



History. Метод получения пользовательской истории

Изменено Ирина Вахитова

в 15/05/2019 10:35

Оглавление

Запрос	Ответ																																
<p>Данный метод предназначен для получения истории покупок, начислений и списаний текущего пользователя.</p> <p>Параметры:</p> <ul style="list-style-type: none"> Метод: GET; URL: api/v1.1/History; Обязательные параметры URL: <ul style="list-style-type: none"> <code>filter.toDate</code>={Дата, до которой необходимо вывести историю} <code>filter.count</code>={Количество элементов для вывода} Дополнительные параметры URL: <ul style="list-style-type: none"> <code>filter.fromDate</code>={Дата, с которой необходимо вывести историю} <code>filter.cardId</code>={ID карты (если указан, будет получена история по указанной карте)} <code>filter.from</code>={Начиная с какого элемента будет выводиться история}. Дополнительные заголовки: Authorization. <p>Замечания:</p> <ul style="list-style-type: none"> Если не указан ни один из обязательных параметров URL, или указан только «<code>filter.count</code>», то ответ процессинга будет содержать «0» строк. Если указан только параметр URL «<code>filter.toDate</code>», то процессинг вернёт в ответе количество строк истории без детализации. Если все обязательные параметры URL указаны, то ответ процессинга будет содержать количество строк и их детализацию. Строки отсортированы от новых к старым. 	<p>Ответ процессинга содержит историю покупок, начислений и списаний авторизованного Пользователя.</p> <table border="1"> <thead> <tr> <th colspan="2">Поле "type"</th> </tr> <tr> <th>Значение</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>Purchase</td> <td>События, связанные с совершением покупки</td> </tr> <tr> <td>RewardData</td> <td>События, связанные с начислением бонусов без покупки (например, ручные начисления, начисления за привязку аккаунта в соц. сети, вступление в группу в соц. сети и т.п.)</td> </tr> <tr> <td>WithdrawData</td> <td>События, связанные со списанием бонусов без покупки (например, ручное списание бонусов, выход из группы в соц. сети и т.п.)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">Поле "withdrawType"</th> </tr> <tr> <th>Значение</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>Bonus</td> <td>Бонусы</td> </tr> <tr> <td>GiftCard</td> <td>Оплата подарочной картой</td> </tr> <tr> <td>Withdraw</td> <td>Списание бонусов на кассе, но не оплата</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">Поле "rewardType"</th> </tr> <tr> <th>Значение</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td>Bonus</td> <td>Бонусы</td> </tr> <tr> <td>Discount</td> <td>Прямая скидка</td> </tr> <tr> <td>Gift</td> <td>Дополнительные бонусы</td> </tr> <tr> <td>Charging</td> <td>Пополнение бонусной карты</td> </tr> </tbody> </table> <pre> { "data": { "allCount": 0, <!--Количество записей--> "rows": [<!--Строки--> <!--Событие в рамках покупки--> { "id": "00000000-0000-0000-0000-000000000000", <!-- ID записи в истории--> "dateTime": "yyyy-MM-ddTHH:mm:ss.sssZ", <!--Дата события--> "type": "Purchase", <!--Тип события - Покупка--> "userId": 0, <!--ID Пользователя в системе--> "identity": "string", <!--Идентификатор Пользователя (номер карты, номер телефона, ...)--> "description": "string", <!--Описание--> "location": { <!--Место события--> "id": 0, "locationId": "00000000-0000-0000-0000-000000000000", <!-- Идентификатор места--> "description": "string", <!--Описание--> "latitude": 0, <!--Широта--> "longitude": 0, <!--Долгота--> "region": { <!--Регион--> "id": 0, "name": "string", <!--Название--> "prefix": "string" <!--Префикс--> }, "city": { <!--Город--> </pre>	Поле "type"		Значение	Описание	Purchase	События, связанные с совершением покупки	RewardData	События, связанные с начислением бонусов без покупки (например, ручные начисления, начисления за привязку аккаунта в соц. сети, вступление в группу в соц. сети и т.п.)	WithdrawData	События, связанные со списанием бонусов без покупки (например, ручное списание бонусов, выход из группы в соц. сети и т.п.)	Поле "withdrawType"		Значение	Описание	Bonus	Бонусы	GiftCard	Оплата подарочной картой	Withdraw	Списание бонусов на кассе, но не оплата	Поле "rewardType"		Значение	Описание	Bonus	Бонусы	Discount	Прямая скидка	Gift	Дополнительные бонусы	Charging	Пополнение бонусной карты
Поле "type"																																	
Значение	Описание																																
Purchase	События, связанные с совершением покупки																																
RewardData	События, связанные с начислением бонусов без покупки (например, ручные начисления, начисления за привязку аккаунта в соц. сети, вступление в группу в соц. сети и т.п.)																																
WithdrawData	События, связанные со списанием бонусов без покупки (например, ручное списание бонусов, выход из группы в соц. сети и т.п.)																																
Поле "withdrawType"																																	
Значение	Описание																																
Bonus	Бонусы																																
GiftCard	Оплата подарочной картой																																
Withdraw	Списание бонусов на кассе, но не оплата																																
Поле "rewardType"																																	
Значение	Описание																																
Bonus	Бонусы																																
Discount	Прямая скидка																																
Gift	Дополнительные бонусы																																
Charging	Пополнение бонусной карты																																

```

        "id": 0,
        "regionId": 0, <!--Идентификатор региона-->
        "name": "string", <!--Название-->
        "prefix": "string" <!--Префикс-->
    },
    "street": "string", <!--Улица-->
    "house": "string", <!--Номер дома-->
    "building": "string", <!--Строение-->
    "office": "string" <!--Офис-->
},
"partnerId":
"00000000-0000-0000-0000-000000000000", <!--ID
Партнера-->
"brandId": "00000000-0000-0000-0000-000000000000",
"brand": { <!--Информация о бренде-->
    "externalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
идентификатор бренда-->
    "name": "string", <!--Название бренда-->
    "images": [ <!--Изображения-->
        {
            "fileId": "00000000-0000-0000-0000-000000000000",
            "description": "string"
        }
    ]
},
"data": {
    "externalPurchaseId":
"00_000_0000_0000000_00000000", <!--Внешний
идентификатор покупки-->
    "chequeItems": [ <!--Позиции чека-->
        {
            "description": "string", <!--Наименование-->
            "count": 1, <!--Количество-->
            "amount": 10.50 <!--Стоимость-->
        }
    ],
    "withdraws": [ <!--Списание бонусов в рамках покупки-->
    {
        "withdrawType": "string", <!--Тип списания-->
        "description": "string", <!--Описание-->
        "amount": {
            "amount": -1, <!--Сумма-->
            "currency": "string",
            "currencyInfo": { <!--Информация о валюте-->
                "id": 1, <!--ID валюты-->
                "name": "string", <!--Название валюты-->
                "externalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
идентификатор валюты-->
                "description": "string", <!--Описание-->
                "isDeleted": false, <!--Является ли валюта
архивной-->
                "nameCases": { <!--Список вариантов
написания названия-->
                    "nominative": "string", <!--В единственном числе
и именительном падеже-->
                    "genitive": "string", <!--В единственном числе и
родительном падеже-->
                    "plural": "string", <!--Во множественном числе
и именительном падеже-->
                    "abbreviation": "string" <!--Сокращенный
вариант-->
                }
            }
        }
    ]
}
}
}

```

```

    }
  ],
  "rewards": [ <!--Информация о вознаграждении-->
    {
      "offerExternalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
ID акции-->
      "rewardType": "string", <!--Тип вознаграждения-->
      "description": "string", <!--Описание-->
      "amount": {
        "amount": 1, <!--Сумма-->
        "currency": "string",
        "currencyInfo": { <!--Информация о валюте-->
          "id": 0, <!--ID валюты-->
          "name": "string", <!--Название валюты-->
          "externalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
идентификатор-->
          "description": "string", <!--Описание -->
          "isDeleted": false, <!--Является ли валюта
архивной-->
          "nameCases": { <!--Список вариантов
написания названия-->
            "nominative": "string", <!--В единственном числе
и именительном падеже-->
            "genitive": "string", <!--В единственном числе и
родительном падеже-->
            "plural": "string", <!--Во множественном числе
и именительном падеже-->
            "abbreviation": "string" <!--Сокращенный
вариант-->
          }
        }
      }
    }
  ],
  "isRefund": false, <!--Признак возврата-->
  "chequeNumber": "0", <!--Номер чека-->
  "amount": {
    "amount": 0, <!--Итоговая сумма чека-->
    "currency": "string",
    "currencyInfo": { <!--Информация о валюте-->
      "id": 0, <!--ID валюты-->
      "name": "string", <!--Название валюты-->
      "externalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
идентификатор-->
      "description": "string", <!--Описание-->
      "isDeleted": true, <!--Является ли валюта
архивной-->
      "nameCases": { <!--Варианты написания-->
        "nominative": "string", <!--В единственном числе и
именительном падеже-->
        "genitive": "string", <!--В единственном числе и
родительном падеже-->
        "plural": "string", <!--Во множественном числе и
именительном падеже-->
        "abbreviation": "string" <!--Сокращенный вариант--
>
      }
    }
  }
},
<!--Начисление без покупки-->
{

```

```

        "id": "00000000-0000-0000-0000-000000000000", <!--
ID записи в истории-->
        "dateTime": "yyyy-MM-ddTHH:mm:ss.sssZ", <!--Дата
события-->
        "type": "RewardData", <!--Тип события - начисление
бонусов без покупки-->
        "userId": 0, <!--ID Пользователя в системе-->
        "identity": "string", <!--Идентификатор Пользователя
(номер карты, номер телефона, ...)-->
        "description": "string", <!--Описание-->
        "location": null, <!--Место-->события
        "partnerId":
"00000000-0000-0000-0000-000000000000", <!--ID
Партнера-->
        "brandId": null, <!--ID бренда-->
        "brand": null, <!--Информация о бренде-->
        "data": {
            "offerExternalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
ID акции-->
            "rewardType": "string", <!--Тип вознаграждения-->
            "description": null, <!--Описание-->
            "amount": {
                "amount": 15, <!--Сумма начисленных бонусов-->
                "currency": "string",
                "currencyInfo": { <!--Информация о валюте-->
                    "id": 1, <!--ID валюты-->
                    "name": "string", <!--Название валюты-->
                    "externalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
ID валюты-->
                    "description": "string", <!--Описание валюты-->
                    "isDeleted": false, <!--Является ли валюта
архивной-->
                    "nameCases": { <!--Варианты написания
названия-->
                        "nominative": "string", <!--В единственном числе и
именительном падеже-->
                        "genitive": "string", <!--В единственном числе и
родительном падеже-->
                        "plural": "string", <!--Во множественном числе и
именительном падеже-->
                        "abbreviation": "string" <!--Сокращенный вариант-->
                    }
                }
            }
        },
<!--Списание без покупки-->
        {
            "id": "00000000-0000-0000-0000-000000000000", <!--
ID записи в истории-->
            "dateTime": "yyyy-MM-ddTHH:mm:ss.sssZ", <!--Дата
события-->
            "type": "WithdrawData", <!--Тип события – списание
бонусов без покупки-->
            "userId": 0, <!--ID Пользователя в системе-->
            "identity": "string", <!--Идентификатор Пользователя
(номер карты, номер телефона, ...)-->
            "description": "string", <!--Описание-->
            "location": null,
            "partnerId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
идентификатор Партнера-->
            "brandId": null,

```

```
"brand": null,
"data": {
  "withdrawType": "string", <!--Тип списания-->
  "description": null,
  "amount": {
    "amount": -10, <!--Сумма списанных бонусов-->
    "currency": "string",
    "currencyInfo": { <!--Информация о валюте-->
      "id": 1, <!--ID валюты-->
      "name": "string", <!--Название валюты-->
      "externalId":
"00000000-0000-0000-0000-000000000000", <!--Внешний
идентификатор-->
      "description": "string", <!--Описание-->
      "isDeleted": false, <!--Является ли валюта
архивной-->
      "nameCases": { <!--Варианты написания-->
        "nominative": "string", <!--В единственном числе и
именительном падеже-->
        "genitive": "string", <!--В единственном числе и
родительном падеже-->
        "plural": "string", <!--Во множественном числе и
именительном падеже-->
        "abbreviation": "string" <!--Сокращенный вариант--
>
      }
    }
  }
},
"result": {
  "state": "Success",
  "message": "string",
  "validationErrors": [
    {
      "field": "string",
      "errorMessages": [
        "string"
      ]
    }
  ]
}
}
```

Registration. Методы процесса регистрации

Изменено Ирина Вахитова

в 15/05/2019 10:35

Оглавление

Шаги регистрации: 3

Шаги регистрации:

- [Ввод номера телефона и принятие оферты](#)
- [Ввод кода подтверждения номера телефона](#)
- [Установка пароля](#)
- [Привязка электронной почты](#)
- [Заполнение анкеты](#)
- [Привязка карты](#)
- [Завершение регистрации](#)

Порядок шагов и их обязательность определяются бизнес-требованиями в программе лояльности. Чаще всего привязка электронной почты и номера карты являются необязательными шагами.

Дополнительно в этом разделе:

- [Повторная отправка кода подтверждения номера телефона](#)
- [Повторный ввод номер телефона](#)
- [Повторная отправка кода подтверждения электронной почты](#)
- [Повторный ввод адреса электронной почты](#)
- [Продолжение регистрации](#)
- [Сообщение об ошибке в анкете](#)
- [Системная ошибка](#)

Ввод номера телефона и принятие оферты

Ввод номера телефона, по которому будет производиться регистрация, и принятие условий ПЛ (+ согласие на обработку персональных данных) происходит на одном экране.

Пользователь вводит номер телефона и соглашается с условиями ПЛ (договор оферты и т.д.), отмечая соответствующие поля.

Запрос	Ответ
<p>Запрос на начало регистрации BeginRegistration</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/Registration/BeginRegistration • Формат тела запроса: {"login":"номер_телефона","password":""} <p>Тело запроса должно содержать введенный пользователем номер телефона.</p> <p>{"login":"79123487979","password":""}</p>	<p>Ответ процессинга содержит авторизационный токен (параметр "authToken"), который должен использоваться в последующих запросах, и код для подтверждения.</p> <pre>{ "data": { "state": "Success", "errorMessage": null, "authToken": "MeRVIIOAX_qCxCGLFww0uKr2" }, "result": { "state": "Success", "message": null, "validationErrors": null } }</pre>
<p>Запрос обязательных шагов регистрации actions.</p> <p>Запрос необходим для получения списка шагов регистрации и их статуса Завершен/Не завершен.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: GET • URL: /api/v1/user/actions • Дополнительные заголовки: Authorization <p>Заголовок запроса должен содержать поле Authorization, содержащее ранее полученный авторизационный токен.</p> <p>Поле Authorization должно иметь формат Authorization: Bearer Авторизационный_токен.</p> <p>Authorization: Bearer MeRVIIOAX_qCxCGLFww0uKr2</p>	<p>Ответ процессинга содержит список шагов регистрации и их статус.</p> <p>Список шагов:</p> <ol style="list-style-type: none"> 1. AcceptTenderOffer – Принятие договора оферты; 2. ChangePhone – Установка номера телефона; 3. ChangeEmail – Установка адреса электронной почты; 4. PasswordRequired – Установка пароля; 5. Questions – Заполнение анкеты; 6. AssignCard – Привязка карты. <pre>{ "data": { "actions": [</pre>

```

"userActionType": "AcceptTenderOffer", <!--
Наименование шага регистрации-->
"actionState": "Required",
"isDone": false <!--Статус шага (true – завершен/ false
– не завершён)-->
},
...
{
"userActionType": "AssignCard",
"actionState": "Required",
"isDone": false
}
}
},
"result": {
"state": "Success",
"message": null,
"validationErrors": null
}
}

```

Повторное прохождение завершенных шагов регистрации не допускается. В случае если на процессинг повторно поступит запрос, связанный с уже завершенным шагом, то процессинг ответит на него протокольной ошибкой.

Запрос принятия договора оферты **AcceptTenderOffer**

Параметры:

- Метод: POST
- URL: /api/v1/User/AcceptTenderOffer
- Дополнительные заголовки: Authorization

Ответ процессинга содержит только статус обработки запроса.

Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос **actions**, в ответе на который шаг **AcceptTenderOffer** должен иметь статус **"isDone": true**, означающий завершение шага **Принятие оферты**, иначе шаг считается незавершенным.

Ввод кода подтверждения номера телефона

На данном шаге пользователь должен провалидировать указанный на предыдущем шаге номера телефона, введя в соответствующее поле полученный код подтверждения.

Запрос	Ответ
<p>Запрос PhoneNumber необходим для получения информации о подтверждаемом номере телефона.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: GET • URL: /api/v1/User/PhoneNumber/ • Дополнительные заголовки: Authorization 	<p>Ответ процессинга содержит поле phoneNumber, имеющее значение вида «***1234» - последние 4 цифры введенного пользователем номера телефона. Данное значение отображается пользователю.</p> <pre> { "data": { "currentPhoneNumber": "", "newPhoneNumber": { "phoneNumber": "***1234" } }, "confirmCodeLength": 6 }, "result": { "state": "Success", "message": null, "validationErrors": null } } </pre>

Пользователь вводит код подтверждения, полученный на указанный в начале регистрации номер телефона, и нажимает кнопку **Далее**.

Введенный код подтверждения должен быть передан на процессинг в запросе **PhoneNumber/Confirm**.

Результатом такого запроса является подтверждение номера телефона.

Ответ процессинга содержит обновленный авторизационный токен. В последующих запросах в

Параметры:

- Метод: POST
- URL: /api/v1/User/PhoneNumber/Confirm
- Дополнительные заголовки: Authorization
- Формат тела запроса: {confirmCode: "код_подтверждения"}

Пример: {confirmCode: "384308"}

заголовке **Authorization** должен быть указан последний полученный авторизационный токен.

Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос **actions**, в ответе на который шаг **ChangePhone** должен иметь статус **"isDone": true**, означающий завершение шага **Установка номера телефона**, иначе шаг считается незавершенным.

Установка пароля

На данном шаге устанавливается пароль от Личного кабинета ПЛ. Пользователю отображаются поля для ввода пароля и его подтверждения повторным вводом.

Запрос	Ответ
<p>После ввода пароля пользователь нажимает кнопку Далее. На процессинг должен быть отправлен POST-запрос Set, в котором передаётся введенный пользователем пароль.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/User/Password/Set • Дополнительные заголовки: Authorization • Формат тела запроса: {"password":"Пароль"} <p>Пример: {"password":"111111"}</p>	<p>Результатом выполнения данного запроса является установка пароля для доступа к Личному кабинету ПЛ.</p> <p>Ответ процессинга содержит обновленный авторизационный токен. В последующих запросах в заголовке Authorization должен быть указан последний полученный авторизационный токен.</p> <p>Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос actions, в ответе на который шаг PasswordRequired должен иметь статус "isDone": true, означающий завершение шага Установка пароля, иначе шаг считается незавершенным.</p>

Привязка электронной почты

Запрос	Ответ
<p>При переходе на данный этап регистрации (при загрузке страницы) на процессинг должен быть отправлен GET-запрос Email для получения информации о текущем установленном для пользователя адресе электронной почты.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: GET • URL: /api/v1/User/Email/ • Дополнительные заголовки: Authorization <p>Пользователь вводит адрес своей электронной почты и нажимает кнопку Далее. На процессинг должен быть отправлен POST-запрос Email, в котором передается введенный пользователем адрес электронной почты.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/User/Email/ • Дополнительные заголовки: Authorization • Формат тела запроса: {email: "e-mail"} <p>Пример: {email: "test@loymax.ru"}</p> <p>В случае выбора пользователем верификации посредством кода он вводит полученный на указанный адрес электронной почты код подтверждения и нажимает кнопку Далее. На процессинг должен быть отправлен POST-запрос Email/Confirm, в котором передаётся введенный пользователем код подтверждения.</p>	<pre>{ "data": { "currentEmail": null, "newEmail": { "email": null }, "confirmCodeLength": 6 }, "result": { "state": "Success", "message": null, "validationErrors": null } }</pre> <p>Ответ процессинга содержит статус обработки запроса. В случае его успешной обработки на указанный пользователем e-mail отправляется письмо с кодом подтверждения и ссылкой для привязки, а на странице отображается поле для ввода кода подтверждения.</p> <p>Ответ процессинга содержит статус обработки запроса.</p> <p>Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос actions, в ответе на который шаг ChangeEmail должен иметь статус "isDone": true, означающий завершение шага</p>

Параметры:

- Метод: POST
- URL: /api/v1/User/Email/Confirm
- Дополнительные заголовки: Authorization
- Формат тела запроса: {confirmCode: "Код_подтверждения"}

Пример: {confirmCode: "182341"}

Привязка электронной почты, иначе шаг считается незавершенным.

Заполнение анкеты

Запрос	Ответ
--------	-------

На процессинг должен быть отправлен GET-запрос **Questions**, необходимый для получения списка вопросов анкеты.

Параметры:

- Метод: GET
- URL: /api/v1/User/Questions
- Параметры URL: filter.onlyRequired=true
- Дополнительные заголовки: Authorization

Ответ процессинга содержит дерево вопросов анкеты в формате JSON.

```
{
  "data": {
    "questions": [
      {
        "nodeType": "Header",
        "displayName": "Личные данные",
        "description": null,
        "logicalName": null,
        "value": null,
        "readOnly": false,
        "required": null,
        "id": null,
        "groupID": null,
        "selected": null,
        "maxLength": null,
        "fieldType": null,
        "children": [
          {
            "nodeType": "Question",
            "displayName": "Фамилия", <!--Отображаемое имя
            вопроса-->
            "description": null,
            "logicalName": "LastName",
            "value": null,
            "readOnly": false,
            "required": true,
            "id": 2, //ID вопроса
            "groupID": null, //ID группы
            "selected": false,
            "maxLength": null,
            "fieldType": "String",
            "children": null
          }, ...
        ]
      }
    ]
  },
  "result": {
    "state": "Success",
    "message": null,
    "validationErrors": null
  }
}
```

После заполнения анкеты пользователь нажимает кнопку **Далее** и на процессинг отправляется POST-запрос **Answers**, в котором передаются ответы на вопросы анкеты.

Параметры:

- Метод: POST
- URL: /api/v1/User/Answers

Ответ процессинга содержит статус обработки запроса.

Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос **actions**, в ответе на который шаг **Questions** должен иметь статус **"isDone": true**, означающий завершение шага **Заполнение анкеты**, иначе шаг считается незавершенным.

- Параметры URL: filter.onlyRequired=true
- Дополнительные заголовки: Authorization
- Формат тела запроса:

```

{"answers":[
  {"groupID":id_группы,
   "id":id_вопроса,
   "value":"Ответ_введенный_пользователем",
   "selected":Ответ_выбран_из_предложенных(true)/
   Ответ_введен_пользователем(false)
  },...
]}

"value":"Ответ_введенный_пользователем"
передается только в случае, если
"selected":Ответ_выбран_из_предложенных(true)
    
```

Пример:

```

{"answers":[
  {"groupID":null,
   "id":2,
   "value":"1",
   "selected":false},
  {"groupID":5,
   "id":1,
   "value":null,
   "selected":true}
]}
    
```

Привязка карты

Запрос	Ответ
<p>На процессинг отправляется запрос выпуска виртуальной карты. Данный запрос необходим для получения конфигурации системы: доступен ли выпуск виртуальной карты</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: GET • URL: /api/v1/Cards/EmitVirtual • Дополнительные заголовки: Authorization <p>Если пользователь вводит номер физической карты и нажимает кнопку Далее, на процессинг должен отправляться POST-запрос для привязки физической карты.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/Cards/Set • Дополнительные заголовки: Authorization • Формат тела запроса: {"cardNumber":"Номер_карты"} <p>Пример: {"cardNumber":"9116580844911085"}</p> <p>Если у пользователя отсутствует физическая карта, а настройки системы позволяют выдать виртуальную карту, то пользователь нажимает кнопку У меня нет карты. В таком случае отправляется запрос EmitVirtual, необходимый для выпуска и привязки виртуальной карты.</p> <p>Параметры:</p>	<pre> { "data": { "currentCountOfVirtualCards": 0, "isVirtualCardEmissionAllowed": true } } </pre> <p>Параметр isVirtualCardEmissionAllowed имеет значение false, если выпуск виртуальных карт запрещен и true, если выпуск виртуальной карты возможен. В случае если данный параметр имеет значение true, на этапе привязки физической карты возможно отображение кнопки У меня нет карты, по нажатию на которую производится выпуск виртуальной карты.</p> <p>В ответе процессинга будет содержаться статус обработки запроса.</p> <p>Ответ процессинга содержит статус обработки запроса.</p> <p>Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос actions, в ответе на который шаг AssignCard должен иметь статус "isDone": true, означающий завершение шага Привязка карты, иначе шаг считается незавершенным.</p>

- Метод: PUT
- URL: /api/v1/Cards/EmitVirtual
- Дополнительные заголовки: Authorization

Завершение регистрации

Запрос	Ответ
<p>После завершения всех шагов регистрации обязательно требуется отправить запрос на завершение регистрации TryFinishRegistration.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/Registration/TryFinishRegistration • Дополнительные заголовки: Authorization 	<p>Ответ процессинга содержит статус процедуры регистрации и новый авторизационный токен.</p> <pre>{ "data": { "registrationCompleted": true, <!--Статус процедуры регистрации--> "authToken": "gyDoRSvKXrmpeQlhbe1gG7RM7lDbBMg9acaSosC5m5SwW9ygrlBPQw" }, <!--Авторизационный токен--> "result": { "state": "Success", "message": null, "validationErrors": null } }</pre>

Повторная отправка кода подтверждения номера телефона

Запрос	Ответ
<p>Если пользователю необходима повторная отправка кода подтверждения на указанный номер телефона, то пользователь нажимает на соответствующую кнопку, после чего на процессинг необходимо отправить запрос SendConfirmCode.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/User/PhoneNumber/SendConfirmCode • Дополнительные заголовки: Authorization 	<p>Ответ процессинга будет содержать статус обработки запроса.</p>

Повторный ввод номер телефона

Запрос	Ответ
<p>Если пользователь ввел неверный номер телефона, то он нажимает кнопку Повторный ввод номера телефона и в открывшемся окне указывает верный номер телефона и нажимает кнопку Отправить. На процессинг отправляется POST-запрос PhoneNumber.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/User/PhoneNumber/ • Дополнительные заголовки: Authorization • Формат тела запроса: {"phoneNumber": "Номер_телефона"} <p>Пример: {"phoneNumber": "75645615615"}</p>	<p>Ответ процессинга будет содержать статус обработки запроса и код для подтверждения.</p>

Повторная отправка кода подтверждения электронной почты

Запрос	Ответ
<p>Если пользователю необходима повторная отправка кода подтверждения на указанный адрес электронной почты, то пользователь нажимает на соответствующую кнопку, после чего на процессинг необходимо отправить запрос SendConfirmCode.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST 	<p>Ответ процессинга будет содержать статус обработки запроса.</p>

- URL: /api/v1/User/Email/SendConfirmCode
- Дополнительные заголовки: Authorization

Повторный ввод адреса электронной почты

Запрос	Ответ
<p>Если пользователь ввел неверный адрес электронной почты, то он нажимает кнопку Повторный ввод e-mail и в открывшемся окне указывает верный e-mail и нажимает кнопку Отправить. На процессинг отправляется POST-запрос Email.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/User/Email/ • Дополнительные заголовки: Authorization • Формат тела запроса: {"email": "E-mail "} <p>Пример: {"email": "test@loymax.ru"}</p>	<p>Ответ процессинга будет содержать статус обработки запроса.</p>

Продолжение регистрации

Пользователь может прервать регистрацию на любом из шагов, и продолжить её позже.

Запрос	Ответ
<p>Для продолжения регистрации отправляется запрос BeginRegistration.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/Registration/BeginRegistration • Формат тела запроса: {"password": "", "login": "Номер_телефона"} <p>Пример: {"password": "", "login": "79123487979"}</p> <p>Для пользователя необходимо вывести форму ввода пароля. Пользователь вводит пароль и нажимает Регистрация. На процессинг повторно отправляется запрос BeginRegistration, в котором передается пароль и номер телефона.</p> <p>Параметры:</p> <ul style="list-style-type: none"> • Метод: POST • URL: /api/v1/Registration/BeginRegistration • Формат тела запроса: {"login": "Номер_телефона", "password": "Пароль"} <p>Пример: {"login": "79123487979", "password": "352678"}</p>	<p>В ответе процессинг вернёт протокольную ошибку Требуется пароль и формат ответа PasswordRequired.</p> <pre>"data": { "state": "PasswordRequired", <!--Формат ответа--> "errorMessage": "Требуется пароль.", <!--Текст ошибки--> } > "authToken": null }</pre> <p>Ответ процессинга содержит авторизационный токен (параметр authToken: "Авторизационный_токен"), который должен использоваться в последующих запросах.</p> <p>Перед переходом на следующий шаг регистрации на процессинг может быть отправлен запрос actions для получения информации о завершённых шагах регистрации.</p>

Сообщение об ошибке в анкете

В случае если при заполнении анкеты возникла ошибка, процессинг вернёт протокольную ошибку с указанием идентификатора вопроса анкеты, с которым связана ошибка.

```
"data": {
  "errors": [
    {
      "idQuestion": "7", <!--id вопроса-->
      "errors": [
        "Не допускается участие в программе лиц младше 18 лет" <!--Текст ошибки-->
      ]
    }
  ]
}
```

Системная ошибка

Если в ходе регистрации возникла системная ошибка на стороне процессинга, то ответ процессинга будет иметь статус **Fail**.

```
{
  "result": {
    "state": "Fail", <!--Формат ответа-->
    "message": "Идентификатор ошибки: 90078f44ee954cb39f3d4e8c26898da2", <!--Текст ошибки-->
    "validationErrors": null
  }
}
```

В таком случае необходимо направить заявку на support@loymax.ru с описанием шагов, повлекших ошибку, а также вводимые идентификаторы пользователей, на которых она проявилась.{{/id}}
)))

Регистрация по номеру телефона с выпуском виртуальной и физической карт

Изменено Александра Коваленко

в 21/05/2019 12:54

Оглавление

1. Ввод номера телефона и принятие оферты	3
2. Ввод кода подтверждения номера телефона	4
3. Установка пароля, метод Password/Set	4
4. Заполнение анкеты	4
5. Привязка карт	5
6. Завершение регистрации	5

Регистрация клиента в программе лояльности - Регистрация по номеру телефона с выпуском виртуальной и физической карт

Для регистрации клиента в ПЛ по номеру телефона с выпуском виртуальной и физической карт необходимо настроить параметры Личного кабинета через интерфейс АРМ в разделе **Панель администратора - Конфигурации**. Обязательно должны быть включены следующие параметры:

Параметр	Значение	
EmitVirtualCard	1	
MainCardAssigningRequired	1 или 2*	
RegistrationProviderAnonymouse	1	
RegistrationProviderPhone	1	*При установке значения 2 рассматривается процесс

Подробное описание правил конфигурирования выпуска карт читайте [здесь](#).

Регистрация клиента по номеру телефона с выпуском виртуальной и физической карт происходит в следующем порядке:

1. На кассе магазина клиент получает карту ПЛ, которую ему необходимо будет активировать во время регистрации.
2. Клиент заходит на сайт или в мобильное приложение партнера ПЛ и проходит процедуру регистрации по номеру телефона в соответствующем разделе:
 1. [Вводит номер телефона](#).
 2. [Соглашается с условиями ПЛ](#) (в том числе принимает условия договора публичной оферты).
 3. [Вводит код подтверждения](#), отправленный на номер телефона, указанный ранее.
 4. [Устанавливает пароль](#).
 5. [Заполняет анкету](#).
 6. [Привязывает физическую карту, выпускается виртуальная карта](#).
 7. [Завершает регистрацию](#).

1. Ввод номера телефона и принятие оферты

Пользователь вводит номер телефона для регистрации в ПЛ и соглашается с условиями ПЛ (договор оферты и т.д.), отмечая соответствующие поля.

.

Запрос на начало регистрации, метод [BeginRegistration](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/Registration/BeginRegistration

Тело запроса содержит номер телефона, введенный пользователем.

Запрос обязательных шагов регистрации, метод [actions](#)

Запрос

Параметры:

- Метод: GET
- URL: /api/v1/user/actions
- Дополнительные заголовки: Authorization

Запрос необходим для получения списка шагов регистрации и их статуса.

В данном запросе поле **Authorization** должно содержать авторизационный токен, полученный в ответе на запрос **BeginRegistration**.

.

Запрос принятия договора оферты, метод [AcceptTenderOffer](#)

Запрос

Параметры:

- Метод: POST

Регистрация клиента в программе лояльности - Регистрация по номеру телефона с выпуском виртуальной и физической карт

- URL: /api/v1/User/AcceptTenderOffer
- Дополнительные заголовки: Authorization

2. Ввод кода подтверждения номера телефона

Пользователь подтверждает указанный на предыдущем шаге номера телефона, введя в соответствующее поле полученный код подтверждения.

Запрос на получение информации о подтверждаемом номере телефона, метод [PhoneNumber](#)

Запрос

Параметры:

- Метод: GET
- URL: /api/v1/User/PhoneNumber/
- Дополнительные заголовки: Authorization

Запрос необходим для получения информации о подтверждаемом номере телефона пользователя, который был отправлен в запросе **E**

Пользователь вводит код подтверждения, полученный на указанный в начале регистрации номер телефона, и нажимает кнопку **Далее**.

Запрос на подтверждение номера телефона, метод [Confirm](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/User/PhoneNumber/Confirm

С помощью данного запроса на процессинг передается код подтверждения, введенный пользователем для подтверждения номера телефона.

3. Установка пароля, метод [Password/Set](#)

На данном шаге устанавливается пароль от Личного кабинета ПЛ. Пользователю отображаются поля для ввода пароля и его подтверждения повторным вводом.

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/User/Password/Set
- Дополнительные заголовки: Authorization

В запросе передаётся введенный пользователем пароль.

4. Заполнение анкеты

На данном шаге пользователь отвечает на вопросы анкеты.

Запрос на получение списка вопросов анкеты, метод [Questions](#)

Запрос

Параметры:

- Метод: GET
- URL: /api/v1/User/Questions
- Параметры URL: filter.onlyRequired=true
- Дополнительные заголовки: Authorization

Запрос необходим для получения списка вопросов анкеты.

Регистрация клиента в программе лояльности - Регистрация по номеру телефона с выпуском виртуальной и физической карт

Запрос на передачу ответов пользователя, метод [Answers](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/User/Answers
- Параметры URL: filter.onlyRequired=true
- Дополнительные заголовки: Authorization

В запросе передаются ответы на вопросы анкеты.

5. Привязка карт

На данном шаге пользователь привязывает к аккаунту физическую карту ПЛ, виртуальная карта выдается по умолчанию.

Запрос на привязку физической карты, метод [Cards/Set](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/Cards/Set
- Дополнительные заголовки: Authorization

С помощью данного запроса на процессинг передается номер физической карты, введенный пользователем.

Запрос на выдачу виртуальной карты, метод [EmitVirtual](#)

Запрос

Запрос предназначен для выдачи виртуальной карты.

Параметры:

- Метод: PUT
- URL: /api/v1/Cards/EmitVirtual
- Дополнительные заголовки: Authorization

Ответ

Ответ содержит информацию о статусе обработки запроса.

Результатом выполнения запроса является выпуск виртуальной карты и её привязка к пользователю.

6. Завершение регистрации

Пользователь завершает регистрацию.

Запрос на завершение регистрации, метод [TryFinishRegistration](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/Registration/TryFinishRegistration
- Дополнительные заголовки: Authorization

С помощью данного запроса происходит завершение регистрации клиента.

Регистрация по номеру карты с выпуском виртуальной карты и привязкой телефона

Изменено Александра Коваленко

в 21/05/2019 12:54

Оглавление

1. Ввод номера карты и принятие оферты	3
2. Привязка номера телефона	4
3. Установка пароля, метод Password/Set	4
4. Заполнение анкеты	4
5. Завершение регистрации, метод TryFinishRegistration	5

Регистрация клиента в программе лояльности - Регистрация по номеру карты с выпуском виртуальной карты и привязкой телефона

Для регистрации клиента в ПЛ по номеру карты с выпуском виртуальной карты и привязкой телефона необходимо настроить параметры Личного кабинета через интерфейс АРМ в разделе **Панель администратора - Конфигурации**. Обязательно должны быть включены следующие параметры:

Параметр	Значение	
EmitVirtualCard	1 или 2*	
MainCardAssigningRequired	1	
RegistrationProviderAnonymous	1	
RegistrationProviderCard	1	*При установке значения ПЛ, включающий в себя

Подробное описание правил конфигурирования выпуска карт читайте [здесь](#).

Регистрация по номеру карты с выпуском виртуальной карты и привязкой телефона происходит в следующем порядке:

1. На кассе магазина клиент получает карту ПЛ, номер которой он будет вводить при регистрации.
2. Клиент заходит на сайт или в мобильное приложение партнера ПЛ и проходит процедуру регистрации по номеру карты в соответствующем разделе:
 1. [Вводит номер физической карты](#).
 2. [Соглашается с условиями ПЛ](#) (в том числе принимает условия договора публичной оферты).
 3. [Создается виртуальная карта](#).
 4. [Привязывает к аккаунту номер телефона](#).
 5. [Вводит код подтверждения](#), отправленный на номер телефона, указанный ранее.
 6. [Устанавливает пароль](#).
 7. [Заполняет анкету](#).
 8. [Завершает регистрацию](#).

1. Ввод номера карты и принятие оферты

Пользователь вводит номер физической карты для регистрации в ПЛ и соглашается с условиями ПЛ (договор оферты и т.д.), отмечая соответствующие поля.

.

Запрос на начало регистрации - активация карты, метод [BeginRegistration](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/Registration/BeginRegistration

С помощью данного запроса на процессинг передается номер физической карты, введенный пользователем

Запрос обязательных шагов регистрации, метод [actions](#)

Запрос

Параметры:

- Метод: GET
- URL: /api/v1/user/actions
- Дополнительные заголовки: Authorization

Данный запрос позволяет получить список шагов регистрации пользователя.

Поле **Authorization** должно содержать авторизационный токен, полученный в ответе на запрос **BeginRegistration**.

.

Запрос принятия договора оферты, метод [AcceptTenderOffer](#)

Запрос

Параметры:

- Метод: POST

Регистрация клиента в программе лояльности - Регистрация по номеру карты с выпуском виртуальной карты и привязкой телефона

- URL: /api/v1/User/AcceptTenderOffer
- Дополнительные заголовки: Authorization

Запрос принятия договора оферты.

После принятия клиентом условий договора публичной оферты при значении параметра EmitVirtualCard 1 автоматически создается виртуальная карта и привязывается к профилю клиента.

•

Запрос на выдачу виртуальной карты, метод [EmitVirtual](#)

Запрос

Запрос предназначен для выдачи виртуальной карты.

Параметры:

- Метод: PUT
- URL: /api/v1/Cards/EmitVirtual
- Дополнительные заголовки: Authorization

2. Привязка номера телефона

Пользователь вводит номер своего телефона, после чего получает по SMS код подтверждения и его введением подтверждает привязываемый номер телефона.

Запрос на привязку номера телефона, метод [PhoneNumber](#)

Запрос

Запрос предназначен для привязки номера телефона пользователя.

Параметры:

- Метод: POST
- URL: /api/v1/User/PhoneNumber
- Дополнительные заголовки: Authorization

•

Запрос на ввод кода подтверждения, метод [Confirm](#)

Запрос

С помощью данного запроса на процессинг отправляется вводимый пользователем код подтверждения, отправленный пользователю на [PhoneNumber](#).

3. Установка пароля, метод [Password/Set](#)

На данном шаге устанавливается пароль от Личного кабинета ПЛ. Пользователю отображаются поля для ввода пароля и его подтверждения повторным вводом.

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/User/Password/Set
- Дополнительные заголовки: Authorization

С помощью данного запроса на процессинг передается пароль, заданный пользователем.

4. Заполнение анкеты

На данном шаге пользователь отвечает на вопросы анкеты.

Запрос на получение списка вопросов анкеты, метод [Questions](#)

Запрос

Параметры:

- Метод: GET
- URL: /api/v1/User/Questions
- Параметры URL: filter.onlyRequired=true
- Дополнительные заголовки: Authorization

Запрос позволяет получить список вопросов анкеты.

Запрос на передачу ответов пользователя, метод [Answers](#)

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/User/Answers
- Параметры URL: filter.onlyRequired=true
- Дополнительные заголовки: Authorization

С помощью данного запроса на процессинг передаются ответы пользователя на вопросы анкеты.

5. Завершение регистрации, метод [TryFinishRegistration](#)

Пользователь завершает регистрацию.

Запрос

Параметры:

- Метод: POST
- URL: /api/v1/Registration/TryFinishRegistration
- Дополнительные заголовки: Authorization

После завершения всех шагов регистрации обязательно требуется отправить запрос на завершение регистрации.

Покупка с использованием API

Изменено Ирина Вахитова

в 15/05/2019 10:38

Оглавление

Примеры запросов и ответов при совершении покупки через интернет-магазин	3
Расчет скидки и доступной суммы оплаты в бонусах	3
Оплата бонусами	4
Расчет и начисление бонусов	6

При совершении покупок в интернет-магазине запросы и ответы в формате **.json** проходят через **PublicApi**. Для работы с командами интернет-магазинов используется контроллер - **Processing**, включающий в себя соответствующие методы.

Для проведения покупок через PublicApi необходимо обеспечить идентификацию партнера и его устройств, а также получить доступ к данным пользователя. Для этого нужно авторизоваться в **SystemApi** под учетной записью администратора (ссылка вида <http://localhost/api/swagger/ui/index>), и затем с помощью методов SystemApi:

1. Создать **OAuth-приложение** для обеспечения идентификации партнера и его устройств. Подробнее о создании OAuth-приложений можно прочитать в соответствующем [разделе](#).
2. Получить **токен пользователя** для доступа к данным счета и проведению операций по начислению/ списанию бонусов. Информацию об использовании и получении токенов можно найти в разделах [Использование токенов](#) и [Интеграция сторонних приложений](#).

При направлении запросов в PublicApi для проведения покупки необходимо соблюдать следующую последовательность использования методов:

1. **Calculate** - расчет скидки и доступной суммы оплаты в бонусах.
2. **Available** - получение максимальной суммы оплаты.
3. **Payment** - оплата бонусами.
4. **Bonus** - расчет и начисление бонусов, расчет скидки.
5. **Refund** - возврат по операциям.
6. **Confirm** - подтверждение покупки.
7. **Cancel** - отмена покупки.
8. **Balance** - получение баланса карты.

Подробнее о методах работы с процессингом можно прочитать в [соответствующем разделе](#).

Примеры запросов и ответов при совершении покупки через интернет-магазин

- [Расчет скидки и доступной суммы оплаты в бонусах](#)
- [Оплата бонусами](#)
- [Расчет и начисление бонусов](#)
- [Подтверждение покупки](#)

Расчет скидки и доступной суммы оплаты в бонусах

Метод **Calculate**

URL: <https://api-test.loymax.net/api/v1/Processing/Purchases/321674212224E23EA7242ED28/Calculate>

В запросе передается информация о цене товара по каталогу (**price**), сумме внешней/кассовой скидки (**discount**), сумме позиции за вычетом прямой партнерской скидки (**amount**).

```
{
  "operationId": "ef59dc2ada294aa285fdbc1e1e05d377",
  "operationDate": "2019-04-09T07:24:26.628Z",
  "cashier": "string",
  "cheque": {
    "number": "string",
    "date": "2019-04-09T07:24:26.628Z",
    "lines": [
      {
        "position": 1,
        "amount": 85,
        "goodsId": "0036961",
        "barcode": "8765653477",
        "quantity": 1,
        "cashback": 0,
        "discount": 15,

```

```

    "name": "Товар100",
    "price": 100,
    "correction": 0
  }
]
},
"coupons": [
  {
    "number": "string"
  }
]
}

```

Ответ содержит информацию о размере скидки, предоставленной на переданный чек (**discount**), а также максимально возможное количество бонусов, доступное для оплаты покупки (**availableBonusAmount**).

```

{
  "data": [
    {
      "availableAmount": 38.25,
      "availableBonusAmount": 38.25,
      "transactionId": null,
      "deviceLogicalId": null,
      "operationId": "string",
      "code": "Ok",
      "errorMessage": null,
      "cashierMessage": null,
      "chequeMessage": null,
      "cheque": {
        "totalAmount": 76.5,
        "totalDiscount": 23.5,
        "totalCashback": 0,
        "lines": [
          {
            "payAmount": null,
            "message": null,
            "position": 1,
            "amount": 76.5,
            "goodsId": null,
            "barcode": "8765653477",
            "quantity": 1,
            "cashback": null,
            "discount": 23.5
          }
        ]
      }
    }
  ],
  "result": {
    "state": "Success",
    "message": null,
    "validationErrors": null
  }
}

```

Оплата бонусами

Метод **Payment**

<https://api-test.loymax.net/api/v1/Processing/Purchases/321674212224E23EA7242E/Bonus/Payment>

Запрос содержит информацию о сумме позиции до списания бонусов, но с учетом предоставленных скидок (**amount** в массиве **lines**), сумме скидок (**discount**), сумме списываемых бонусов (**amount**).

```
{
  "operationId": "ef59dc2ada294aa285fdb1e1e05d378",
  "operationDate": "2019-04-09T07:24:26.628Z",
  "cashier": "string",
  "cheque": {
    "number": "string",
    "date": "2019-04-09T07:24:26.628Z",
    "lines": [
      {
        "position": 1,
        "amount": 76.5,
        "goodsId": "0036961",
        "barcode": "8765653477",
        "quantity": 1,
        "cashback": 0,
        "discount": 23.5,
        "name": "string",
        "price": 100,
        "correction": 0
      }
    ]
  },
  "coupons": [
    {
      "number": "string"
    }
  ],
  "amount": 36.5,
  "pinCode": "string"
}
```

Ответ процессинга содержит информацию об операции оплаты и статус её обработки.

```
{
  "data": [
    {
      "amount": 36.5,
      "bonusAmount": 36.5,
      "transactionId": 1011575139,
      "deviceLogicalId": null,
      "operationId": "string11",
      "code": "OK",
      "errorMessage": null,
      "cashierMessage": null,
      "chequeMessage": null,
      "cheque": {
        "totalAmount": 76.5,
        "totalDiscount": 23.5,
        "totalCashback": 0,
        "lines": [
          {
            "payAmount": 36.5,
            "message": null,
            "position": 1,
            "amount": 76.5,
            "goodsId": null,
            "barcode": "8765653477",
            "quantity": 1,

```

```

        "cashback": null,
        "discount": 23.5
    }
  ]
}
},
],
"result": {
  "state": "Success",
  "message": null,
  "validationErrors": null
}
}

```

Расчет и начисление бонусов

Метод **Bonus**

URL: <https://api-test.loymax.net/api/v1/Processing/Purchases/321674212224E23EA7242ED28/Bonus>

Запрос содержит информацию о цене товара после применения скидок из предыдущего запроса, сумму списываемых бонусов, а также сумму к оплате с учетом скидок и списываемых бонусов.

```

{
  "operationId": "ef59dc2ada294aa285fdb1e1e05d379",
  "operationDate": "2019-04-09T07:24:26.628Z",
  "cashier": "string",
  "cheque": {
    "number": "string",
    "date": "2019-04-09T07:24:26.628Z",
    "lines": [
      {
        "position": 1,
        "amount": 76.5,
        "goodsId": "0036961",
        "barcode": "8765653477",
        "quantity": 1,
        "cashback": 0,
        "name": "string",
        "price": 0,
        "correction": 0
      }
    ]
  },
  "coupons": [
    {
      "number": "string"
    }
  ],
  "pays": [
    {
      "type": "Card",
      "amount": 40,
      "details": "string"
    },
    {
      "type": "Loymax",
      "amount": 36.5,
      "details": "string"
    }
  ]
}

```

Ответ содержит код ответа процессинга.

```
{
  "data": [
    {
      "transactionId": 1011575140,
      "deviceLogicalId": null,
      "operationId": "string112",
      "code": "Ok",
      "errorMessage": null,
      "cashierMessage": null,
      "chequeMessage": null,
      "cheque": {
        "totalAmount": 76.5,
        "totalDiscount": 23.5,
        "totalCashback": 0,
        "lines": [
          {
            "payAmount": null,
            "message": null,
            "position": 1,
            "amount": 76.5,
            "goodId": null,
            "barcode": "8765653477",
            "quantity": 1,
            "cashback": 0,
            "discount": 23.5
          }
        ]
      }
    ]
  ],
  "result": {
    "state": "Success",
    "message": null,
    "validationErrors": null
  }
}
```

Подтверждение покупки

Метод **Confirm**

URL: <https://api-test.loymax.net/api/v1/Processing/Purchases/321674212224E23EA7242ED28/Confirm>

Запрос содержит номер подтверждаемой операции и ее дату.

```
{
  "operationId": "ef59dc2ada294aa285fdb1e1e05d380",
  "operationDate": "2019-04-09T07:24:26.628Z"
}
```

Ответ содержит информацию о статусе обработки запроса.

```
{
  "data": [
    {
      "transactionId": null,
      "deviceLogicalId": null,
      "operationId": "string22",
      "code": "Ok",
      "errorMessage": null,

```

```
"cashierMessage": null,  
"chequeMessage": null,  
"cheque": null  
}  
],  
"result": {  
  "state": "Success",  
  "message": null,  
  "validationErrors": null  
}  
}
```

API для пакетного изменения балансов

Изменено Александра Коваленко

в 23/05/2019 05:48

Оглавление

Область применения	3
Принцип работы	3
Правила взаимодействия	3
Начисление бонусов и ручное начисление бонусов	3
Возврат бонусов и ручной возврат бонусов	3
Общие параметры запросов	4
Значения параметра type	4
Начисление бонусов	4
Возврат бонусов	5
Ручное начисление бонусов	6
Ручной возврат бонусов	6
Ответы на запросы со стороны процессинга Loymax	7

Содержание:

- [Область применения](#)
- [Принцип работы](#)
- [Правила взаимодействия](#)
 - [Общие параметры запросов](#)
 - [Значения параметра type](#)
- [Начисление бонусов](#)
- [Возврат бонусов](#)
- [Ручное начисление бонусов](#)
- [Ручной возврат бонусов](#)
- [Ответы на запросы со стороны процессинга Loytech](#)

Область применения

Данный протокол позволяет производить групповые начисления и списания бонусов в системе Loytech, если суммы были рассчитаны во внешней системе.

В результате выполнения данных операций в истории клиента будет добавлена запись с указанием суммы и описанием, переданным из внешней системы.

Со стороны аналитики можно будет отслеживать количество проводимых операций начисления/списания.

Принцип работы

Для работы с протоколом используется специальный сервис, имеющий вид **https://<ваш_адрес>/loytech.batchoperationsservice/swagger/ui/index** (например, <https://api-rshb-stg.loytech.net/loytech.batchoperationsservice/swagger/ui/index>).

Правила взаимодействия

Протокол поддерживает несколько типов запросов:

- [Начисление бонусов](#)
- [Возврат бонусов](#)
- [Ручное начисление бонусов](#)
- [Ручной возврат бонусов](#)

Начисление бонусов и ручное начисление бонусов

В систему Loytech могут поступать запросы на начисление бонусов за покупки совершенные во внешней системе. В таком случае рекомендуется использовать запрос **Начисление бонусов**.

Также пополнение счета может происходить, например, в виде фиксированных регулярных вознаграждений, компенсации при ошибках расчета суммы во внешней системе и т.п. В таком случае рекомендуется использовать запрос **Ручное начисление бонусов**.

При таком разделении в аналитике можно рассматривать по отдельности эти два типа начислений.

Возврат бонусов и ручной возврат бонусов

Аналогично построена логика и при списании средств с бонусного счета.

Для списания бонусов в оплату покупки во внешней системе или при возврате покупки (операции, связанные с кассой), рекомендуется использовать запрос **Возврат бонусов**.

Для дополнительных списаний, например, в случае прекращения участия в ПЛ, корректировкам по счету, фрода и т.д. рекомендуется использовать запрос **Ручной возврат бонусов**.

В случае успешной обработки запроса возвращается статус запроса 0, в случае ошибки - 1. В конце текущего раздела представлен формат ответа от системы loyтах.

Общие параметры запросов

Параметр	Значение
<i>requestId</i>	Уникальный идентификатор запроса
<i>type</i>	Параметр, определяющий тип запроса
<i>accrueDescription*</i>	Описание операции
<i>currencyExternalId**</i>	Идентификатор валюты
<i>lifeTimeDefinitionExternalId***</i>	Идентификатор времени жизни валюты

*Необязательный параметр.

**Если начисление/возврат бонусов будет производится без идентификатора валюты, то будет использована валюта по умолчанию.

***Если начисление/возврат бонусов будет производится без идентификатора времени жизни, то будет использовано время жизни по умолчанию.

Значения параметра *type*

Параметр и его значение	Действие
"type": 2	Запрос на начисление бонусов
"type": 3	Запрос на возврат бонусов
"type": 4	Запрос на ручное начисление бонусов
"type": 5	Запрос на ручной возврат бонусов

Начисление бонусов

Пример запроса:

```
{
  "requestId": "123",
  "type": 2,
  "accrueDescription": "string",
  "currencyExternalId": "string",
  "lifeTimeDefinitionExternalId": "string",
  "accrue": [
    {
      "memberNo": "012345",
      "Description": "string",
      "operationDate": "2013-08-14 12:26:14",
      "merchantName": " merchantName",
      "transactionAmount": 152246.54,
      "pointsAmount": 1522.46
    },
    {
      "memberNo": "012345",
      "Description": "string",
      "operationDate": "2013-08-14 12:26:14",
      "merchantName": " merchantName",
      "transactionAmount": 152246.54,
      "pointsAmount": 1522.46
    },
    {
      "memberNo": "012345",
```

```
"Description": "string",
"operationDate": "2013-08-14 12:26:14",
"merchantName": " merchantName",
"transactionAmount": 152246.54,
"pointsAmount": 1522.46
}
]
```

Параметр	Значение
<i>memberNo</i>	Уникальный номер Участника, идентифицирующий его в системе Лоутах (Аналог параметра
<i>Description</i>	Описание операции
<i>operationDate</i>	Дата и время проведения операции
<i>merchantName</i>	Название магазина, в котором произведена оплата по карте
<i>transactionAmount</i>	Сумма транзакции
<i>pointsAmount</i>	Сумма баллов к начислению

Возврат бонусов

Пример запроса:

```
{
"requestId": "123",
"type": 3,
"accrueDescription": "string",
"currencyExternalId": "string",
"lifeTimeDefinitionExternalId": "string",
"accrue": [
{
"memberNo": "012345",
>Description": "string",
"operationDate": "2013-08-14 12:26:14",
"merchantName": " merchantName",
"transactionAmount": 152246.54,
"pointsAmount": 1522.46
},
{
"memberNo": "012345",
>Description": "string",
"operationDate": "2013-08-14 12:26:14",
"merchantName": " merchantName",
"transactionAmount": 152246.54,
"pointsAmount": 1522.46
},
{
"memberNo": "012345",
>Description": "string",
"operationDate": "2013-08-14 12:26:14",
"merchantName": " merchantName",
"transactionAmount": 152246.54,
"pointsAmount": 1522.46
}
]
}
```

Параметр	Значение
<i>memberNo</i>	Уникальный номер Участника, идентифицирующий его в системе Лоутах (Аналог параметра
<i>Description</i>	Описание операции

<i>operationDate</i>	Дата и время проведения операции
<i>merchantName</i>	Название магазина, в котором произведена оплата по карте
<i>transactionAmount</i>	Сумма транзакции
<i>pointsAmount</i>	Сумма баллов к возврату

Ручное начисление бонусов

Пример запроса:

```
{
  "requestId": "123",
  "type": 4,
  "accrueDescription": "string",
  "currencyExternalId": "string",
  "lifeTimeDefinitionExternalId": "string",
  "accrue": [
    {
      "memberNo": "012345",
      "Description": "string",
      "operationDate": "2013-08-14 12:26:14",
      "merchantName": " merchantName",
      "transactionAmount": 152246.54,
      "pointsAmount": 1522.46
    },
    {
      "memberNo": "012345",
      "Description": "string",
      "operationDate": "2013-08-14 12:26:14",
      "merchantName": " merchantName",
      "transactionAmount": 152246.54,
      "pointsAmount": 1522.46
    },
    {
      "memberNo": "012345",
      "Description": "string",
      "operationDate": "2013-08-14 12:26:14",
      "merchantName": " merchantName",
      "transactionAmount": 152246.54,
      "pointsAmount": 1522.46
    }
  ]
}
```

Параметр	Значение
<i>memberNo</i>	Уникальный номер Участника, идентифицирующий его в системе Лоутех (Аналог параметра
<i>Description</i>	Описание операции
<i>operationDate</i>	Дата и время проведения операции
<i>merchantName</i>	Название магазина, в котором произведена оплата по карте
<i>transactionAmount</i>	Сумма транзакции
<i>pointsAmount</i>	Сумма баллов к начислению

Ручной возврат бонусов

Пример запроса:

```
{
```

```

"requestId": "123",
"type": 5,
"accrueDescription": "string",
"currencyExternalId": "string",
"lifeTimeDefinitionExternalId": "string",
"accrue": [
  {
    "memberNo": "012345",
    "Description": "string",
    "operationDate": "2013-08-14 12:26:14",
    "merchantName": " merchantName",
    "transactionAmount": 152246.54,
    "pointsAmount": 1522.46
  },
  {
    "memberNo": "012345",
    "Description": "string",
    "operationDate": "2013-08-14 12:26:14",
    "merchantName": " merchantName",
    "transactionAmount": 152246.54,
    "pointsAmount": 1522.46
  },
  {
    "memberNo": "012345",
    "Description": "string",
    "operationDate": "2013-08-14 12:26:14",
    "merchantName": " merchantName",
    "transactionAmount": 152246.54,
    "pointsAmount": 1522.46
  }
]
}

```

Параметр	Значение
<i>memberNo</i>	Уникальный номер Участника, идентифицирующий его в системе Loymax (Аналог параметра <i>memberNo</i>)
<i>Description</i>	Описание операции
<i>operationDate</i>	Дата и время проведения операции
<i>merchantName</i>	Название магазина, в котором произведена оплата по карте
<i>transactionAmount</i>	Сумма транзакции
<i>pointsAmount</i>	Сумма баллов к возврату

Ответы на запросы со стороны процессинга Loymax

Пример ответа:

```

{
  "requestId": "123",
  "type": 2,
  "status": 0,
  "message": "Операция выполнена успешно"
}

```

Параметр	Значение
<i>requestId</i>	Идентификатор запроса, на который даётся ответ
<i>type</i>	Тип ответа, совпадающий с типом запроса
<i>status</i>	Статус запроса: 0 – успех, 1 – ошибка

transactionAmount
message

Сумма транзакции
Текстовый комментарий. Текстовым сообщением сопровождаются как успешно обработанные

Социальные сети

Изменено Ирина Вахитова

в 15/05/2019 08:21

Оглавление

Система Loymax интегрирована с наиболее популярными социальными сетями



ВКонтакте



Одноклассники



Facebook



Instagram



Мой мир
(Mail.Ru)

В системе фиксируются следующие **события**, связанные с социальными сетями:

Событие	Описание
Привязка профиля соц. сети	Событие происходит при привязке Участником в ЛК профиля социальной сети. Это позволит ему в дальнейшем авторизоваться в системе с помощью учетных данных привязанной социальной сети, а также участвовать в различных маркетинговых акциях, направленных на социальные сети.
Вступление в группу	Событие происходит при вступлении Участника в группу социальной сети
Выход из группы	Событие происходит при выходе Участника из группы социальной сети
Активность в социальных сетях	Событие происходит, когда Участник ставит отметки Нравится к записям в группе социальной сети, оставляет комментарии или делает репосты записей (объекты учета активности настраиваются в акции). Чтобы это событие произошло, клиенту следует привязать к своей учетной записи социальные сети в личном кабинете ПЛ.
Удаление активности в социальных сетях	Событие инициируется при отмене Участником отметок Нравится, комментариев или репостов на своей странице. Чтобы это событие зарегистрировалось, у клиента в личном кабинете должны быть привязаны социальные сети.

Зарегистрироваться в Программе лояльности также можно с использованием любой из перечисленных социальных сетей

Активность Соц.сеть					
Отметка Нравится	+	+	+		
Репост	+				
Комментарий	+	+	+		
Вступление в группу	+		+		
Привязка аккаунта	+	+	+	+	+

Для каждого события, связанного с социальной сетью, в акциях могут быть настроены следующие **действия**:

- Начисление дополнительных бонусов или списание бонусов
- Отправка сообщения – SMS, email, push или viber
- Счетчик – для подсчета количества событий
- Включение/выключение рассылки

Обработка события **Привязка профиля социальной сети** происходит сразу в момент добавления связи в Личном кабинете.

Например, начисление бонусов на счет клиента за привязку аккаунта социальной сети произойдет сразу после совершения данного действия (если такие правила предусмотрены в акции).

Для обработки остальных событий необходимо создать регулярную задачу для сбора данных по заданным группам социальных сетей.

Периодически запускаемый обработчик проверяет последние публикации в группе социальной сети, ассоциированной с Программой лояльности, и собирает информацию об участниках группы, комментариях, репостах, отметках Нравится. Далее в соответствии с настройками акций происходит обработка полученной информации, по результатам которой выполняются или не выполняются заданные действия.

Таким образом, на основе событий в социальных сетях можно создавать [различные акции](#):

- Бонусы за привязку профиля социальной сети
- Бонусы за вступление в группу социальной сети
- Бонусы за репосты/ комментарии/ отметки Нравится
- 5 бонусов за первую отметку Нравится, по 1 бонусу за последующие, но не более 15 в месяц
- Списание бонусов в случае выхода из группы
- Списание бонусов в случае удаления репоста

ЧИТАЙТЕ ТАКЖЕ:

- [Инструкция по настройке взаимодействия системы Loytax с социальными сетями](#)

