passteam
MARKETING AUTOMATION

Основные функции
и техническое описание

Общая информация

API позволяет работать с виртуальными картами лояльности в системе Passteam. С помощью API Passteam можно выполнить следующее:
получать информацию о шаблонах;
создавать клиентов;
получать информацию о клиенте;
изменять информацию в карточке клиента (например, баланс или данные держателя);
отправлять ссылки на карты в SMS и Email;
отправлять push-уведомления на карты;
и т.д.

Postman коллекция (примеры запросов)

Песочница: https://documenter.getpostman.com/view/3906161/SVzw41At?version=latest
Общий вид запросов:https://documenter.getpostman.com/view/3906161/SVtN4XSh?version=latest

Формат запроса
Каждый метод доступен по URL вида: [server_address]oapi/[version_number]/[method_name]
В качестве server_address используется URL: https://getpass.passteam.ru
Все методы API доступны с помощью POST запросов. Параметры version_number и method_name необходимо передавать как параметры URL.

Необходимо указывать тип содержимого как multipart/form-data
Пример
POST https://getpass.passteam.ru/oapi/v1/getatoken

Параметры запроса

Path Parameters

Name	Type	Description
method_name	string	getatoken, gettemplates, createcard, updatecards и другие
version_number:	string	v1
server_address	string	https://getpass.passteam.ru/

Headers

Name	Type	Description
authorization	string	{токен, полученный методом getatoken или в личном кабинете в разделе - профиль}

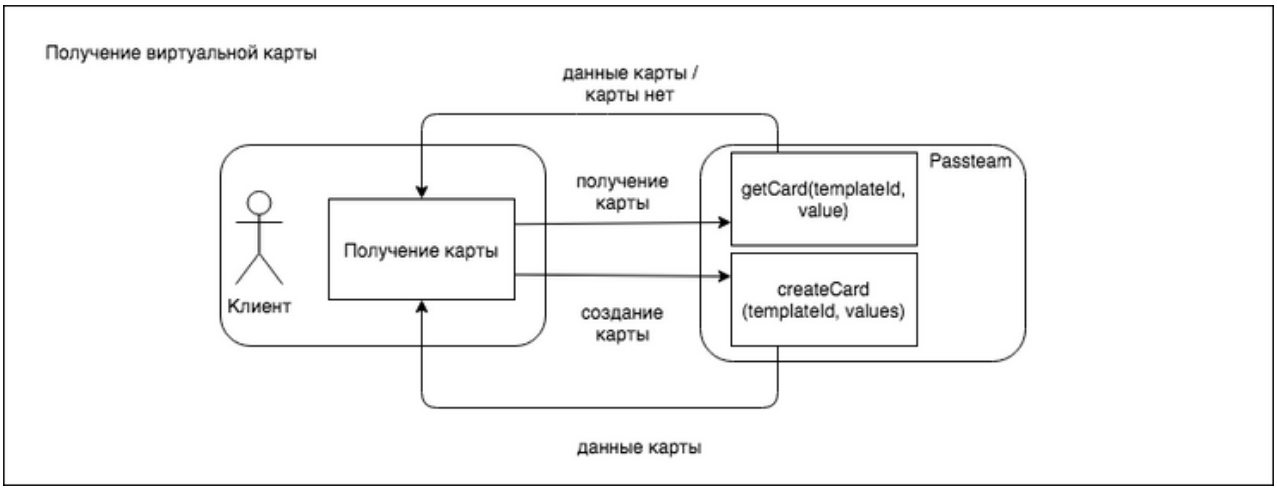
Формат ответа
Ответ в формате JSON.

Параметр	Значение
result (object)	Результат выполнения метода
message (string)	Сообщение об ошибке
code (integer)	Код ответа сервера
error (boolean)	Флаг наличия ошибки

Примеры сценариев интеграции

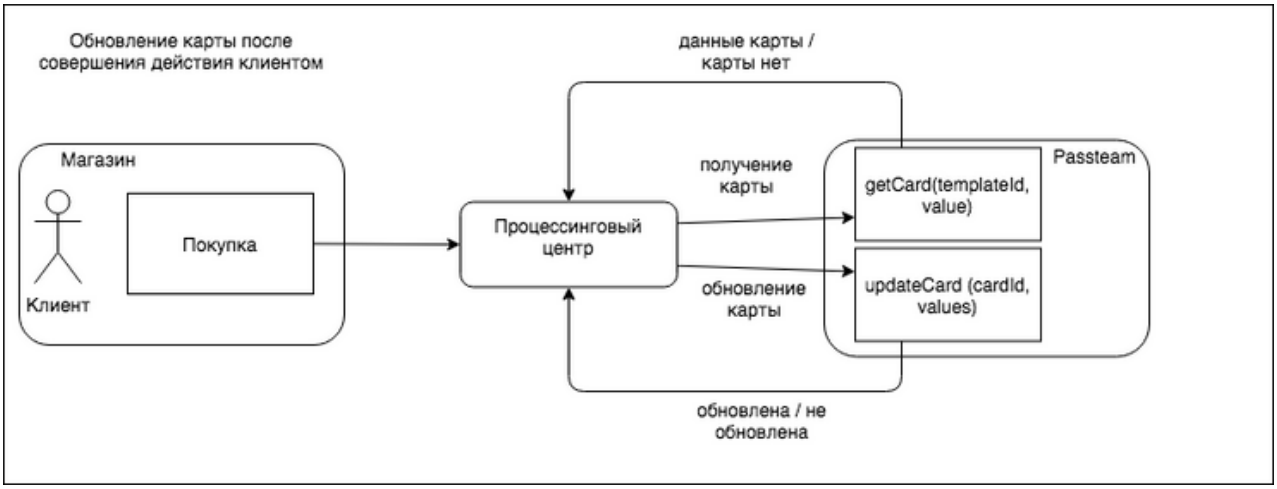
Получение клиента
Когда клиент хочет получить виртуальную карту необходимо:

1. Проверка, создан ли клиент в Passtem (метод getCard).
2. Если создан, то метод вернет данные по нему. В этих данных будет ссылка на скачивание виртуальной карты.
3. Если клиент не создан, то необходимо его создать (метод createCard). После создания, будет возвращены данные по клиенту в которых будет ссылка на карту.

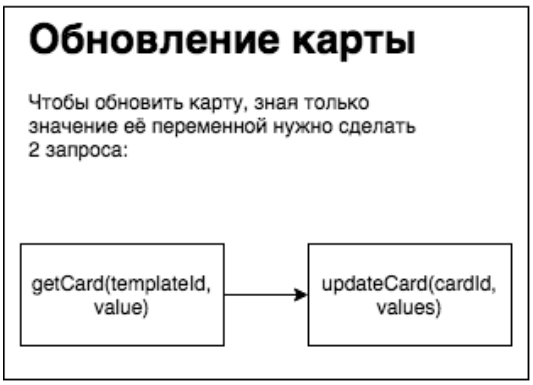


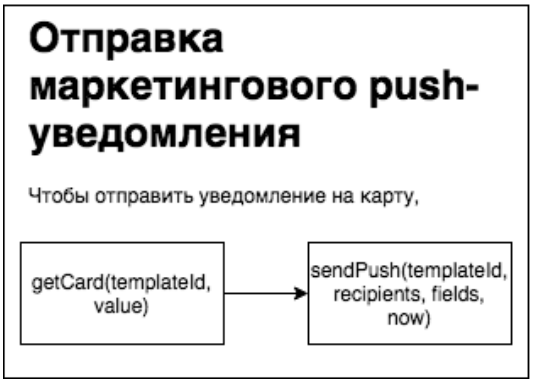
Обновление данных по клиенту
Для того, чтобы обновить данные клиента необходимо:

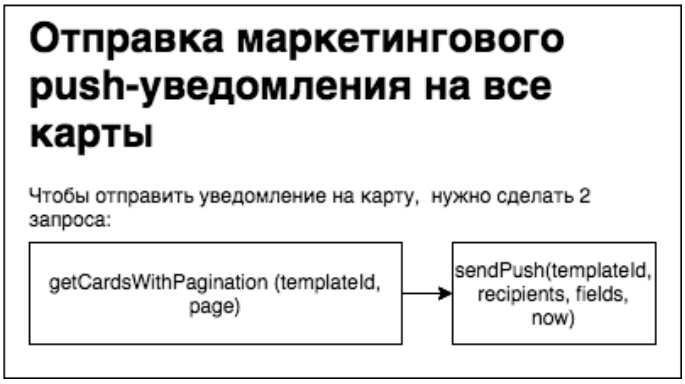
1. Получить клиента (метод getCard). Если клиент создан, то вернутся его данные. Если клиента нет, то см. сценарий получения клиента.
2. Из данных, которые вернул метод getCard необходимо взять cardId и обновить данные клиента методом updateCard



Дополнительные сценарии







Интеграция с 1С

На данной странице приводится пример интеграции 1С с Passteam

Описание интеграции

Интеграция 1С и Passteam рассматривается на примере программного продукта 1С:Розница 2.3.1.40.

Карты лояльности представлены справочником ИнформационныеКарты. Необходимо предусмотреть нетиповые реквизиты:

- cardId - строка, длина 24. Идентификатор карты в Passteam.
- cardlUrl – строка, неограниченной длины. Ссылка на скачивание карты.
- installed – булево. Статус установки карты.

Справочник информационные карты

В модуле объекта справочника ИнформационныеКарты, в событии ПередЗаписью добавляем проверку: Если ПустаяСтрока(ЭтотОбъект.cardId) Тогда … Если данное условие выполняется отправляем запрос CreateCard и после получения ответа необходимо записать cardId и cardUrl в карту лояльности 1С в соответствующие реквизиты. Если условие не выполняется, то необходимо отправить запрос UpdateCard и после получения ответа записать installed в соответствующий реквизит карты лояльности 1С.

Регистр накопления БонусныеБаллы

Для того, чтобы своевременно обновлять бонусный баланс клиента на карте, необходимо в регистре накопления БонусныеБаллы в событии ПриЗаписи вызывать функцию UpdateCard.

Примеры кода

Листинг справочника ИнформационныеКарты

```
// Процедура - обработчик события "ПередЗаписью".
Процедура ПередЗаписью(Отказ)

 Если ОбменДанными.Загрузка Тогда
  Возврат;
 КонецЕсли;

 Если ВидКарты = Перечисления.ВидыИнформационныхКарт.Штриховая Тогда
  КодКарты = "";
 КонецЕсли;

 Если ТипЗнч(ВладелецКарты) = Тип("СправочникСсылка.Контрагенты")
  ИЛИ ТипКарты = Перечисления.ТипыИнформационныхКарт.Регистрационная Тогда
  ДатаСледующегоОпроса = Дата("00010101");
 КонецЕсли;

 Попытка
  Если ПустаяСтрока(ЭтотОбъект.cardId) Тогда
   CreateCard();
  Иначе
   updateCard();
  КонецЕсли;
 Исключение
  Сообщить("При создании карты в Passteam возникла ошибка" + ОписаниеОшибки());
 КонецПопытки;
```

```
КонецПроцедуры

#КонецОбласти

#КонецЕсли

# Область ИнтеграцияPassteam
Процедура CreateCard()
 Соединение = Новый HTTPСоединение("getpass.passteam.ru",443,,,,Новый
ЗащищенноеСоединениеOpenSSL());

 Заголовки = Новый Соответствие;
 Заголовки.Вставить("Authorization", "АВТОРИЗАЦИОННЫЙ ТОКЕН");
 Заголовки.Вставить("Content-Type", "application/x-www-form-urlencoded");

 Запрос = Новый HTTPЗапрос("/oapi/v1/createcard", Заголовки);
 СтрокаЗапроса =
("templateId=%1&values[%%CLIENT_NAME%%]=%2&values[%%BIRTHDAY%%]=%3&values[%%PHONE%%]=%
4&values[%%EMAIL%%]=%5&values[%%BONUS%%]=%6") ;

 Если ЗначениеЗаполнено(ЭтотОбъект.ВладелецКарты) Тогда
 НомерТелефона = ?(ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация.Количество
()=0,"",ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация[0].НомерТелефона);
 АдресЭП = ?(ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация.Количество
()=0,"",ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация[0].АдресЭП);

 Если ТипЗнч(ЭтотОбъект.ВладелецКарты) = Тип("СправочникСсылка.ФизическиеЛица") Тогда
 ДатаРождения = Формат(ЭтотОбъект.ВладелецКарты.ДатаРождения,"ДФ=dd.ММ.уууу");
 Иначе
 ДатаРождения = "";
 КонецЕсли;
 Иначе
 НомерТелефона = "";
 АдресЭП = "";
 ДатаРождения = "";
 КонецЕсли;

 СтрокаЗапроса = СтрШаблон(СтрокаЗапроса,"ИДЕНТИФИКАТОР
ШАБЛОНА",ЭтотОбъект.ВладелецКарты,ДатаРождения,НомерТелефона,АдресЭП,0);
 Запрос.УстановитьТелоИзСтроки(СтрокаЗапроса);

 Ответ = Соединение.ОтправитьДляОбработки(Запрос);

 ЧтениеJSON = Новый ЧтениеJSON;
 СтрокаДоЧтения = СтрЗаменить(Ответ.ПолучитьТелоКакСтроку(), "%","");
 ЧтениеJSON.УстановитьСтроку(СтрокаДоЧтения);
```

```
Данные = ПрочитатьJSON(ЧтениеJSON, Ложь);
 Если Данные.code = 200 Тогда
 Сообщить("Карта создана успешно!");
 ЭтотОбъект.cardId = Данные.result.cardId;
 ЭтотОбъект.CardUrl = Данные.result.cardUrl;
 ИначеЕсли Данные.code = 409 Тогда
 Сообщить("Данный номер телефона уже привязан к карте");
 ИначеЕсли Данные.code = 500 И Данные.message = "Provided field (PHONE) is unique and must be set."
Тогда
 Сообщить("Для создания карты необходимо указать номер мобильного телефона");
 КонецЕсли;

КонецПроцедуры

Процедура UpdateCard() Экспорт
 Соединение = Новый HTTPСоединение("getpass.passteam.ru",443,,,,Новый
ЗащищенноеСоединениеOpenSSL());
 Заголовки = Новый Соответствие;
 Заголовки.Вставить("Authorization", "АВТОРИЗАЦИОННЫЙ ТОКЕН");
 Заголовки.Вставить("Content-Type", "application/x-www-form-urlencoded");

 Запрос = Новый HTTPЗапрос("/oapi/v1/updatecard", Заголовки);

 СтрокаЗапроса =
("cardId=%1&values[%%CLIENT_NAME%%]=%2&values[%%BIRTHDAY%%]=%3&values[%%PHONE%%]=%4&v
alues[%%EMAIL%%]=%5&values[%%BONUS%%]=%6") ;
 Если ЗначениеЗаполнено(ЭтотОбъект.ВладелецКарты) Тогда
 НомерТелефона = ?(ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация.Количество
()=0,"",ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация[0].НомерТелефона);
 АдресЭП = ?(ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация.Количество
()=0,"",ЭтотОбъект.ВладелецКарты.КонтактнаяИнформация[0].АдресЭП);

 Если ТипЗнч(ЭтотОбъект.ВладелецКарты) = Тип("СправочникСсылка.ФизическиеЛица") Тогда
 ДатаРождения = Формат(ЭтотОбъект.ВладелецКарты.ДатаРождения,"ДФ=dd.MM.yyyy");
 Иначе
 ДатаРождения = "";
 КонецЕсли;
 Иначе
 НомерТелефона = "";
 АдресЭП = "";
 ДатаРождения = "";
 КонецЕсли;

 ЗапросБаллов = Новый Запрос;
 ЗапросБаллов.Текст =
 "ВЫБРАТЬ
 | БонусныеБаллыОстатки.НачисленоОстаток КАК НачисленоОстаток
 |ИЗ
 | РегистрНакопления.БонусныеБаллы.Остатки(&ТекущаяДата, ДисконтнаяКарта =
&ДисконтнаяКарта) КАК БонусныеБаллыОстатки";
```

```
ЗапросБаллов.УстановитьПараметр("ДисконтнаяКарта", ЭтотОбъект.Ссылка);
ЗапросБаллов.УстановитьПараметр("ТекущаяДата", ТекущаяДата());

РезультатЗапроса = ЗапросБаллов.Выполнить().Выгрузить();
Если РезультатЗапроса.Количество() = 0 Тогда
Баллы = 0;
Иначе Баллы = РезультатЗапроса[0].НачисленоОстаток;
КонецЕсли;

СтрокаЗапроса =
СтрШаблон(СтрокаЗапроса,ЭтотОбъект.cardId,ЭтотОбъект.ВладелецКарты,ДатаРождения,НомерТелефона,АдресЭП,Баллы);
Запрос.УстановитьТелоИзСтроки(СтрокаЗапроса);

Ответ = Соединение.ОтправитьДляОбработки(Запрос);

ЧтениеJSON = Новый ЧтениеJSON;
СтрокаДоЧтения = СтрЗаменить(Ответ.ПолучитьТелоКакСтроку(), "%","");
ЧтениеJSON.УстановитьСтроку(СтрокаДоЧтения);

Данные = ПрочитатьJSON(ЧтениеJSON, Ложь);

Если Данные.code = 200 Тогда
Сообщить("Карта успешно обновлена!");
ЭтотОбъект.installed = Данные.result.card.installed;
ИначеЕсли Данные.code = 409 Тогда
Сообщить("Данный номер телефона уже привязан к карте");
ИначеЕсли Данные.code = 500 И Данные.message = "Provided field (PHONE) is unique and must be set."
Тогда
Сообщить("Для создания карты необходимо указать номер мобильного телефона");
КонецЕсли;

КонецПроцедуры
#КонецОбласти
```

В АВТОРИЗАЦИОННЫЙ ТОКЕН и ИДЕНТИФИКАТОР ШАБЛОНА необходимо подставить соответствующие значения. Если вы не знаете, где их взять, уточните эту информацию у менеджера Passteam.

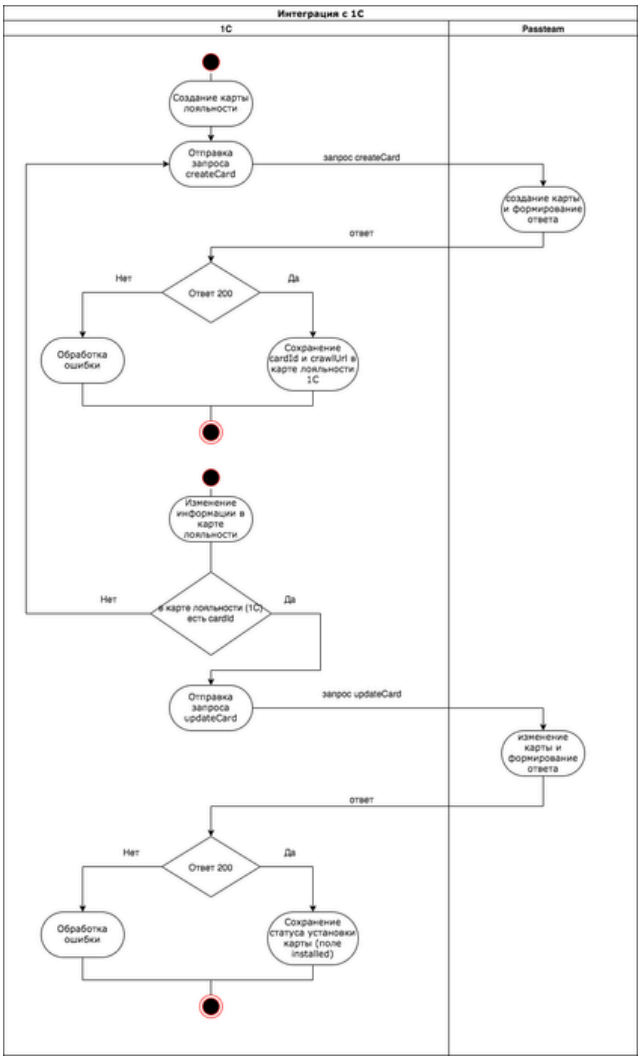
В строке
Copy
СтрокаЗапроса = ("templateId=%1&values[%%CLIENT_NAME%%]=%2&values[%%BIRTHDAY%%]
CLIENT_NAME и BIRTHDAY - названия переменных в шаблоне. Они могут быть другими, чтобы узнать какие у вас переменные - обратитесь к менеджеру Passteam.

В некоторых случаях могут возникать проблемы с передачей спецсимволов при отправке http запроса.
В таких случаях необходимо передавать символы как **[** как **%5B%** как **%25]** как **%5D** То есть как в чистом http запросе. Например, строка запроса после всех подстановок будет выглядеть так:
cardId=5bea7e305abcfd6b708b4573&values%5B%25BALANCE%25%5D=7500&values%5B%25_CODE%25%5D=3006880536055&values%5B%25NUMBERCARD%25%5D=3006880536055&values%5B%25DISCOUNT%25%5D=5

Листинг регистра накопления БонусныеБаллы

```
Процедура ПриЗаписи(Отказ, Замещение)
 Попытка
  Для каждого эл Из ЭтотОбъект Цикл
   нОбъект = Эл.ДисконтнаяКарта.ПолучитьОбъект();
   нОбъект.UpdateCard();
  КонецЦикла;
 Исключение
  Сообщить(ОписаниеОшибки());
 КонецПопытки;
КонецПроцедуры
```

Схема интеграции



Получение токена

Описание: Если вам необходимо получить токен с помощью API, используйте метод getatoken.
Как использовать: Обычно нет необходимости получать токен с помощью метода, достаточно скопировать токен (API ключ) в личном кабинете (в разделе профиль) и использовать его в методах. Метод используется в случае разработки интеграций с CRM-системами или процессингом, в котором планируется работа большого количества брендов. В таком случае внутри интерфеса CRM-системы пользователь вводит логин и пароль Passteam, по которым можно получить токен для работы интеграций. Если ваша интеграция кастомная и только для одного бренда (пользователя Passteam), не используйте этот метод. Не используйте метод при каждом запросе. Токен действителен в течение 365 дней.
При изменении логина, email или пароля необходимо заново получать токен

Getatoken
POST https://getpass.passteam.ru/oapi/v1/getatoken

Request Body

Name	Type	Description
nickname	string	Логин пользователя
password	string	Пароль пользователя

200 Cake successfully retrieved.

```
{
    "accessToken": "ACCESS_TOKEN",
}
```

Получить идентификаторы шаблонов пользователя

Описание: Для получения всех идентификаторов шаблона, которые принадлежат пользователю, существует метод getlisttemplates.
Как использовать: Обычно метод используется при разработке интеграций с CRM системами или процессингом. Используется для проверки доступности определенного шаблона пользователю.
getlisttemplates

POST https://getpass.passteam.ru/oapi/v1/getlisttemplates

Headers

Name	Type	Description
Authorization	string	Токен авторизации пользователя

200

```
{
 "result": {
 "templateIds": [
 "id1",
 "id2",
 "id3",
 "id4"
 ]
 },
 "message": "OK",
 "code": 200,
 "error": false
}
```

Получить информацию о шаблоне

Описание: Метод **gettemplate** позволяет получить все данные о шаблоне по его идентификатору.
Как использовать: Метод используется при создании интеграций. С помощью данного метода можно получить общую статистику по шаблону.
template - это шаблон компании в котором хранятся данные по дизайну карт, онлайн анкете, сертификатам и тд. 1 компания - 1 template.
gettemplate

`POST` `https://getpass.passteam.ru/oapi/v1/gettemplate`

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона


```
{
 "result": {
  "templateId": "TEMPLATE_ID",
  "passTypeId":"PASSTYPE_ID"
  "title": "TEMPLATE_TITLE",
  "teamIdentifier":"TEAM_IDENTIFIER",
  "company":"MY_COMPANY",
  "city":"MY_CITY",
  "type":"TEMPLATE_TYPE"
  "cardsCount": CARDS_COUNT,
  "devicesCounter": {
   "summ": SUMM_DEVICES,
   "0": SUMM_IOS,
   "1": SUMM_ANDROID,
   "2": SUMM_WINDOWS_PHONE
  },
  "fields": [{
   "title": "FIELD_TITLE",
   "name": "FIELD_NAME",
   "type": "text"
  }, ...],
  "alias": null,
  "qrUrl": null,
  "vacant":false,
  "userTitle":"MyFirstTemplate",
  "owner":"MY_USERNAME"
 },
 "message": "OK",
 "code": 200,
 "error": false
}
```

Создание клиента

Описание: Метод позволяет создать клиента с нужными параметрами. Переменные (values) необходимо создать в личном кабинете, вы можете использовать любые переменные в количестве не более 24.
Как использовать: Метод нужно использовать при необходимости создать нового клиента. Обычно при создании клиента передаются определенные параметры, например, номер телефона, ФИО, количество баллов. В ответе будет содержаться ссылка на созданную карту и другие данные.
Валидация номера телефона: Валидация работает на основе библиотеке libphonenumber от Google. Онлайн демонстрация работы библиотеки https://giggsey.com/libphonenumber/index.php

createcard

`POST` `https://getpass.passteam.ru/oapi/v1/createcard`

Request Body

Name	Type	Description
sendEmail	string	Отправка Email после создания карты, 1 или 0. Если в личном кабинете настроена автоматическая отправка при создании карты, не указывать
templateId	string	Идентификатор шаблона
values	string	Массив значений карты. Если не указать, карта будет пустой
sendSms	integer	Отправка SMS после создания карты, 1 или 0. Если в личном кабинете настроена автоматическая отправка при создании карты, не указывать


```
{
 "result": {
  "cardId": "CARD_ID",
  "cardCode":"CARD_CODE",
  "cardUrl": "CARD_URL",
      "appleUrl": "Apple_Wallet_CARD_URL",
      "googleUrl": "Google_Pay_CARD_URL",
  "qrcodeUrl": "QR_CODE_URL",
  "values":{
  FIELD_NAME1: FIELD_VALUE1
  }
  "deviceRegistered":false,
  "expirationDate": "",
      "voided": 0,
      "phoneNumber": "MY_PHONE_NUMBER",
      "email": "MY_EMAIL",
      "strip": "",
      "logo": "",
      "icon": "",
      "templateId": "TEMPLATE_ID",
      "importUid": "",
      "installedGPay": 0,
      "installedAW": 0,
      "installed": false,
      "created": "CREATED_DATE",
      "updated": "UPDATED_DATE"
       },
 "message": "OK",
 "code": 200,
 "error": false
}
```

Поиск дубликатов при создании клиента:
Сценарий работает для уникальной переменной (в списке переменных).
При создании клиента проверяется уникальность значения выбранной переменной. Если значение уникально, то клиент создается. Если значение повторяется, то клиент создана не будет, а вернется ошибка с указанием идентификаторов клиентов, имеющих такое же значение переменной.
В качестве примера, мы рассмотрим сценарий, при котором создается клиента с уникальной переменной %EXAMPLE%. При этом значение этой переменной не уникально.
createcard
POST https://getpass.passteam.ru/oapi/v1/createcard
Пример срабатывания метода с дублирующим значением cardCode

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона
values[%EXAMPLE%]	string	значение переменной


```
{
   "result": null,
   "message": "Provided value (value) of field (%EXAMPLE%) is duplicated. Duplicated cards: cardId1 cardId2
",
   "code": 409,
   "error": true
}
```


Информация о клиенте

Описание: Для получения информации об определенном клиенте используется метод **getcard**. Карту можно искать или по системным параметрам Passteam (cardId или cardCode), или по значениям переменных (values).
Как использовать: Метод обычно используется при необходимости получить ссылку на существующую карту. Например, метод можно использовать при интеграции выдачи карт на сайте.

getCard
POST https://getpass.passteam.ru/oapi/v1/getcard
Поиск карты по cardId или cardCode

Request Body

Name	Type	Description
cardId	string	Уникальный ID карты. Поиск сначала производится по cardId, если он указан. Если нет, то поиск осуществляется по связке templateId и cardCode или templateId и values.
templateId	string	Идентификатор шаблона. Обязательно при поиске карты по cardCode или values.
cardCode	string	Девятизначный код карты с ведущими нулями.

```
{
 "result": {
  "cardId": "CARD_ID",
  "cardCode":"CARD_CODE",
  "cardUrl": "CARD_URL",
       "appleUrl": "Apple_Wallet_CARD_URL",
       "googleUrl": "Google_Pay_CARD_URL",
  "qrcodeUrl": "QR_CODE_URL",
  "values":{
    "%VALUE_NAME%": "VALUE_VALUE1"
  }
  "deviceRegistered":false,
  "expirationDate": "",
       "voided": 0,
       "phoneNumber": "MY_PHONE_NUMBER",
       "email": "MY_EMAIL",
       "strip": "",
       "logo": "",
       "icon": "",
       "templateId": "TEMPLATE_ID",
       "importUid": "",
       "installedGPay": 0,
       "installedAW": 0,
       "installed": false,
       "created": "CREATED_DATE",
       "updated": "UPDATED_DATE"
        },
 "message": "OK",
 "code": 200,
 "error": false
}
```


getCard

POST https://getpass.passteam.ru/oapi/v1/getcard
Поиск карты по значению переменной

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона. Обязательно при поиске карты по cardCode или values.
values[%value%]	string	Переменная и ее значение, с помощью которого будет производится поиск карты, например, [%PHONE%]=+71234567890.

```
{
 "result": {
  "cardId": "CARD_ID",
  "cardCode":"CARD_CODE",
  "cardUrl": "CARD_URL",
      "appleUrl": "Apple_Wallet_CARD_URL",
      "googleUrl": "Google_Pay_CARD_URL",
  "qrcodeUrl": "QR_CODE_URL",
  "values":{
  "PHONE": "71234567890"
  }
  "deviceRegistered":false,
  "expirationDate": "",
      "voided": 0,
      "phoneNumber": "MY_PHONE_NUMBER",
      "email": "MY_EMAIL",
      "strip": "",
      "logo": "",
      "icon": "",
      "templateId": "TEMPLATE_ID",
      "importUid": "",
      "installedGPay": 0,
      "installedAW": 0,
      "installed": false,
      "created": "CREATED_DATE",
      "updated": "UPDATED_DATE"
       },
 "message": "OK",
 "code": 200,
 "error": false
}
```


Обновление карты

Описание: Для обновления информации о клиенте используется метод **updatecard**.
Как использовать:
Метод обновляет значения системных полей и переменных клиента по идентификатору карты.
Изменения на карту приходят путем push - уведомления.
updateCard
POST https://getpass.passteam.ru/oapi/v1/updatecard

Request Body

Name	Type	Description
cardId	string	Идентификатор карты
values	array	Массив переменных для изменения
email	string	Электронная почта
phoneNumber	string	Номер телефона
expirationDate	string	Срок действия карты в формате дд.мм.гггг чч:мм
voided	number	Активна ли карта (1 или 0)

```
{
   "result": {
      "messages": [],
      "updated": true,
      "card":{
      "cardId": "CARD_ID",
      "cardCode":"CARD_CODE",
      "cardUrl": "CARD_URL",
           "appleUrl": "Apple_Wallet_CARD_URL",
         "googleUrl": "Google_Pay_CARD_URL",
     "qrcodeUrl": "QR_CODE_URL",
     "values":{
        "%VALUE_NAME%": "VALUE_VALUE1"
     }
     "deviceRegistered":false,
     "expirationDate": "",
          "voided": 0,
          "phoneNumber": "MY_PHONE_NUMBER",
          "email": "MY_EMAIL",
          "strip": "",
          "logo": "",
          "icon": "",
          "templateId": "TEMPLATE_ID",
          "importUid": "",
          "installedGPay": 0,
          "installedAW": 0,
          "installed": false,
          "created": "CREATED_DATE",
          "updated": "UPDATED_DATE"
    },
   },
   "message": "OK",
   "code": 200,
   "error": false
}
```


Постраничное получение клиентов

Описание: Метод getCardsWithPagination позволяет получать информацию о клиентах шаблона постранично. Метод возвращает информацию о количестве страниц, количестве клиентов, номера предыдущей и следующей страницы. Максимальное количество клиентов на страницу - 50.
Как использовать: Для использования метода, необходимо указывать идентификатор шаблона, номер страницы, начиная с первой и , при необходимости, количество клиентов на каждой странице.

getcardswithpagination
POST https://getpass.passteam.ru/oapi/v1/getcardswithpagination

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона
page	number	Номер страницы
cardsPerPage	number	Количество карт на странице


```
{
 "result":
      "pageCount": 4,
      "nextPage": 3,
      "previouslyPage": 1,
      "cardsCount": CARDS_COUNT,
      "cards":[{
  "cardId": "CARD_ID",
  "cardCode":"CARD_CODE",
  "cardUrl": "CARD_URL",
      "appleUrl": "Apple_Wallet_CARD_URL",
      "googleUrl": "Google_Pay_CARD_URL",
  "qrcodeUrl": "QR_CODE_URL",
  "values":{
  "%VALUE_NAME%": "VALUE_VALUE1"
  }
  "deviceRegistered":false,
  "expirationDate": "",
      "voided": 0,
      "phoneNumber": "MY_PHONE_NUMBER",
      "email": "MY_EMAIL",
      "strip": "",
      "logo": "",
      "icon": "",
      "templateId": "TEMPLATE_ID",
      "importUid": "",
      "installedGPay": 0,
      "installedAW": 0,
      "installed": false,
      "created": "CREATED_DATE",
      "updated": "UPDATED_DATE"
 }, ...],
 "message": "OK",
 "code": 200,
 "error": false
}
```

Проверка переменных на уникальность

Описание: Для проверки значений переменных на уникальность, используется метод checkvariablevalue. Для этого вам необходимо указать идентификатор шаблона, название переменной, которую необходимо проверить, и ее значение.
Как использовать: Используйте метод для проверки уникальности переменной перед созданием клиента. Метод поможет предотвратить создание дубликатов.

checkvariablevalue
POST https://getpass.passteam.ru/oapi/v1/checkvariablevalue

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона
variable	string	Название вашей переменной
value	string	Значение вашей переменной


```
{
 "result": {
 "unique": true,
 "cardsIds": []
 },
 "message": "OK",
 "code": 200,
 "error": false
}
```


Распространение карт

Описание: Для распространения ссылок на карты с помощью SMS и Email используется метод **distribute**. Шаблоны Email и SMS задаются в личном кабинете. Пожалуйста, тестируйте тексты смс и вид email перед массовыми отправками.
Как использовать: Например, помощью данного метода можно реализовать кнопки отправки ссылок на карты в SMS или Email в вашей POS или создать такие кнопки на сайте для удобства установки карт на десктопной версии.

distribute
POST https://getpass.passteam.ru/oapi/v1/distribute

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона (Обязательно)
cards[0][cardId]	string	Идентификатор карты (Обязательно)
cards[0][type][1]	string	Способ отправки ссылки для установки карты (Обязательно)

```
{
 "result": true,
 "message": "OK",
 "code": 200,
 "error": false
}
```


Удаление карт

При удалении карты, карта больше не будет отображаться в базе данных в личном кабинете, при этом электронная карта не удаляется с устройства, на которое она была установлена. После удаления карты, её не возможно будет обновить, отравить уведомления или изменить информацию на карте. Для удаления карты используется метод **deletecard**
POST https://getpass.passteam.ru/oapi/v1/deletecard

Request Body

Name	Type	Description
cardId	string	Идентификатор карты. Если поиск производится по cardId, то остальные параметры указывать не нужно
templateId	string	Идентификатор шаблона. Если поиск производится по temlateId, то необходимо указывать либо cardCode либо cardId. Необязательно
cardCode	string	Девятизначный код карты с ведущими нулями.


```
{
   "result": {
      "n": 1,
      "ok": 1,
      "err": null,
      "errmsg": null
   },
   "message": "OK",
   "code": 200,
   "error": false
}
```


Webhooks

Уведомления о событиях с картами по протоколу http(s)

Принцип работы
Когда происходит событие с картой, система отправляет POST запрос с телом в JSON на указанный URL. Отправка вебхука повторяется через 1-2 часа и через 24 часа. Система не ждет и не обрабатывает ответ по URL.

Возможно предусмотреть запись идентификаторов обработанных вебхуков, чтобы не обрабатывать повторные запросы.

Настройка

- Настройка вебхуков производится в разделе интеграций с CRM (https://app.passteam.io/vue/dist/settings/integrations/webhooks ссылка для настройки).

1. В качестве URL необходимо ввести адрес на который будут отправляться уведомления
2. Необходимо выбрать события, о которых вы хотите получать уведомления

События

1. Создание клиента - происходит, когда в системе создается клиент
2. Обновление клиента - происходит, когда клиент обновляется
3. Удаление клиента - происходит, когда клиента удаляют
4. Установка карты на устройство - происходит, когда клиент устанавливает карту в приложение Apple Wallet или Google Pay
5. Удаление карты с устройства - происходит, когда клиент удаляет карту в приложении Apple Wallet или Google Pay

Обработка вебхука

Вебхук отправляется POST запросом по введенному URL в формате JSON. Для обработки вебхука необходимо обрабатывать входящие POST запросы формата JSON.

Пример обработки вебхука

```
if (!empty(file_get_contents('php://input'))) {
    $webhook = json_decode(file_get_contents('php://input'), true);
}
```

Отправляемые данные

Вебхук состоит из следующих полей:

- webhookId - строка, идентификатор вебхука.
- event - строка, тип события (createCard, updateCard, deleteCard, installCard, unInstallCard)
- card - массив, данные по карте. В таком же формате как возвращает запрос на получение информации о карте

Пример отправляемых данных


```
{
 "card": {
  "cardId": "5e451571cea6c6141807b792",
  "cardCode": "000000018",
  "cardUrl": "testUrl",
  "appleUrl": "testUrl",
  "googleUrl": "testUrl",
  "qrcodeUrl": "testUrl",
  "values": {
   "%POINTS%": "1000"
  },
  "deviceRegistered": false,
  "expirationDate": "",
  "voided": 0,
  "phoneNumber": "",
  "email": "",
  "strip": "",
  "logo": "",
  "icon": "",
  "templateId": "5e424c181dbf0d537c8b4569",
  "importUid": "",
  "installedGPay": 0,
  "installedAW": 0,
  "installed": false,
  "created": "13.02.2020 12:22:57",
  "updated": "13.02.2020 12:22:57"
 },
 "event": "createCard",
 "webhookId": "5e4515b1cea6c61fba2a4212"
}
```

Обновление карт [deprecated]

Вместо этого метода следует использовать updateCard
Описание: Для обновления информации на картах используется метод **updatecards**.
Как использовать: С помощью метода вы можете изменять значение переменных. Список переменных (values), которые используются, можно редактировать во вкладке Переменные в личном кабинете. Данный метод рекомендуется использовать для простых обновлений информации на карте, а также при отправке транзакционных push-уведомлений (например, изменение баланса). Для отправки рекламных уведомлений, используйте метод sendpush. Этим методом лучше обновлять карты по одной. Используйте один из двух вариантов работы данного метода:

Вариант 1. Поиск карты для обновления по значениям переменных, которые вы задали ранее при создании карты
Если вы не храните cardCode или cardId, передаваемые из Passteam, можно искать карту по значению одной из переменных. Переменная должна быть уникальной. Если система найдет более одной карты, подходящей под условия запроса, в ответе будет отправлена ошибка.

updatecards
POST https://getpass.passteam.ru/oapi/v1/updatecards

Request Body

Name	Type	Description
cards[0].templateId	string	Идентификатор шаблона
cards[0].cardValues	array	Переменная и ее значение, по которому будет осуществлен поиск нужной карты для обновления.карт со значениями
cards[0].values	array	Переменная и ее новое значение (указывайте только если не указали cardCode)
cards[0].expirationDate	string	Срок действия карты в формате дд.мм.гггг чч:мм
cards[0].voided	integer	Деактивация карты 1 или 0


```
{
  "result": {
    "messages": [],
    "updated": 1,
    "processed": 1
  },
  "message": "OK",
  "code": 200,
  "error": false
}
```

Вариант 2. Поиск карты для обновления по системным параметрам cardId или cardCode
Используйте этот вариант если при создании карты вы сохранили cardId или cardCode, переданные в ответе на запрос createcard.

updatecards
POST https://getpass.passteam.ru/oapi/v1/updatecards

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона
cards[0].cardId	string	Идентификатор карты (указывайте только если не указали cardCode)
cards[0].cardCode	string	Код карты (указывайте только если не указали cardId)
cards[0].values	string	Переменная и ее новое значение
cards[0].expirationDate	string	Срок действия карты в формате дд.мм.гггг чч:мм
cards[0].voided	integer	Деактивация карты 1 или 0


```
{
  "result": {
    "messages": [],
    "updated": 1,
    "processed": 1
  },
  "message": "OK",
  "code": 200,
  "error": false
}
```


Информация о шаблонах [deprecated]

Вместо этого метода следует использовать gettemplate
В случае использования нескольких шаблонов, вы можете получить подробную информацию используя метод gettemplates. В запросе нужно указать только токен и название метода, в ответе вы получите список шаблонов, которые доступны пользователю. Если интеграция подразумевает от 1 до 10 шаблонов, нет необходимости использовать данный метод. ID шаблона, необходимый для всех запросов, вы можете узнать в списке шаблонов в интерфейсе системы.
Этот метод может выполнятся долго.

Gettemplates
POST https://getpass.passteam.ru/oapi/v1/gettemplates

Headers

Name	Type	Description
authorization	string	Ключ авторизации


```
"result": [{
 "templateId": "TEMPLATE_ID",
  "passTypeId":"PASSTYPE_ID"
  "title": "TEMPLATE_TITLE",
  "teamIdentifier":"TEAM_IDENTIFIER",
  "company":"MY_COMPANY",
  "city":"MY_CITY",
  "type":"TEMPLATE_TYPE"
  "cardsCount": CARDS_COUNT,
  "devicesCounter": {
   "summ": SUMM_DEVICES,
   "0": SUMM_IOS,
   "1": SUMM_ANDROID,
   "2": SUMM_WINDOWS_PHONE
  },
  "fields": [{
   "title": "FIELD_TITLE",
   "name": "FIELD_NAME",
   "type": "text"
  }, ...],
  "alias": null,
  "qrUrl": null,
  "vacant":false,
  "userTitle":"MyFirstTemplate",
  "owner":"MY_USERNAME"
 },],
 "message": "OK",
 "code": 200,
 "error": false
```


Информация о картах [deprecated]

Вместо этого метода следует использовать метод getcard
Для получения информации о картах для указанного шаблона используется метод getcards.
Метод может выполнятся долго.

getCards
POST https://getpass.passteam.ru/oapi/v1/getcards

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона
installedGPay	string	Установка в Google Pay
installedAW	string	Установка в Apple Wallet или копии AW на Android
created	string	Период создания карт


```
{
 "result": [{
  "cardId": "CARD_ID",
  "cardCode":"CARD_CODE",
  "cardUrl": "CARD_URL",
      "appleUrl": "Apple_Wallet_CARD_URL",
      "googleUrl": "Google_Pay_CARD_URL",
  "qrcodeUrl": "QR_CODE_URL",
  "values":{
  "%VALUE_NAME%": "VALUE_VALUE1"
  }
  "deviceRegistered":false,
  "expirationDate": "",
      "voided": 0,
      "phoneNumber": "MY_PHONE_NUMBER",
      "email": "MY_EMAIL",
      "strip": "",
      "logo": "",
      "icon": "",
      "templateId": "TEMPLATE_ID",
      "importUid": "",
      "installedGPay": 0,
      "installedAW": 0,
      "installed": false,
      "created": "CREATED_DATE",
      "updated": "UPDATED_DATE"
 }, ...],
 "message": "OK",
 "code": 200,
 "error": false
}
```


Отправка push-уведомлений [deprecated]

Вы можете отправить push-уведомление на все карты. После отправки push-уведомления, на заблокированном экране устройства, на котором установлена электронная карта, появится сообщение с текстом уведомления (задается в конструкторе шаблона) и текстом, который вы отправили. Значение изменяемого поля также заменится на текст, который вы отправили. Для отправки уведомлений используется метод **sendpush.**

Используйте метод только для отправки рекламных рассылок об акциях, спецпредложениях. Для передачи баланса и других данных используйте метод updatecard

sendpush
POST https://getpass.passteam.ru/oapi/v1/sendpush

Request Body

Name	Type	Description
templateId	string	Идентификатор шаблона
recipients	array	Получатели уведомления (массив идентификаторов карт cardId)
fields	string	Переменная и ее значение (для нужных переменных обязательно установить галочку "Доступно в центре уведомлений" в списке переменных)
now	boolean	Флаг мгновенной отправки
date	string	Дата отправки в формате (дд-мм-гггг чч:мм) Обязателен при now===FALSE


```
{
 "result": true,
 "message": "OK",
 "code": 200,
 "error": false
}
```


Обработка ошибок

Если сервер не может обработать запрос, заголовок HTTP ответа будет содержать код ошибки. Кроме того, ответ содержит подробную информацию в формате JSON.
Пример


```
# Ответ
{
  "result": true,
  "message": "OK",
  "code": 200,
  "error": false
}
```

Коды ошибок

Код:	**200**
Описание	Успешный запрос
Код:	**400**
Описание	Ошибка передачи параметров
Код:	**401**
Описание	Ошибка авторизации
Код:	**403**
Описание	Ошибка доступа
Код:	**404**
Описание	Запрошенный ресурс не найден
Код:	**409**
Описание	Попытка дублирования записи по уникальной переменной
Код	429
Описание	Слишком много запросов. Превышен лимит запросов.
Код:	**500, 502, 503, 504**
Описание	Внутренняя ошибка сервера

Лимиты

API Passteam имеет ограничения по количеству запросов. Количество запросов соотвествует тарифному плану. Passteam оставляет за собой право изменять количество запросов по собственному усмотрению для того, чтобы обеспечивать лучшее качество услуг. Как пользователю API, вам доступно следующее количество запросов:

Увеличение мощности API доступно только до 350 тысяч запросов в сутки всего и не более 30 тысяч запросов в час.
В случае превышения лимита в ответе на запрос вы получите следующее: "message": "You can only make N requests per hour"

Тариф	Запросы в минуту (справочно)	Запросы в час	Запросы в сутки
Стандартный	0	0	0
Профессиональный	16	300	7000
Профессиональный (с лимитом от 10 тыс. устройств)	65	850	20000
Энтерпрайз	165	10000	150000
Увеличение мощности API (только для Энтерпрайз)	165	+10000	+100000

Типы и форматы данных

Переменные могут иметь следующие типы:

- text - любые данные в текстовом формате
- phone - номер телефона в международном формате (с +7). Неправильный формат приведет к ошибке.
- email - адрес электронной почты. Неправильный формат приведет к ошибке
- date - дата, формат указывается в params->dateStyle.
- date_time - дата с временем, формат указывается в params->dateStyle.
- list - список. Элементы списка указаны в params->forList. Если указано значение не из списка, то генерируется ошибка.
- flag - логический тип (true/false)
- number - числовой тип данных

Покупки

Документация для управления покупками в PassteamAPI

GET Get Sale
Параметры:
title[string] - название покупки
paymentType[string] - тип покупки

- CARD
- CASH

paymentStatus[string] - статус покупки

- DRAFT - черновик
- PAID - оплачено

totalPrice[int] - общая цена покупки
cardId[int] - идентификатор клиента, для которого покупка
products[array] = Массива продуктов в формате:

- _id - идентификатор продукта
- name - название продукта
- quantity - количество продуктов в покупке
- price - цена

clientName - имя покупателя

PUT Update Sale

Параметры:
id [string] - идентификатор покупки для изменения
title[string] - название покупки
paymentType[string] - тип покупки

- CARD
- CASH

paymentStatus[string] - статус покупки

- DRAFT - черновик
- PAID - оплачено

totalPrice[int] - общая цена покупки
cardId[int] - идентификатор клиента, для которого покупка
products[array] = Массива продуктов в формате:

- _id - идентификатор продукта
- name - название продукта
- quantity - количество продуктов в покупке
- price - цена

clientName - имя покупателя

DELETE Delete Sale

Параметры:
id [string] - идентификатор покупки

Товары

Документация для управления товарами в PassteamAPI

POST CreateProduct

Параметры:

- companyId - идентификатор компании
- name - название продукта
- price - цена продукта

PUT UpdateProduct

Параметры:

- id - идентификатор продукта
- name - название продукта
- price - цена продукта

DELETE DeleteProduct

Параметры:

- id - идентификатор продукта

POST Products

Параметры:

- companyId - идентификатор компании
- page - номер страницы
- perPage - количество товаров на страниц