/1003/purchases/4",
 "sum_discount": "30.00",
 "curr_iso_code": 810,
 "curr_iso_name":
 "RUR", "pos": "1",
 "discount": 3,
 "coupons": "10741",
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/?id=4",
 "items": [{"group_code": "4678523457850",
            "sum_total": "1000.00",
            "item_code": "2561389462075",
            "sum_with_discount":
            "0.00", "item_gtin": "",
            "quantity": "1.700"},
           {"group_code": "4678523457850",
            "sum_total": "1000.00",
            "item_code": "2561389462074",
```

```
            "sum_with_discount":
            "0.00", "item_gtin": "",
            "quantity": "1.700"}],
 "sum_bonus": 0,
 "doc_id":
 "КЧ-205", "id": 4,
 "items_url": "http://dm.api/20130701/users/1003/purchases/4/items/"}
```

Этот же ответ в формате XML:

```
<root>
  <sum_total>1000.00</sum_total>
  <date>2013-02-01 17:06:53 +0700</date>
  <url>http://dm.api/20130701/users/1003/purchases/4</url>
  <sum_discount>30.00</sum_discount>
  <curr_iso_code>810</curr_iso_code>
  <curr_iso_name>RUR</curr_iso_name>
  <pos>1</pos>
  <discount>3</discount>
  <coupons>10741</coupons>
  <coupons_url>
http://dm.api/20130701/users/1003/coupons/?id=4</coupons_url>
  <items>
    <list-item>
      <group_code>4678523457850</group_code>
      <sum_total>1000.00</sum_total>
      <item_code>2561389462075</item_code>
      <sum_with_discount>0.00</sum_with_discount>
      <item_gtin></item_gtin>
      <quantity>1.700</quantity>
    </list-item>
    <list-item>
      <group_code>4678523457850</group_code>
      <sum_total>1000.00</sum_total>
      <item_code>2561389462074</item_code>
      <sum_with_discount>0.00</sum_with_discount>
      <item_gtin></item_gtin>
      <quantity>1.700</quantity>
    </list-item>
  </items>
  <sum_bonus>0</sum_bonus>
  <doc_id>КЧ-205</doc_id>
  <id>4</id>
  <items_url>
http://dm.api/20130701/users/1003/purchases/4/items/</items_url>
</root>
```

Параметры ответа подробно описаны в Таблица 18, «Параметры ответа метода **POST** ресурса покупок пользователя.» [38]. Если в запросе на проведение (или предварительный расчёт) покупки был указан цифровой купон по акции, которая действует на конкретные номенклатурные позиции, и были указаны все реквизиты номенклатурных позиций покупки, то параметр *items* ответа содержит рассчитанную стоимость товаров покупки с учётом гашения указанного купона (купонов) и применения программы лояльности точки продажи.

Если покупка не проводится (параметр *commit* не указан или имеет значение «false»), то в ответ на запрос возвращаются автоматически рассчитанные для указанного покупателя, соответствующие программе лояльности организации - владельца магазина, реквизиты покупки - такие как сумма покупки со скидкой, количество начисленных бонусов, сумма уменьшения стоимости покупки при оплате бонусами. Пример ответа от API в формате JSON на запрос без проведения покупки с указанием оплаты части покупки бонусами:

```
{"sum_total": "1000.00",
 "date": "2012-11-26 11:39:37 +0700",
 "url": null,
 "sum_discount":
 "250.00",
 "curr_iso_code": 810,
 "curr_iso_name":
 "RUR", "pos": "1",
 "discount": 25,
 "coupons_url":
```

```
 null, "items_url":
 null, "items": [],
 "sum_bonus": -50,
 "doc_id":
 "КЧ-128", "id": 4}
```


Этот же ответ в формате XML:


```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <sum_total>1000.00</sum_total>
  <date>2012-11-26 11:49:28 +0700</date>
  <url></url>
  <sum_discount>250.00</sum_discount> <curr_iso_code>810</curr_iso_code>
  <curr_iso_name>RUR</curr_iso_name>
  <pos>1</pos>
  <discount>25</discount>
  <coupons_url></coupons_url>
  <items></items>
  <items_url></items_url>
  <sum_bonus>-50</sum_bonus>
  <doc_id>КЧ-128</doc_id>
  <id>4</id>
</root>
```

Примечание

Запрос со значением «`true`» параметра *`commit`* всегда приводит к полному расчету всех параметров покупки и гашению цифровых купонов. Поэтому проведе- ние покупки (*`commit`* равен «`true`») необходимо выполнять с теми же параметрами, что и запрос на предварительный расчёт реквизитов покупки без проведения (*`commit`* не указан или его значение «`false`»).

Одновременное начисление и списание бонусных баллов. При одновременном начислении и списании бонусных баллов (что настраивается в программе лояльности) ответ от API содержит раздельную информацию о том, сколько бонусных баллов было начислено и спи- сано. Например:


```
{"sum_total": "1000.00",
 "date": "2012-11-26 11:39:37 +0700",
 "url": null,
 "sum_discount":
 "250.00",
 "curr_iso_code": 810,
 "curr_iso_name":
 "RUR", "pos": "1",
 "discount": 25,
 "coupons_url":
 null, "items_url":
 null, "items": [],
 "sum_bonus": "50 начислено, 100
 списано", "doc_id": "КЧ-128",
 "id": 4}
```


Этот же ответ в формате XML:


```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <sum_total>1000.00</sum_total>
  <sum_discount>250.00</sum_discount>
  <pos>1</pos>
  <discount>25</discount>
  <date>2012-11-26 11:39:37 +0700</date>
  <coupons_url></coupons_url>
  <id>4</id>
  <coupons></coupons>
  <sum_bonus>50 начислено, 100 списано</sum_bonus>
  <url></url>
  <items></items>
  <curr_iso_code>643</curr_iso_code>
  <curr_iso_name>RUB</curr_iso_name>
  <override>False</override>
```

```
  <doc_id>КЧ-127</doc_id>
  <items_url>http://testserver/20130701/users/1003/purchases/4/items/</items_url>
</root>
```

Расширение подарочных накоплений. При включенном расширении подарочных накоплений (в запросе был передан параметр *gift* с значением «true») ответ от API содержит блок данных вида:

```
"gift_counters":[
  {"applied":true,
   "coupon":null,
   "current":"2.000",
   "limit":"5.000",
   "until_date":2018-12-10,
   "exp_date":"2018-12-17",
   "offer_name":"6th cup as a
   gift",
   "offer_info":"Купи 5 чашек и получи 6-ю в подарок!"
  }]
```

Этот же ответ в формате XML:

```
<gift_counters>
  <list-item>
    <applied>True</applied>
    <coupon></coupon>
    <current>2.000</current>
    <limit>5.000</limit>
    <until_date>2018-12-10</until_date>
    <exp_date>2018-12-17</exp_date>
    <offer_name>6th cup as a gift</offer_name>
    <offer_info>Купи 5 чашек и получи 6-ю в подарок!</offer_info>
  </list-item>
</gift_counters
       >
```

Блок данных содержит список накоплений *Пользователя* в подарочных акциях.

Таблица 19. Атрибуты элемента подарочных накоплений.

Параметр	Тип данных	Описание
applied	Строка вида «True» или «False».	Истина, если подарочная акция применена к покупке (изменён счетчик или выдан подарок).
coupon	Строка арабских цифр длиной 25.	Идентификатор выданного купона на получение вознаграждения по подарочной акции.
current	Неотрицательное число с фиксированной точкой.	Текущее значение счётчика накоплений *Пользователя* в подарочной акции.
limit	Неотрицательное число с фиксированной точкой.	Значение счетчика накоплений Пользователя в подарочной акции для получения подарка.
until_date	Дата в формате ISO.	Дата, до которой необходимо увеличить счётчик накоплений до значения *limit*, чтобы получить подарок. В эту дату счётчик накоплений сбрасывается в 0.
exp_date	Дата в формате ISO.	Срок действия подарочной акции.
offer_name	Строка символов.	Наименование подарочной акции.
offer_info	Строка символов.	Описание подарочной акции.

4.9. Покупка пользователя (`/users/{user_id}/purchases/{id}`)

Ресурс представляет отдельную покупку пользователя у данного *участника* Системы (*участник* находится по *токену* авторизации). Пример ссылки на покупку с идентификатором `1325`:

```
https://dm.api/20130701/users/1003/purchases/1325
```

Для ресурса доступны следующие методы:

- **GET.** Возвращает полную информацию о покупке. Пример ответа *API* в формате JSON на вызов метода по приведённой выше ссылке:

```
{"sum_bonus": 0,
 "sum_total": "3000.00",
 "sum_discount": "30.00",
 "curr_iso_code": 810,
 "curr_iso_name":
 "RUR", "pos": "1",
 "discount": 1,
 "coupons": "10741",
 "coupons_url": "http://dm.api/20130701/users/1003/coupons/?id=1",
 "items": [{"group_code": "8974126380001",
            "sum_total": "2000.00",
            "item_code": "8974126385215",
            "sum_with_discount": "1980.00",
            "item_gtin": "1693574465685",
            "quantity": "2.000"},
           {"group_code": "8974126380001",
            "sum_total": "1000.00",
            "item_code": "8974126385216",
            "sum_with_discount": "990.00",
            "item_gtin": "1693574465686",
            "quantity": "1.000"}],
 "date": "2012-09-15 20:00:00 +0700",
 "doc_id": "101",
 "id": 1,
 "items_url": "http://dm.api/20130701/users/1003/purchases/1/items/"}
```

  Тот же самый ответ в формате XML:

```
<root>
  <sum_bonus>0</sum_bonus>
  <sum_total>3000.00</sum_total>
  <sum_discount>30.00</sum_discount>
  <curr_iso_code>810</curr_iso_code>
  <curr_iso_name>RUR</curr_iso_name>
  <pos>1</pos>
  <discount>1</discount>
  <coupons>10741</coupons>
  <coupons_url>
http://dm.api/20130701/users/1003/coupons/?id=1</coupons_url>
  <items>
    <list-item>
      <group_code>8974126380001</group_code>
      <sum_total>2000.00</sum_total>
      <item_code>8974126385215</item_code>
      <sum_with_discount>1980.00</sum_with_discount>
      <item_gtin>1693574465685</item_gtin>
      <quantity>2.000</quantity>
    </list-item>
    <list-item>
      <group_code>8974126380001</group_code>
      <sum_total>1000.00</sum_total>
      <item_code>8974126385216</item_code>
      <sum_with_discount>990.00</sum_with_discount>
      <item_gtin>1693574465686</item_gtin>
      <quantity>1.000</quantity>
    </list-item>
  </items>
  <date>2012-09-15 20:00:00 +0700</date>
  <doc_id>101</doc_id>
```

```
    <id>1</id>
    <items_url>
  http://dm.api/20130701/users/1003/purchases/1/items/</items_url>
  </root>
```

- **DELETE.** Проведение возврата на всю сумму покупки. При успешном выполнении метод возвращает HTTP ответ со статусом 204 No Content.

4.9.1. Номенклатура покупки (`/users/{user_id}/purchases/{id}/items/`)

Ресурс представляет контейнер номенклатурных позиций покупки пользователя у данного *участника* Системы (*участник* находится по *токену* авторизации). Пример ссылки на контейнер номенклатурных позиций покупки с идентификатором `1325`:

```
https://dm.api/20130701/users/1003/purchases/1325/items/
```

Для ресурса доступны следующие методы:

- **GET.** Возвращает полный список номенклатурных позиций покупки. Пример ответа *API* в формате JSON на вызов метода по приведённой выше ссылке:

```
{"results": [
  {"group_code": "8974126380001",
   "sum_total": "2000.00",
   "item_code": "8974126385215",
   "sum_with_discount": "1980.00",
   "id": 1,
   "quantity": "2.000"},
  {"group_code": "8974126380001",
   "sum_total": "1000.00",
   "item_code": "8974126385216",
   "sum_with_discount": "990.00",
   "id": 2,
   "quantity": "1.000"}],
 "page": 1,
 "next": null,
 "per_page":
 20,
 "total": 2,
 "pages": 1,
 "previous":
 null}
```

Тот же самый ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <results>
    <list-item>
      <group_code>8974126380001</group_code>
      <sum_total>2000.00</sum_total>
      <item_code>8974126385215</item_code>
      <sum_with_discount>1980.00</sum_with_discount>
      <id>1</id>
      <quantity>2.000</quantity>
    </list-item>
    <list-item>
      <group_code>8974126380001</group_code>
      <sum_total>1000.00</sum_total>
      <item_code>8974126385216</item_code>
      <sum_with_discount>990.00</sum_with_discount>
      <id>2</id>
      <quantity>1.000</quantity>
    </list-item>
  </results>
  <page>1</page>
  <next></next>
  <per_page>20</per_page>
  <total>2</total>
  <pages>1</pages>
  <previous></previous>
```

```
</root>
```

4.9.2. Номенклатурная позиция покупки (/users/{user_id}/ purchases/{id}/items/{item_id})

Ресурс представляет конкретную номенклатурную позицию покупки пользователя у *участника* Системы (*участник* находится по *токену* авторизации). Пример ссылки на номенклатурную позицию с идентификатором `3517`, которая содержится в покупке с идентификатором `1325`:

```
https://dm.api/20130701/users/1003/purchases/1325/items/3517
```

Для ресурса доступны следующие методы:

- **GET.** Возвращает конкретную номенклатурную позицию покупки. Пример ответа *API* в формате JSON на вызов метода по приведённой выше ссылке:

```
{"group_code": "8974126380002",
 "sum_total": "2070.00",
 "item_code": "8974126385217",
 "sum_with_discount": "2028.60",
 "id": 3517,
 "quantity": "1.000"}
```

  Тот же самый ответ в формате XML:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <group_code>8974126380002</group_code>
  <sum_total>2070.00</sum_total>
  <item_code>8974126385217</item_code>
  <sum_with_discount>2028.60</sum_with_discount>
  <id>3517</id>
  <quantity>1.000</quantity>
</root>
```

Приём-передаточные испытания

Модуль должен быть проверен в каждом из режимов работы, каждая отдельная операция, которую способен выполнять модуль, должна быть проверена в нормальных условиях работы кассы (согласно документации кассового программного обеспечения), и каналов связи, а также при следующих условиях:

- Имитация неустойчивого канала связи;
- Имитация отсутствия связи;
- Аварийное завершение работы кассы в ходе выполнения операций;
- Некорректный ввод информации оператором кассы.

Модуль должен демонстрировать стабильную работу во всех испытаниях. Не допускается запись модулем некорректных данных в базу данных кассы или отправка модулем некорректных данных на сервер или аварийное завершение работы модуля и/или кассы.

1. Общие требования к приёмке работы

Приемка работ должна проводиться комиссией, в которой присутствуют ответственные со стороны Исполнителя и Заказчика.

2. Приёмно-передаточные испытания перечень

1. Идентификация по карте, номеру телефона, номеру купона.
    1.1. Какая информация по пользователю отражается. Использовать другие идентификаторы.
2. Проведение покупки с бонусными баллами.
    2.1. Программа лояльности настроена на раздельное начисление/списание бонусных баллов.
        2.1.1. Проведение покупки без расходования бонусных баллов. Производится начисление.
        2.1.2. Проведение покупки со списанием бонусных баллов. Начисление бонусных баллов не производится.

2.2. Программа лояльности настроена на одновременное начисление/списание бонусных баллов.

2.2.1. Проведение покупки без расходования бонусных баллов. Производится начисление.

2.2.2. Проведение покупки со списанием бонусных баллов. Начисление бонусных баллов производится на сумму покупки с учётом всех скидок.

3. Проведение возврата (бонусная программа).

3.1. Проведение возврата в операционный день.

3.2. Проведение возврата в отличный от текущего операционного дня (когда покупка была проведена днем ранее, например). Данный бизнес тестируется, если он заложен ПО.

Обработка цифровых купонов. В сценарии тестирования участвуют цифровые купоны: номенклатурные (скидка на определенную товарную позицию), номенклатурные (скидка на определенную товарную позицию, redeem_auto), цифровой купон со скидкой на весь документ, цифровой купон со скидкой на весь документ (redeem_auto), с номиналом скидки: % и сумма с признаком redeem_auto и без данного признака соответственно. Таким образом, формируется необходимое множество цифровых купонов. Цифровые купоны обрабатываются как в комбинации с программой лояльности, так и отдельно.

4. Проведение покупки цифровой купон (скидка на весь документ) (номинал - % скидки) и бонусная программа лояльности.

4.1. Проведение покупки с использованием купона (скидка на весь документ) (номинал % скидки).

5. Проведение возврата (скидка на весь документ) (бонусная программа, использованный цифровой купон (номинал % скидки)).

5.1. Проведение возврата в операционный день.

5.2. Проведение возврата в отличный от текущего день (когда покупка была проведена днем ранее, например). Данный бизнес тестируется, если он заложен ПО.

6. Проведение покупки с использованием цифрового купона (скидка на весь документ) (номинал - сумма скидки) и бонусная программа лояльности.

7. Проведение возврата – цифровой купон (скидка на весь документ) (номинал - сумма скидки) и бонусная программа лояльности в тот же опер день и в отличный от текущего (если предусмотрено ПО).

8. Проведение покупки с использованием цифрового купона (скидка на товар) (номинал купона %) и бонусная программа лояльности.

9. Проведение покупки с использованием цифрового купона (скидка на товар) (номинал - сумма) и бонусная программа лояльности.

10. Проведение покупки с использованием купона (скидка на документ %) и купона (скидка на товар %) и бонусная программа лояльности.

11. Проведение покупки с использованием купона (скидка на документ $) и купона (скидка на товар $) и бонусная программа лояльности.

12. Проведение покупки с использованием купона (скидка на документ %) и купона (скидка на товар $) и бонусная программа лояльности.

13. Проведение покупки с использованием купона (скидка на документ $) и купона (скидка на товар %) и бонусная программа лояльности.

14. Проведение покупки с использованием купона (скидка на товар $) и купона (скидка на товар $) и бонусная программа лояльности. Возврат.

15. Проведение покупки с использованием купона (скидка на товар %) и купона (скидка на товар %) и бонусная программа лояльности. Возврат.

16. Проведение покупки с использованием купона (скидка на товар $) и купона (скидка на товар %) и бонусная программа лояльности. Возврат.

17. Проведение покупки со скидкой процентом – Дисконт.

18. Проведение возврата со скидкой процентом – Дисконт. Текущий опер день.

19. Проведение возврата со скидкой процентом – Дисконт. Другой операционный день (если предусмотрено ПО).

20. Проведение покупки цифровой купон (скидка на весь документ) (номинал - % скидки) и Программа лояльности – Дисконт.

21. Проведение возврата покупки с цифровым купоном (скидка на весь документ, номинал %) и Программа лояльности – Дисконт.

    21.1. Проведение возврата в операционный день.

    21.2. Проведение возврата в отличный от текущего день (когда покупка была проведена днем ранее, например). Данный бизнес тестируется, если он заложен ПО.

22. Проведение покупки с использованием цифрового купона (скидка на весь документ) (номинал - сумма скидки) и программа лояльности Дисконт.

23. Проведение возврата – цифровой купон (скидка на весь документ) (номинал - сумма скидки) и программа лояльности Дисконт в тот же опер день и в отличный от текущего (если предусмотрено ПО).

24. Проведение покупки с использованием цифрового купона (скидка на товар) (номинал купона %) и программа лояльности Дисконт.

25. Проведение возврата покупки с использованием цифрового купона (скидка на товар) (но- минал - %) и программа лояльности Дисконт.

26. Проведение покупки с использованием цифрового купона (скидка на товар) (номинал - сумма) и программа лояльности Дисконт.

27. Проведение возврата покупки с использованием цифрового купона (скидка на товар) (номинал - сумма) и программа лояльности Дисконт.

28. Проведение покупки с использованием купона (скидка на документ %) и купона (скидка на товар %) и программа лояльности Дисконт.

29. Проведение покупки с использованием купона (скидка на документ $) и купона (скидка на товар $) и программа лояльности Дисконт.

30. Проведение покупки с использованием купона (скидка на документ %) и купона (скидка на товар $) и программа лояльности Дисконт.

31. Проведение покупки с использованием купона (скидка на документ $) и купона (скидка на товар %) и программа лояльности Дисконт.

32. Проведение покупки с использованием купона (скидка на товар $) и купона (скидка на товар $) и программа лояльности Дисконт. Возврат.

33. Проведение покупки с использованием купона (скидка на товар %) и купона (скидка на товар %) и программа лояльности Дисконт. Возврат.

34. Проведение покупки с использованием купона (скидка на товар $) и купона (скидка на товар %) и программа лояльности Дисконт. Возврат.

35. Программа лояльности отсутствует – проведение покупки с использованием цифровых купонов: номенклатурный, на весь документ, с автоматическим гашением, без автоматического гашения, вознаграждение %, сумма, бонус, товар.

36. Ведомая система – проведение покупки с использованием локальной программы лояльности – перечень программ лояльности обусловлен функциональностью ПО, на платформе которого производится интеграция.

37. Проверка перечня настроек, организованных для работы с Системой.

38. Осуществление настройки модуля, смены значений настроек – переподключение на данному модуле другого Участника системы.

39. Проверка процессинга активации POS.

40. Проверка процессинга деактивации POS.

41. Проверка процессинга попытки активации POS, когда данная POS уже активирована.

42. Проверка отработки процессинга «Новый клиент» - проведение первой скидки.

43. Проверка передачи информации в заголовке HTTP *User-Agent*.

Пример протокола ППИ – *Протокол тестирования интеграционного решения* [53].

Алгоритм предоставления скидки по акции

Это приложение содержит описание алгоритмов предоставления вознаграждения по акции.

1. Купон на скидку

Предоставление скидки по акции с вознаграждением вида «процент скидки на всю покупку».

Таблица Приложение 2.1. Условные обозначения

Сумма покупки.	`S`
Скидка по акции, в %.	`E%`
Скидка текущая клиента, в %.	`Et%`
Ограничение суммы, точная.	`A`
Ограничение суммы, минимальная.	`B`
Ограничение суммы, максимальная.	`C`
Итоговая сумма.	`Z`

Далее в таблицах представлены различные варианты алгоритма начисления вознаграждения в зависимости от того, какие именно ограничения (`A`, `B` и `C`) заданы в акции.

Таблица Приложение 2.2. Ограничение суммы, точная (параметр `A`).

`S < A`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.
`S = A`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`S > A`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.

Таблица Приложение 2.3. Ограничение суммы, минимальная (параметр `B`).

`S < B`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.
`S = B`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`S > B`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**

Таблица Приложение 2.4. Ограничение суммы, максимальная (параметр `C`).

`S < C`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`S = C`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`S > C`	`Z = C * (1 - E% / 100) + (S - C) * (1 - Et% / 100)`	**Предоставить скидку по купону и текущую скидку *Пользователя* на сумму свыше ограничения.**

Таблица Приложение 2.5. Ограничение суммы, минимальная и максимальная (параметры `B` и `C`).

`S < B`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.
`S = B`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`B < S < C`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`S = C`	`Z = S * (1 - E% / 100)`	**Предоставить скидку по купону.**
`S > C`	`Z = C * (1 - E% / 100) + (S - C) * (1 - Et% / 100)`	**Предоставить скидку по купону и текущую скидку *Пользователя* на сумму свыше ограничения.**

2. Купон на сумму

Предоставление скидки по акции с вознаграждением вида «сумма скидки на всю покупку».

Таблица Приложение 2.6. Условные обозначения

Сумма покупки.	`S`
Скидка по акции, в валюте покупки.	`E`
Скидка текущая клиента, в %.	`Et%`
Ограничение суммы, точная.	`A`
Ограничение суммы, минимальная.	`B`
Ограничение суммы, максимальная.	`C`
Итоговая сумма.	`Z`

Далее в таблицах представлены различные варианты алгоритма начисления вознаграждения в зависимости от того, какие именно ограничения (`A`, `B` и `C`) заданы в акции.

Таблица Приложение 2.7. Без ограничений (параметры не заданы).

`S < E`	`Z = S - E`	**Предоставить скидку по купону.**
`S = E`	`Z = S - E`	**Предоставить скидку по купону.**
`S > E`	`Z = S - E - (S - E) * (1 - Et% / 100)`	**Предоставить скидку по купону и текущую скидку *Пользователя* на сумму свыше скидки по акции.**

Таблица Приложение 2.8. Ограничение суммы, точная (параметр `A`).

`S < A`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.
`S = A`	`Z = S - E`	**Предоставить скидку по купону.**
`S > A`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.

Таблица Приложение 2.9. Ограничение суммы, минимальная (параметр `B`).

`S < B`	`Z = S * (1 - Et% / 100)`	Не предоставлять скидку по купону.
`S = B`	`Z = S - E`	**Предоставить скидку по купону.**

S > B	Z = S - E - (S - E) * (1 - Et % / 100)	**Предоставить скидку по купону и текущую скидку *Пользователя* на сумму свыше скидки по акции.**

Таблица Приложение 2.10. Ограничение суммы, максимальная (параметр C).

S < C	-	**Нет варианта.**
S = C	-	**Нет варианта.**
S > C	-	**Нет варианта.**

Таблица Приложение 2.11. Ограничение суммы, минимальная и максимальная (параметры B и C).

S < B	-	**Нет варианта.**
S = B	-	**Нет варианта.**
B < S < C	-	**Нет варианта.**
S = C	-	**Нет варианта.**
S > C	-	**Нет варианта.**

Протокол тестирования интеграционного решения

Протокол тестирования интеграционного решения

(Наименование интеграционного решения, версия интеграционного решения)

Город «____»_____20____г.

Указанным протоколом отражены бизнес-процессы и замечания к ним, возникшие в ходе демонстрации интеграционного решения «____»_____20__г. Демонстрация бизнес-процессов производилась согласно тезисов ППИ (*Приём-передаточные испытания* [45]), определенных в Техническом задании.

№	Бизнес процесс	Бизнес процесс. Детально.	Резюме	Примечание исполнителя

Исполнитель:

ФИО

Глоссарий

API
: *Application Programming Interface* - программный интерфейс клиентских приложений, использующих функции *Системы*.

ПО
: Программное обеспечение.

Система
: Программно-аппаратный комплекс, выполняющий функции, доступные через описываемый настоящим документом *API*.

Участник
: Торговое предприятие, магазин, сервис (юридическое лицо), являющееся частью Системы, и использующее технологии компании на основании заключенного договора.

Пользователь
: Пользователь *Системы*, совершающий покупки у *Участника* и обладающий накоплениями различных цифровых объектов.

Токен авторизации (token)
: Уникальный секретный идентификатор *ПО* кассового терминала (POS-системы). Каждый такой идентификатор связан с одним (и только одним) *участником* системы и точкой продажи, принадлежащей *участнику*.

Токен авторизации приложения (apptoken)
: Уникальный секретный идентификатор *ПО* интеграционного решения, которое взаимодействует с *API Системы* от имени кассового терминала (POS-системы).

ФИО
: Фамилия, имя и отчество человека (например, пользователя *Системы*).

Identifier Number (ID)
: Идентификатор объекта (например, организации-*участника* или пользователя) в учетной системе компании. Представляет собой положительное целое число.

User Identifier (GSRN)
: Идентификатор пользователя *Системы* в учетной системе компании. Представляет собой строку десятичных цифр длиной `25`.