Регламент подключения мерчанта к сервису Почта.Pay

==========

Оглавление

Подготовка сервисов к работе

Общие положения

Проверяется платформа мерчанта на ограничения

- Технические:
  - Поддержка протокола https (валидный и не просроченный сертификат)
  - Возможность размещения в публичном доступе сервиса для информирования статуса оплат (только https)
- Организационные:
  - Активное подключение к сервису Мультикарта

Подключается платформа мерчанта в Почта.Pay и Почта.ID:

- Получается и встраивается на front-end части мерчанта вспомогательное PochtaPaySDK
- Выполняется интеграция по основному сценарию **"Покупка и оформление доставки товара"**

На стороне мерчанта:

- Поддерживается сервис для получения набора маркеров доступа
- Реализуется вызов старта оплаты
- Поддерживается сервис для получения информация по оплатам

Дополнительная техническая информация

Адреса сервисов

Сервис	Точки соединения	Протокол
Почта.Pay	pay.pochta.ru	https
Почта.ID	passport.pochta.ru	https

PochtaPaySDK.js - получается у команды Почта.Pay

Подключение к сервису интернет-эквайринга Почта Банка и Мультикарты

Для подключения интернет-эквайринга необходимо предоставить согласованный пакет документов на проверку сотрудниками банка. В перечень документов входит:

1. Учредительные документы (устав, учредительный договор (при наличии)
2. Сведения об имеющихся лицензиях
3. Приказ о назначении на должность единоличного исполнительного органа Предприятия
4. Документ, удостоверяющий личность Уполномоченного лица
5. Список участников, содержащий ФИО и даты рождения участников (по шаблону банка)
6. Карточка клиента (по шаблону банка, Word, PDF с подписью и печатью)

Учредительные документы предоставляются в виде:

- последней актуальной и действующей на дату предоставления в Банк редакции (если изменения в учредительные документы вносились путем регистрации новой редакции учредительных документов);
- действующей редакции с приложением листов изменений, зарегистрированных на дату предоставления документов в Банк, с отметкой о такой регистрации (если изменения в учредительные документы внесены и зарегистрированы в виде отдельных листов изменений).

После проверки документов партнеру присваивается MerchantID.

Подключение к тестовой среде осуществляется после присваивания партнеру MerchantID и состоит из следующих этапов:

1. Почта Банк направляет заявку на регистрацию тестового мерчанта в Мультикарту
2. Мультикарта заводит тестового мерчанта в своей системе
3. Мультикарта отправляет инструкцию по созданию сертификата
4. Мерчант формирует запрос на получение сертификата
5. Мультикарта формирует сертификат, отправляет мерчанту
6. Мерчант передает сертификат в Почту России по доступному каналу связи (почта, telegram)
7. Тестирование (по ранее согласованным тест-кейсам, которые проводятся на стороне мерчанта)
8. Подтверждение успешных результатов тестирования
9. Переход в боевую среду

После завершения тестирования и перехода в боевую среду необходимо также выполнить настройки:

1. Почта Банк направляет заявку на регистрацию боевого мерчанта в Мультикарту
2. Мультикарта заводит боевого мерчанта в своей системе
3. Мультикарта отправляет инструкцию по созданию боевого сертификата
4. Мультикарта настраивает коннективности (Мультикарта + CPS)
5. Мерчант формирует запрос на получение сертификата
6. Мультикарта формирует сертификат, отправляет мерчанту
7. Мерчант передает боевой сертификат в Почту России по доступному каналу связи (почта, telegram)
8. Пилотное тестирование сервиса
9. Подтверждение успешных результатов боевого тестирования
10. Выход в боевые транзакции

Подключение к сервисам Почта.Pay и Почта.ID

Подключение к сервису Почта.ID необходимо для проведения единой авторизации пользователей Почты Росси и бесшовному переходу в сервис Почта.Pay.
Почта.ID - сервис построенный на протоколе oauth2, при интеграции можно воспользоваться бесплатными публичными библиотеками. Со стороны сервисы есть поддержка открытых спецификаций по rfc.

Для подключения к сервисам заполняется таблица и передается команде Почта.Pay для проведения регистрации

Наименование (rus)	Фабрика одежды
Наименование (eng)	SpaceWear
Адрес портала (https)	https://space-wear.ru
Редирект для прохождения авторизации callBackPostId (https)	https://space-wear.ru/pochtaid/callbackAuth
Редирект отправки информации по платежу callbackPay (https) *** сервис на стороне мерчанта**	https://space-wear.ru/pochtapay/callbackPayInfo

Дополнительно применяются неизменные настройки:

Refresh_token	Не разрешается
Scope данных ***фиксированный список, может быть уменьшен. openid - обязателен**	`middleName firstName address email phone lastName openid`

По завершению регистрации, будет выданы: client_id и client_secret

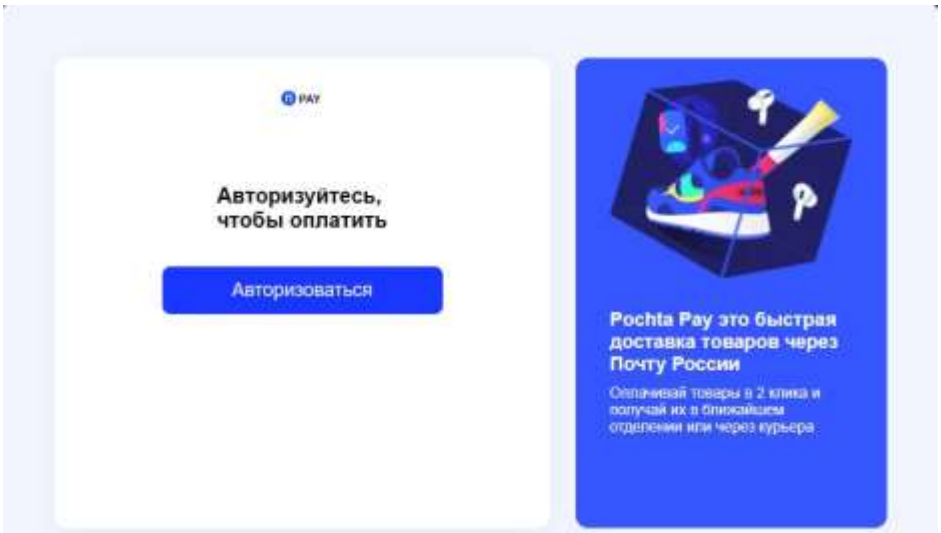
client_id - публичная информация не требует дополнительной защиты

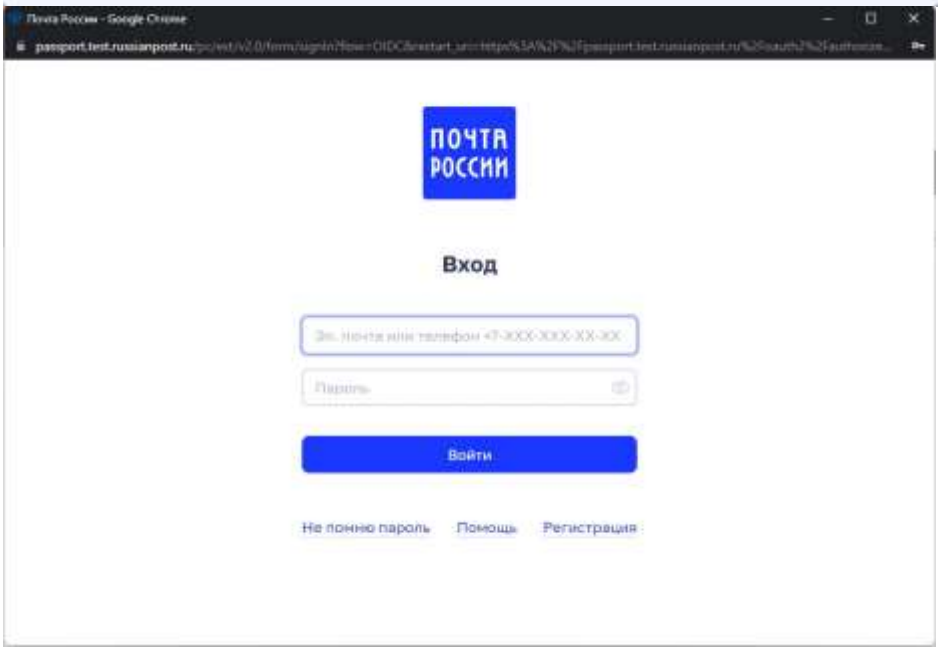
client_secret - требуется защитить и скрыть внутри сервиса, запрещена утечка во внешние сервисы и передача другим лицам

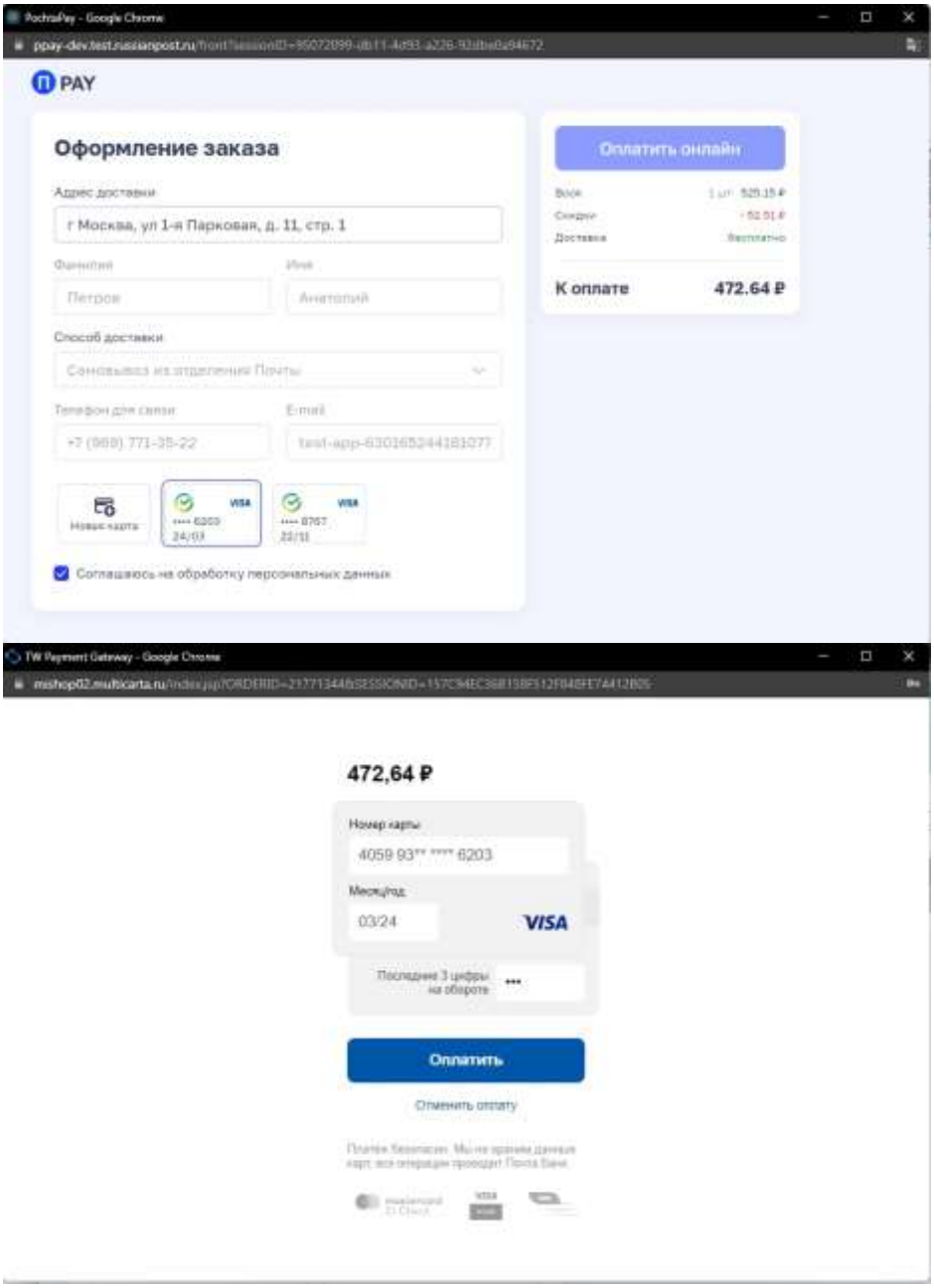
Сценарий - "Покупка и оформление доставки товара"

Общее описание сценария

- Пользователь попадает на страницу магазина, производится сбор состава корзины или точечный выбор товара.
- На странице оплаты отображается кнопка Почта.Pay (покупка и оформление доставки), которая конфигурируется с помощью PochtaPaySDK **(шаг 1)**
- По нажатию кнопки, происходит авторизация через сервис Почта.ID
- Прохождение авторизации завершается редиректом на callBackPostId, в результате которого мерчант получает набор маркеров и извлекает идентификатор пользователя, необходимого для старта процесса оплаты **(шаг 2)**
- Мерчант, далее, выполняет редирект на оплату, **в том же контексте**, где и производилась авторизация **(шаг 3) (обработчик callBackPostId, должен инициировать редирект на ссылку оплаты)**
- Почта.Pay отображает состав корзины для проведения оплаты
- Пользователь проводит оплату, выбирая адрес доставки и карту для оплаты
- Процесс оплаты проводится через сервис "Мультикарта"
- По завершению оплаты пользователь получает экран с успешным прохождением оплаты
- После проведения оплаты производится, в фоне, вызов сервиса мерчанта для информирования о статусе оплаты **(шаг 4)**


PAY
Авторизуйтесь,
чтобы оплатить
Авторизоваться
Pochta Pay это быстрая доставка товаров через Почту России
Оплачивай товары в 2 клика и получай их в ближайшем отделении или через курьера


Почта России - Google Chrome
passport.test.russianpost.ru
ПОЧТА РОССИИ
Вход
Войти
Не помню пароль
Помощь
Регистрация



Конфигурирование кнопки Почта.Pay (шаг 1)

Для интеграции кнопки, потребуется:

- Встроить на front-end приложения PochtaPaySDK

Пример:

необходимо добавить в тег <head />

```
<script src="https://{Почта.Pay-host}/PochtaPaySDK.js"></script>
```

- Библиотека PochtaPaySDK станет доступной и прикрепится к объекту window
- Заводится html-тег, в который будет смонтирована кнопка. В примере ниже кнопка смонтируется в тег с id = 'pochta-pay-button'.

Пример:

кнопка смонтируется в тег с id = 'pochta-pay-button'

```
const ppayButton = document.getElementById('pochta-pay-button');
```

- Перед рендером кнопки необходимо передать в PochtaPaySDK ряд параметров
  - redirectUrl = {callBackPostId}, указанный при подключении к сервису Почта.Pay, отвечает за редирект после успешного процесса авторизации
  - clientId = {client_id}, полученный после подключения к сервису Почта.Pay
  - orderId= {orderId}, добавляется информация по идентификатору корзины в SDK

Пример:

```
PochtaPaySDK.setAuthRedirectURI(redirectUrl);
PochtaPaySDK.setClientID(clientId);
PochtaPaySDK.setCart(orderId)
```

- Добавляется кнопка Почта.Pay в заранее объявленный тег

Пример:

```
PochtaPaySDK.addButton(ppayButton);
```

Пример общего исходного кода для формирования кнопки

```
window.onload = () => {
  fetch(host + 'demo/redirect-url', getOpts) //получение ссылки на редирект
на бэкенд мерчанта после успешной атворизации
    .then(function (response) {
      return response.json();
    })
    .then(function (body) {
      console.log('url from body: ' + body.url);
      const redirectUrl = body.url;
      const clientId = body.clientId;
      console.log('redirect url: ' + redirectUrl);

      const ppayButton = document.getElementById('pochta-pay-button');
      PochtaPaySDK.setAuthRedirectURI(redirectUrl);
      PochtaPaySDK.setClientID(clientId);

      PochtaPaySDK.addButton(ppayButton);
    });
};
```

Пример общего исходного кода для установки идентификатора корзины

```
let arr = [0.0];
let order;
function addToCart(productId) {
  let request = { userId: 1, productId: productId };
  postMarketRequest(request, 'demo/cart/add').then((body) => {
    order = body.cartId;
    console.log(order);
    PochtaPaySDK.setCart(order);
    arr.push(parseFloat(body.cost));
    let a = arr.reduce((x, y) => x + y, 0.0);
    document.getElementById('sum-label-1').innerHTML =
a.toFixed(2).toString();
  });
}
```

Обработка авторизации от Почта.ID (шаг 2)

- По факту прохождения авторизации, выполняется редирект для дальнейшего на получения набора маркеров с секретным параметром code и state (обработчик callBackPostId на стороне мерчанта)
- Выполняется получение набора маркеров: id_token/access_token
- Проверяется подпись id_token/access_token (**опционально**)
- Производится извлечение идентификатора пользователя, который потребуется для старта процесса оплаты
- Выполняется отслеживание времени использования маркера доступа

Обработка callback при авторизации

Запрос на callBackPostId - обработчик на стороне бэкенда мерчанта поддерживает обработку редиректа

Протокол	HTTPS
Тип	GET
Путь	{callBackPostId}

Query параметры:

Параметр	Обязательность	Описание	Пример
`code`	Да	Авторизационный код	`78bc036d-ffe2-4a01-8505-610986474450`
`state`	Нет	Идентификатор корзины мерчанта	`30a6101d-c69c-4a59-927f-29037448c3f9`

session_state	Нет	Дополнительный сессионный статус, не используется далее	QnNqT0llWGg0cWVuM0.....

Пример:

/pc/success?code=78bc036d-ffe2-4a01-8505-610986474450&state=30a6101d-c69c-4a59-927f-29037448c3f9&session_state=QnNqT0llWGg0cWVuM0tETnFSS1pKTThKNXlNYSBodHRwczovL3Bhc3Nwb3J0LWFwcC50ZXN0LnJ1c3NpYW5wb3N0LnJ1IGsyZU9ubmRBRDNfajF3aU1uTFRNckdCUmR3SlVXdlMxR3U3THZ5cXRPRE0ubE5RRGpnIEJSc2hEMUlVX18wPQ%3D%3D.BRshD1IU__0%3D

Получение набора маркеров

По факту получения авторизационного кода (code), получается набор маркеров.

В набор маркеров входит:

1. Маркер идентификации;
2. Маркет доступа;

Точный состав маркеров зависит от конфигурации приложения.

С маркерами доступа передается также дополнительная информация:

- Время жизни маркера доступа;
- Набор scope.

Запрос для получения набора маркеров

Отправляется запрос на уровне backend сервисов

Протокол	HTTPS
Тип	POST
Путь	/oauth2/token
Заголовки	Content-Type = "application/x-www-form-urlencoded"
	Authorization = 'Basic <base64({client_id}:{client_secret})>'

Form параметры, кодируются как **application/x-www-form-urlencoded**:

Параметр	Значение	Обязательность	Описание
grant_type	authorization_code	Да	Процесс Authorization Code, фиксированное значение
redirect_uri	{callBackPostId}	Да	callBackPostId, указанный при подключении к сервису
code	{code}	Да	Код авторизации

Ответ на получение набора

Заголовки	Content-Type = "application/json"
Тело	JSON

Положительный сценарий ответа

Атрибут	Обязательность	Описание
id_token	Нет	Маркер идентификации (id_token), далее для получения идентификатора пользователя
access_token	Да	Маркер доступа (access_token) для проведения операций в Почта.Pay
token_type	Да	Тип маркера доступа

scope	Нет	Область доступа, список разрешений для получения данных. Возможно сужение списка доступов, если превышены полномочия (см.справочник "Scope" и "Указание области доступа (scope)")
expires_in	Да	Время жизни маркера доступа

Пример:


```
{
  "id_token":
"eyJhbGciOiJSUzUxMiJ9.eyJpc3MiOiJodHRwczovL3Bhc3Nwb3J0LnRlc3QucnVzc2lhbnBvc3Q
ucnUvcGMvIiwic3ViIjoiMTU4Y2VkZGItNmVhYy00YjVkLWI3NzEtMDI4NzVmZGRiY2JlIiwiYXVk
IjpbIkJzak9JZVhoNHFlbjNLRE5xUktaSk04SjV5TWEiXSwiZXhwIjoxNTg4MDk5MjY0LCJpYXQiO
jE1ODgwOTc0NjQsImF1dGhfdGltZSI6MTU4ODA5NzQ2NCwiYW1yIjpbIlBXRCJdLCJhenAiOiJCc2
pPSWVYaDRxZW4zS0ROcVJLWkpNOEo1eU1hIiwiYXRfaGFzaCI6IjhReEY4aC1XaWkyQTBQazlfRlZ
uQUNSTWprUWlDSXhxY056RjlLNHFjcHM9In0.SmaGRVQ6PzaveVxgyGm0-
lUjFcTxpgEkLpDYbguiXU0EYyYDOgI7CQ_4Vi035HGTO-go3BOiNQAiEl1huQy0nJeB-
foYpZ1PZZZSf6E6P7CzJ5LDZg2i_qW-
KFou9CpcqQww7Xsl0LJFX1u0ctfHS0ntHSSomFiXrsZAmg4s8fNxwtialz7x_b8-
fCkfoQw1KGIpIWsM5vNXBwAm7xj8xIQv-
r6RA5MkIWYCnqfYwVizg8D7GOvuxT7LVdMEoG5ZWS55j12oXfqs3zUtEquJHVI2f7EnSyCyuBsfEZ
k2qOySq7ZUdG3lVBTMvWfkXvhFpmJVFaUlDbDkxcd6-
eNwby3a4IRL2M6NO30Nhg91JbfDbiyA2PNK7laAoqww1wc9Dco25qRxCtN4VE2HwSPfU6fy-
kBfFcDhu0J49RZR5vkDktE8H9mLNkYufef4svuR8NpYYuT2dUAknCGSoOrEpOGrU70H9OgytKTnao
ZA4rH81GAmna1Ien3YuYH-
6QbByP4aCM7jFF75vhEy3LSgrB9mnjWb2i7xYgw6XZEb0GrVKllnjlRwAojtjufRwg1v2D6BvHurF
F4k6pkvzBhcMhH9Dljbh7sTXv_jBBvliIcffljmVVNRBhBsmW9Ez_RPlDOrx-
mVhGYTnsYz2MVxIbUZYr6lqqXQq9XukI0kHBM",
  "access_token": "oBXg0ogUdLB6n7xB3B8MSeSB9wAM-
YYhKiOeaJeTXsYTeLGDAPuqlAjiWZ4Os1z6VxFlGydeVCtCmM410hqlNfaMXJ4hEVseWv5HmXST0y
Qmb0EUYcr3WwI10M400ge3zpa2BdX_zF6WHubGXvrSz2I00d_ovCb8Ky7weg0W03hXFS-
feo8SUh8sMT_LN6IARu2O3lBKZUykoIZ7E8CU",
  "token_type": "Bearer",
  "scope": "openid",
  "expires_in": 3600
```

}

Негативный сценарий ответа

Заголовки	Content-Type = "application/json"
Тело	JSON

Атрибут	Обязательность	Значение	Описание
error	Да	invalid_request	Запрос не содержит обязательные параметры
		invalid_client	Неверно задан идентификатор клиента
		invalid_grant	Авторизационный код • недействительный • просрочен • отозван
		unauthorized_client	Клиент не имеет права запрашивать таким методом (например, процессом Authorization Code)
		unsupported_grant_type	Тип авторизационного кода не поддерживается (см.описание "Формирование ссылки на форму входа.Описание параметров")
error_description	Нет	-	Описание ошибки
error_uri	Нет	-	Ссылка на страницу, для отображения ошибки. Рекомендуется к использованию, если система не имеет заготовленных страниц с ошибками

Пример:

```
{
  "error": "unauthorized_client",
  "error_description": "1101 The authenticated client is not authorized to
use this authorization grant type"
}
```

Негативный сценарий рекомендуется обработать и выдать пользователю информационное сообщение, например: Процесс прерван, обратитесь в техническую поддержку

Проверка подписи id_token

Проверка подписи носит **рекомендательный** характер, но с точки зрения **повышения безопасности** проведения операций **необходимо поддержать.**

- Для обоих маркеров id_token/access_token
  - Поскольку маркер идентификации является JWS токеном, то для его проверки можно использовать библиотеки, поддерживающие работу с JWT токенами и связанными с ними технологиями (JWS и JWE). Список библиотек можно найти на https://jwt.io/
  - Вызывается сервис jwks с публичными ключами
    - Получается первый ключ jwk, проверяется подпись id_token/access_token, используя алгоритм RSA512
  - Текущее время должно быть до значения указанного в атрибуте exp
- Дополнительно проверяется id_token и его содержание
  - aud атрибут содержит client_id приложения клиента
    - Если aud содержит несколько значений, то обязательно должен быть атрибут azp, который должен быть равен client_id приложения клиента.
  - Проверяется nonce, должен быть равен переданному на старте авторизации
  - Вычисляется у access_token контрольная сумма с использованием ключей JWKS и сверяется с at_hash id_token
    1. Контрольная сумма маркера доступа (access token)
       1. Вычисляется следующим образом:
          1. Хэш маркера доступа с помощью SHA-512.
          2. Берутся левые 256 бит и кодируются с помощью base64url
  - Если со времени аутентификации (значение атрибута auth_time) прошло слишком много времени (значение определяется приложением клиентом), то приложение должно запросить повторную аутентификацию.
- Для поддержки ротации, может быть выпущено несколько ключей, тогда проверяется по порядку каждый до получения статуса верификации

Запрос ключей jwk

Протокол	HTTPS
Тип	GET
Путь	/pc/ext/v1.0/jwks

Ответ на получение ключей

Успешный ответ

Http.code	**`keys[]`**
200	набор ключей

Пример:

```
{
  "keys": [
    {
      "kty": "RSA",
      "kid": "1",
      "use": "sig",
      "alg": "RS512",
      "n": "kOIyD3jkQuVwCnzLubG9E5fmz6KPNMyVdaPGTBQMTMMcllhuL-
yzDiwzzAzeSW1fnsrYHpCFLq1X6gnJy7Ywh9vjCNqbkQs8Id2Pr54bfs1ybczXS0eM1Y3KjpH-
Wf9OqcnCOv9swwaCdD1LwCZdtz9yvQc10BECM6R7CxIchHmty5mnsLqBWDVZzaTmFcJYWQ0THJ1zq
d2O8uJJkCfD-
5xpN9ypmd1OC7ycoswVnAKUSq8GuRBi8IwAJWTOWUMmHV6piYWx3whEragFf3yRgHtsexWd5dUYME
2fIoFpj5xflLYVUh_2IwuoHNPdkx63vo98-
Ic4TrLD4zNuM74KQEzKPHW95Zpum2FcOtufdhXOOzrSJ2eHI7jhNPGAif4pu2EMZdCYWbvC9Oxx8z
r0DqyPWxF-EcIqBvmYDNaKEpy8H6qRxAg7hUSHrr5AJcIBIii6cJ9_RB1SqMGMQztJtj4-
OqNUiVUn1uSAxQO8oHTlh8d1Dep36NZNQroPFHCvs6bdUx6UJ1K4KzSHROQtvq-8RXT-
HxxJrcqNAO0-
diNA80KdM4RQPxmE2awZDm7xDHx68jnj14S3C5MGv6cymU9ZoAZTlh97omJh7cGJmCC5mi43c5k3U
SVRWfOJ0kUqh1XOKHCcVbre2-WoSTqogQcSFUE7vVIWTOZi55K2rvU",
      "e": "AQAB",
      "primary": true
    }
  ]
}
```

Извлечение идентификатора пользователя

Маркер идентификации (id_token) является JWT токеном, подписанным с использованием алгоритма RSA512 (технология JWS).

Содержит в себе следующий состав атрибутов:

Атрибут	**Тип**	**Обязательность**	**Описание**
`iss`	Строка	Да	https://passport.pochta.ru/pc/
`sub`	Строка	Да	Идентификатор пользователя
`aud`	Массив[Строка]	Да	Элементы массива это значения client_id приложений, для которых выпущен токен. client_id приложения, запросившего аутентификацию, должен обязательно присутствовать в массиве.
`exp`	Значение JSON number представляющее число секунд прошедших с 1970-01-01T0:0:0Z UTC	Да	Время прекращения действия токена.
`iat`	Значение JSON number представляющее число секунд прошедших с 1970-01-01T0:0:0Z UTC	Да	Время выпуска токена.
`auth_time`	Значение JSON number представляющее число секунд прошедших с 1970-01-01T0:0:0Z UTC	Нет	Время проведения аутентификации пользователя.
`azp`	Строка	Да	client_id приложения, запросившего аутентификацию, для которого выпущен токен.
`at_hash`	Строка	Да	Контрольная сумма маркера доступа (access token). Вычисляется следующим образом: 1. Хэш маркера доступа с помощью SHA-512. 2. Берутся левые 256 бит и кодируются с помощью base64url.

Для выполнения старта процесса оплаты потребуется использовать sub - Идентификатор пользователя

Отслеживание времени жизни маркера доступа

Маркер доступа (access_token) дополняется следующей информацией.

Атрибут	Описание
`scope`	Список разрешенных доступов к данным пользователям
`token_type`	Тип токена, применяется при работе с сервисами системы
`expires_in`	Дублируется время жизни маркера доступа, в сек

Пример:


```
"access_token": "SmV3Gsnst_cJNpN-
gzAzqm6BRQuSa7ocaQeVL2RiUr5plYMldjF82FebfHc10g4mHtFw6j6gibf3L9kWeqftwUBHyfwae
c3BFrEb0gc10043wUSiwFju-
EWhKGTyWKTSZEMH002E77B2uFrE8g86kETqGXbRQQxkB24Deg0I09zZnsDO8V15oo3P7BvAd4a7Gb
2SY7S4qWctNwQLtvY2",
"scope": "email openid",
"token_type": "Bearer",
"expires_in": 1800
```


Важно отслеживать время жизни маркера доступа, если вышел срок, то рекомендуется перезапустить процесс, например: выдать страницу с информацией, что время сессии истекло и кнопкой перезапуска.

Старт процесса оплаты (шаг3)

- Выполняется получение ссылки на оплату
- Выполняется редирект на полученную ссылку на оплату в том же контексте, где проходила операция получения набора маркеров

Получение ссылки на старт процесса оплаты

Запрос ссылки на старт оплаты

Протокол	HTTPS
Тип	POST
Путь	/api/v1/auth/pay

Заголовки	Content-Type = "application/json"
	Authorization = 'Bearer <access_token>'

Query параметры

Параметр	Значение	Обязательность	Описание
userId	sub из id_token	Да	Идентификатор пользователя Почта.ID

Тело запроса

Атрибут		Обязательность	Тип	Описание
`callbackUrl`		Да	Строка	Абсолютный путь для ответа о результате платежа, указанный при подключении к сервису Почта.Pay в **callbackPay**
`orderId`		Да	Строка	Идентификатор корзины мерчанта
`merchantId`		Да	Строка	Идентификатор мерчанта в Мультикарте
`discountPrice`		Да	Строка	Размер скидки
`totalPrice`		Да	Строка	Итоговая цена со скидкой
`products[]`		Да	Массив	Массив товаров в корзине
	`productName`	Да	Строка	Название товара
	`price`	Да	Строка	Цена товара, формат = *.00
	`quantity`	Да	Строка	Количество позиций данного товара
`hold`		Да	Булевое	Холдирование
`description`		Да	Строка	Комментарий к платежу

Пример:

```
{
  "callbackUrl": "string",
  "orderId": "string",
  "merchantId": "string",
  "discountPrice": "string",
  "totalPrice": "string",
  "products": [
    {
      "productName": "string",
      "price": "string",
      "quantity": "string"
    }
  ],
  "hold": boolean,
  "description": "string"
}
```

Ответ на получение ссылки

Заголовки	Content-Type = "application/json"
Тело	JSON

Тело ответа

Атрибут	Обязательность	Тип	Описание
`redirectUrl`	Да	Строка	**Ссылка на страницу оплаты** со случайно сгенерированным идентификатором сессии время жизни которого ограниченно

Пример:

```
{
  "redirectUrl": "string"
```

}

Обработка получения информация по платежу (шаг4)

- Мерчант поддерживает сервис получения информации по платежу, ссылка на сервис указывается в момент старта оплаты

Запрос на обработчик получения информации по платежу

Протокол	HTTPS
Тип	POST
Путь	{callbackPay} - указанный в при старте оплаты
Заголовки	Content-Type = "application/json"

Тело запроса

Атрибут	Обязательность	Тип	Описание
`statusPay`	Да	Строка	Статус платежа(при статусе отличном от **APPROVED** полей **`address, name, surname, phone`** не будет)
`address`	Да	Строка	Адрес пользователя
`name`	Нет	Строка	Имя
`surname`	Нет	Строка	Фамилия
`phone`	Нет	Строка	Номер телефона
`merchantOrderId`	Да	Строка	Идентификатор корзины мерчанта(`orderId`)
`paymentOrderId`	Да	Строка	Идентификатор платёжного заказа в сервисе Мультикарты
`paymentSessionId`	Да	Строка	Идентификатор сессии в сервисе Мультикарты

Допустимый ответ на информацию по платежу

http.code	**Комментарий**
200	Принято