

# Описание программного интерфейса (API) платформы DeStream

## 1. Назначение программного интерфейса (API)

Программный интерфейс (API) DeStream позволяет взаимодействовать с платформой для выполнения следующих действий.

1. Получение информации о платежах (чаевых) пользователю платформы DeStream.
2. Инициирование платежа (перечисления чаевых) пользователю платформы DeStream.
3. Получение информации о пользователе платформы DeStream.
4. Регистрация пользователя на платформе DeStream.
5. Получение уведомлений о событиях:
  - изменение статуса платежа (отправки чаевых) пользователю платформы DeStream;
  - информация в профиле пользователя платформы DeStream изменилась.

## 2. Требования к вызывающей системе

Для подключения внешней системы к платформе DeStream необходимо выполнить следующие действия.

1. Сообщить администрации платформы информацию о подключаемой системе, и в случае одобрения заявки получить параметры **ClientId** и **ClientSecret**, которые будут использоваться для идентификации и аутентификации запросов к программному интерфейсу.
2. Сообщить **AuthRedirectURL** — URL, на который будет выполняться переадресация пользователя (redirect) в случае успешной аутентификации (OAuth 2) при добавлении связи подключаемой системы с его учётной записью на платформе DeStream.
3. Сообщить **PaymentCallbackUrl** — URL метода получения уведомлений о статусе платежей (если требуется получение callback-уведомлений).
4. Сообщить **UserDataChangedCallbackUrl** — URL метода получения уведомлений об изменении информации в профиле пользователя платформы DeStream (если требуется получение callback-уведомлений).

## 3. Общие сведения о программном интерфейсе (API)

Протокол взаимодействия: **HTTPS**.

Формат представления данных: **JSON**.

В случае успешного выполнения запрошенной операции возвращается HTTP-ответ с кодом **200** и результатом выполнения операции в теле ответа.

В случае возникновения ошибок возвращается HTTP-ответ со значением кода больше или равным **400** и следующей структурой данных в теле ответа, описывающей причину ошибки.

```
{
  code: number;           //код ошибки
  error_message: string; //текстовое описание ошибки
}
```

Описание кодов ошибок:

Код ошибки	Описание
<b>1</b>	Не удалось аутентифицировать вызывающую систему, возможно, данные аутентификации содержат ошибку или доступ для данной вызывающей системы заблокирован.

Код ошибки	Описание
2	Переданный токен доступа пользователя ( <code>access_token</code> или <code>refresh_token</code> ) некорректен или недействителен. Необходимо обновить <code>access_token</code> , используя ранее полученный токен обновления <code>refresh_token</code> или выполнить аутентификацию пользователя заново для получения новой пары токенов.
3	Невозможно зарегистрировать нового пользователя, так как переданный адрес электронной почты (e-mail) уже зарегистрирован у другого пользователя платформы.
4	Вызывающая система некорректно сконфигурирована, например, заданный <b>AuthRedirectURL</b> отличается от того, который передан в запросе на аутентификацию или получение токена.
5	Передано некорректное содержимое в X-API-Signature
6	Превышен допустимый интервал времени между датой и временем в заголовке X-API-RequestDate запроса и временем получения такого запроса сервером для обработки.
1000	Общий код ошибки для случаев, если ошибка не описана в списке кодов ошибок. Подробности передаются в поле <b>error_message</b> .

## 4. Ограничение доступа

### 4.1. Ограничение доступа к программному интерфейсу (API)

Для доступа к программному интерфейсу (API) DeStream требуется идентификация и аутентификация вызывающей системы.

Идентификация осуществляется по предварительно выданному вызывающей системе значению **ClientId**.

Аутентификация осуществляется по предварительно выданному вызывающей системе значению **ClientSecret**.

Параметры **ClientId** и **ClientSecret** передаются в одноимённых GET-параметрах вызова метода, или используются для формирования значений дополнительных HTTP-заголовков **X-API-ClientId** и **X-API-Signature**.

### 4.2. Ограничение доступа к пользовательским данным

Некоторые методы программного интерфейса, которые предоставляют данные пользователя, требуют выдачи пользователем платформы DeStream разрешения на доступ к его данным. Предоставление доступа подтверждается выдачей вызывающей системе токена доступа, который затем должен использоваться при доступе к соответствующим методам программного интерфейса.

Токен выдаётся в результате выполнения процедуры аутентификации по стандарту OAuth 2.

Для её выполнения платформе DeStream предварительно должно быть предоставлено значение параметра **AuthRedirectUrl** — URL на странице вызывающей системы для переадресации пользователя после прохождения аутентификации.

Сценарий выполнения аутентификации следующий.

1. Со страницы вызывающей системы, на которой выполняется привязка пользователя платформы DeStream к вызывающей системе, перенаправить пользователя на URL платформы DeStream.

GET

*[https://destream.net/oauth2/authorize?response\\_type=code&client\\_id=<ClientId>&redirect\\_uri=<AuthRedirectUrl>&scope=profile+tips&state=<случайная последовательность>](https://destream.net/oauth2/authorize?response_type=code&client_id=<ClientId>&redirect_uri=<AuthRedirectUrl>&scope=profile+tips&state=<случайная последовательность>)*

Где

**response\_type=code** — обозначает начало процесса обмена данными для получения доступа;

**<ClientId>** — идентификатор вызывающей системы;

**<AuthRedirectUrl>** — URL переадресации пользователя после прохождения процедуры аутентификации, должен совпадать с предварительно предоставленным **AuthRedirectUrl**;

**profile+tips** — обозначение блоков данных пользователя, к которым запрашивается доступ: информация из профиля (profile) и информация о платежах (tips), для разделения идентификаторов доступа используется знак «+»;

**<случайная последовательность>** — любая уникальная (случайная) буквенно-цифровая последовательность, которая будет передана вызывающей стороне при переадресации пользователя после завершения процедуры аутентификации в неизменном виде, должна быть проверена на стороне вызывающей системы, необходима для исключения CSRF-атак.

В результате выполнения метода пользователю будет предложено пройти процедуру аутентификации на платформе DeStream, а также будет запрошено разрешение на предоставление данных вызывающей системе.

Если пользователь выполнил вход и предоставил разрешения для внешней системы, он будет перенаправлен на указанную в параметре *redirect\_url* страницу вызывающей системы. В запросе будет передан параметр **code** со значением кода авторизации, который впоследствии должен использоваться для получения токена доступа. Также будет передан параметр **state** со значением **unique token**, проверка неизменности которого необходима для защиты от CSRF-атак.

В случае возникновения ошибки будет либо возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)», либо выполнено перенаправление на указанную в параметре *redirect\_url* страницу вызывающей системы с указанием в запросе параметра **error**, содержащего текст ошибки для отображения пользователю.

## 2. Получить токен доступа (access token) посредством выполнения запроса

POST

```
https://destream.net/api/v2/oauth2/token?grant_type=authorization_code&client_id=<ClientId>&client_secret=<ClientSecret>&redirect_uri=<AuthRedirectUrl>&code=<code>
```

Где

**grant\_type=authorization\_code** — обозначает, что для доступа будет использоваться код авторизации (authorization\_code);

**<ClientId>** — идентификатор вызывающей системы;

**<ClientSecret>** — «секрет» вызывающей системы;

**<AuthRedirectUrl>** — URL переадресации пользователя после прохождения процедуры аутентификации, должен совпадать с предварительно предоставленным **AuthRedirectUrl**, используется для контроля целостности;

**<code>** — код авторизации, который был получен в параметре *code* при возврате на адрес вызывающей системы после успешного выполнения метода *authorize* (первый шаг).

Должен быть передан HTTP-заголовок *Content-Type* со значением *application/x-www-form-urlencoded*.

**Content-Type: application/x-www-form-urlencoded**

В случае успешного выполнения метода в ответе будет получен JSON-объект следующего вида.

```
{
  access_token: string;
  refresh_token: string;
  expires_in: number;
  scope: string;
  token_type: string;
```

}

Где

***access\_token*** — содержит токен доступа к методам, возвращающим данные пользователя, токен имеет ограниченное время действия, после истечения времени его действия токен должен быть обновлён, доступ к данным с использованием истекшего токена будет прекращён — будет возвращаться соответствующая ошибка при вызове методов доступа к пользовательским данным;

***refresh\_token*** — содержит токен для обновления токена доступа после истечения времени его действия;

***expires\_in*** — содержит время в секундах, через которое истечёт время действия токена доступа;

***scope*** — содержит обозначение блоков данных пользователя, к которым предоставлен доступ посредством выданного токена доступа;

***token\_type*** — содержит тип токена доступа.

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)». В частности, если код авторизации истёк или не действителен, будет возвращён ответ с HTTP-кодом 401 Unauthorized. В этом случае необходимо заново выполнить метод запроса доступа у пользователя `oauth2/authorize` (шаг 1).

3. В случае истечения действия токена доступа необходимо выполнить его обновление посредством вызова следующего метода.

POST

`https://destream.net/api/v2/oauth2/token?`

`grant_type=refresh_token&client_id=<ClientId>&client_secret=<ClientSecret>&scope=<scope>&refresh_token=<refresh_token>`

Где

***grant\_type=refresh\_token*** — обозначает, что для доступа будет использоваться токен обновления токена доступа (refresh token);

***<ClientId>*** — идентификатор вызывающей системы;

***<ClientSecret>*** — «секрет» вызывающей системы;

***<scope>*** — значение поля *scope*, полученного в JSON-объекте в ответе на вызов метода `token` (шаг 2) вместе с токеном доступа, значение которого требуется обновить;

***<refresh\_token>*** — значение поля *refresh\_token*, полученного в JSON-объекте в ответе на вызов метода `token` (шаг 2) вместе с токеном доступа, значение которого требуется обновить.

Должен быть передан HTTP-заголовок *Content-Type* со значением *application/x-www-form-urlencoded*.

**Content-Type: application/x-www-form-urlencoded**

В случае успешного выполнения метода в ответе будет получен JSON-объект того же вида, что и при выполнении метода на шаге 2 с обновлённым значением токена доступа. В случае ошибки результат полностью идентичен описанию вызова метода на шаге 2.

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)». В частности, если токен обновления истёк или не

действителен, будет возвращён ответ с HTTP-кодом 401 Unauthorized. В этом случае необходимо заново выполнить метод запроса доступа у пользователя `oauth2/authorize`.

## 5. Методы программного интерфейса (API)

### 5.1. Пользователи

#### 5.1.1. Создание пользователя на платформе

Метод создания пользователя позволяет автоматически выполнить создание пользователя на платформе DeStream по его e-mail. Пользователь может быть создан только в том случае, если переданный e-mail не используется у уже существующих пользователей платформы.

Метод
<code>POST</code>
URL
<code>https://destream.net/api/v2/users/register</code>
Заголовки запроса
<p><code>X-Api-ClientId</code> содержит <b>ClientId</b>.</p> <p><code>X-Api-RequestDate</code> содержит дату и время формирования запроса в формате ISO8601.</p> <p><code>X-Api-Signature</code> содержит контрольную сумму содержимого тела запроса по формуле SHA-512(ClientId+X-Api-RequestDate+ClientSecret), то есть берётся SHA-512 (ХЭШ) от конкатинации <b>ClientId</b>, даты и времени запроса и значения <b>ClientSecret</b> системы, которая создаёт платёж.</p>
Тело запроса
<pre>{   email: string; // e-mail создаваемого пользователя платформы }</pre>
Результат выполнения
<pre>{   data: {     user_id: string;           // Уникальный идентификатор созданного пользователя     email_confirmed: boolean; // Подтверждён ли адрес электронной почты пользователем     email: string;           // e-mail созданного пользователя     nickname: string;        // Никнейм созданного пользователя     user_token: {       access_token: string;   // Токен доступа к информации о созданном пользователе       refresh_token: string;  // Токен обновления токена доступа       expires_in: number;    // Время действия токена доступа       scope: string;         // Блоки данных пользователя, к которым токен предоставляет доступ       token_type: string;    // Тип токена     }   } }</pre>
В случае ошибки
<p>В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)».</p> <p>В частности, если пользователь с переданным e-mail уже существует на платформе, будет возвращён ответ с HTTP-кодом <i>409 Conflict</i>.</p>

Значение полей из объекта `user_token` необходимо использовать для получения информации о созданном пользователе посредством соответствующих методов.



### 5.1.2. Получение информации из профиля пользователя платформы

Метод получения информации из профиля пользователя возвращает данные о пользователе и его настройки на платформе.

#### Метод

GET

#### URL

<https://destream.net/api/v2/users>

#### Заголовки запроса

*X-API-ClientId* содержит **ClientId**.

*Authorization* содержит строку вида "**token\_type+пробел+access\_token**" (без кавычек и плюсов), где значения *token\_type* и *access\_token* берутся из одноимённых полей объекта токена доступа, возвращаемого методом *auth2/token* или методом создания пользователя *users/register*.

#### Тело запроса

Не передаётся (пустое).

#### Результат выполнения

```
{
  data: {
    user_id: string;           // Уникальный идентификатор пользователя
    email_confirmed: boolean; // Подтверждён ли адрес электронной почты пользователем
    email: string;           // e-mail пользователя
    nickname: string;        // Никнейм пользователя
    limits: [                // Массив объектов со значениями максимальной и минимальной суммы платежа (чаевых)
      {                      // для каждой установленной валюты по одному элементу массива на валюту
        currency: string;    // EUR, RUB, USD
        min: number;         // Значение минимальной суммы платежа (чаевых)
        max: number;         // Значение максимальной суммы платежа (чаевых)
      }
    ]
  }
}
```

#### В случае ошибки

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)».

В частности, если токен доступа истёк или не действителен, будет возвращён ответ с HTTP-кодом **401 Unauthorized**. Необходимо выполнить метод обновления токена доступа с помощью токена обновления (*refresh\_token*) методом *oauth2/token*.



### 5.1.3. Получение информации о полученных пользователем платежах (чаевых)

Метод получения информации о полученных пользователем платежах возвращает список блоков данных о платежах пользователю, начиная с самого позднего.

#### Метод

GET

#### URL

`https://destream.net/api/v2/users/tips?offset=<number of items to skip>&limit=<number of items per response>&after_date=<date in ISO8601 format>`

#### Параметры запроса

**<number of items to skip>** — количество элементов списка платежей, которые необходимо пропустить от начала списка, отсортированного по убыванию даты платежа (от позднего к раннему); необязательный параметр, если не передан, значение равно 0, то есть список будет содержать элементы, начиная с самого позднего платежа;

**<number of items per response>** — количество элементов в списке платежей, которое требуется вернуть в ответе на запрос; необязательный параметр, если не передан, значение равно 10, то есть ответ будет содержать список из десяти элементов; максимально возможное значение равно 30;

**<date in ISO8601 format>** — дата и время в формате ISO8601, платежи, выполненные в которую или позже, включить в список; необязательный параметр, если не передан, возвращаются все платежи, начиная с самого позднего.

#### Заголовки запроса

*X-Api-ClientId* содержит **ClientId**.

*Authorization* содержит строку вида "**token\_type+пробел+access\_token**" (без кавычек и плюсов),

где значения *token\_type* и *access\_token* берутся из одноимённых полей объекта токена доступа, возвращаемого методом *auth2/token* или методом создания пользователя *users/register*.

#### Тело запроса

Не передаётся (пустое).

#### Результат выполнения

```
{
  data: [{
    user_id: string;           // Уникальный идентификатор пользователя получателя платежа
    sender: string;           // Никнейм отправителя платежа
    payment_id: string;       // Идентификатор платежа
    amount: number;          // Сумма платежа
    currency: string;         // Валюта платежа EUR, RUB, USD
    message: string;         // Сообщение от отправителя платежа (чаевых)
    date: string;            // Дата и время платежа в формате ISO8601
    additional_data: string; // Дополнительные параметры в формате «параметр=значение»
                             // которые были получены при создании платежа
  }],
  total: number;             // Общее количество платежей пользователю (начиная с after_date, если передан)
  response_date: string;     // Дата и время ответа сервера в формате ISO8601.
}
```

#### В случае ошибки

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)».

#### 5.1.4. Подписка на оповещения о полученных пользователем платежах

Подписка на оповещения о полученных пользователем платежах позволяет получать информацию о поступивших пользователю платежах (чаевых) в режиме реального времени через web socket. Публикуются только успешно выполненные платежи и платежи через PayPal в состоянии обработки. Для взаимодействия можно использовать клиентскую библиотеку используемого языка программирования для взаимодействия с SignalR.

#### Метод

GET

#### URL

Адрес точки доступа для получения оповещений

`https://destream.net/ws/v2/donations?client_id=<ClientId>&access_token=<UserAccessToken>`

#### Параметры запроса

**<ClientId>** — идентификатор вызывающей системы.

**<UserAccessToken>** — токен доступа, возвращаемый в поле access\_token объекта user\_token, получаемого в результате выполнения метода `oauth2/token`.

#### Результат выполнения

Событие **"donationReceived"** с данными в формате json:

```
{
  user_id: string;           // Уникальный идентификатор пользователя получателя платежа
  sender: string;           // Никнейм отправителя платежа
  payment_id: string;       // Идентификатор платежа
  amount: number;          // Сумма платежа
  currency: string;        // Валюта платежа EUR, RUB, USD
  message: string;         // Сообщение от отправителя платежа (чаевых)
  date: string;            // Дата и время платежа в формате ISO8601
  additional_data: string; // Дополнительные параметры в формате «параметр=значение»
}
```

#### В случае ошибки

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)».

## 5.2. Платежи

### 5.2.1. Создание платежа

Метод создания платежа формирует URL для переадресации на форму выбора метода оплаты и выполнения платежа (отправки чаевых) пользователю платформы. Выполняющий отправку чаевых пользователь вызывающей системы должен быть переадресован на полученный в результате выполнения метода URL.

Полученный URL действителен в течение 24 часов (время может быть изменено).

#### Метод

POST

#### URL

<https://destream.net/api/v2/payments>

#### Заголовки запроса

*X-API-ClientId* содержит **ClientId**.

*X-API-RequestDate* содержит дату и время формирования запроса в формате ISO8601.

*X-API-Signature* содержит контрольную сумму содержимого тела запроса по формуле SHA-512(*ClientId*+*X-API-RequestDate*+*ClientSecret*), то есть берётся SHA-512 (ХЭШ) от конкатинации **ClientId**, даты и времени запроса и значения **ClientSecret** системы, которая создаёт платёж.

#### Тело запроса

```
{
  user_id: string;           // Уникальный идентификатор пользователя получателя платежа
  amount: number;
  currency: string;         // EUR, RUB, USD
  message: string;
  success_url: string;      // URL, на который будет выполнена переадресация в случае успешного платежа
  fail_url: string;         // URL, на который будет выполнена переадресация в случае проблемного платежа
  additional_data: string;  // любой набор дополнительных параметров в формате «параметр=значение».
                           // максимальная длина – 100 символов
                           // которые будут возвращены в неизменном виде при запросе
                           // информации о платеже или в уведомлении об изменении его статуса
}
```

#### Результат выполнения

```
{
  data: {
    payment_url: string;     // URL формы выбора способа оплаты и выполнения платежа
    payment_id: string;     // идентификатор платежа
    user_id: string;        // идентификатор пользователя, который получает оплату (чаевые)
  }
}
```

#### В случае ошибки

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)».

### 5.2.2. Получение информации о платежах

Метод получения информации о платежах возвращает список блоков данных о платежах, ранее созданных вызывающей системой, начиная с самого позднего.

#### Метод

GET

#### URL

`https://destream.net/api/v2/payments?offset=<number of items to skip>&limit=<number of items per response>&after_date=<date in ISO8601 format>&payment_id=<id1>,<id2>...`

#### Параметры запроса

**<number of items to skip>** — количество элементов списка платежей, которые необходимо пропустить от начала списка, отсортированного по убыванию даты платежа (от позднего к раннему); необязательный параметр, если не передан, значение равно 0, то есть список будет содержать элементы, начиная с самого позднего платежа;

**<number of items per response>** — количество элементов в списке платежей, которое требуется вернуть в ответе на запрос; необязательный параметр, если не передан, значение равно 10, то есть ответ будет содержать список из десяти элементов; максимально возможное значение равно 30;

**<date in ISO8601 format>** — дата и время в формате ISO8601, платежи, выполненные в которую или позже, включить в список; необязательный параметр, если не передан, возвращаются все платежи, начиная с самого позднего.

**<id1>,<id2>...** — список разделённых запятой значений идентификаторов платежей *payment\_id*, возвращаемых методом создания платежа, которые должны быть возвращены в ответе; максимально можно указать до 20 идентификаторов; необязательный параметр, если не передан, будут возвращены все платежи.

#### Заголовки запроса

*X-Api-ClientId* содержит **ClientId**.

*X-Api-RequestDate* содержит дату и время формирования запроса в формате ISO8601.

*X-Api-Signature* содержит контрольную сумму содержимого тела запроса по формуле SHA-512(ClientId+X-Api-RequestDate+ClientSecret), то есть берётся SHA-512 (ХЭШ) от конкатинации **ClientId**, даты и времени запроса и значения **ClientSecret** системы, которая создаёт платёж.

#### Тело запроса

Не передаётся (пустое).

#### Результат выполнения

```
{
  data: [{
    user_id: string;           // Уникальный идентификатор пользователя получателя платежа
    sender: string;           // Никнейм отправителя платежа
    payment_id: string;       // Идентификатор платежа
    amount: number;          // Сумма платежа
    currency: string;         // Валюта платежа EUR, RUB, USD
    message: string;         // Сообщение от отправителя платежа (чаевых)
    date: string;            // Дата и время платежа в формате ISO8601
    additional_data: string;  // Дополнительные параметры в формате «параметр=значение»
                              // которые были получены при создании платежа
    transaction_id: string;   // Идентификатор транзакции оплаты
    transaction_status_code: number; // Код статуса платежа:
                              // -1 — транзакция отклонена (DECLINED)
                              // 0 — статус новой транзакции (NEW)
                              // 1 — транзакция в обработке (PROCESSING)
                              // 2 — транзакция выполнена успешно (COMPLETED)
    transaction_status_text: string; // Текст статуса платежа
  }],
  total: number;             // Общее количество платежей пользователю (начиная с after_date, если передан)
  response_date: string;     // Дата и время ответа сервера в формате ISO8601.
}
```

### В случае ошибки

В случае возникновения ошибки будет возвращён ответ в соответствии с описанием в разделе «3. Общие сведения о программном интерфейсе (API)».

## 6. Уведомления

### 6.1. Общие сведения

Уведомления предназначены для информирования сопряжённых систем об изменениях на платформе DeStream. Реализованы уведомления об изменении информации о пользователях и о новых состояниях платежей.

Уведомления отправляются в виде HTTP(S) POST-запроса с типом содержимого Content-Type: application/json.

В дополнительном заголовке *X-Signature* отправляемого POST-запроса содержится контрольная сумма содержимого тела запроса по формуле  $\text{SHA-512}(\text{body} + \text{ClientSecret})$ , то есть берётся SHA-512 хэш от конкатинации тела запроса и значения **ClientSecret** системы, которой отправляется уведомление.

Контрольную сумму необходимо проверять посредством вычисления контрольной суммы на стороне получателя и сравнения её с контрольной суммой, переданной в заголовке. Равенство указанных значений будет подтверждать неизменность содержимого запроса при передаче.

Уведомление считается доставленным, если в ответ на запрос был получен ответ со статусом HTTP 200 ОК.

### 6.2. Уведомления об изменении статуса платежа

Уведомление об изменении статуса платежа отправляется во внешнюю систему посредством вызова **PaymentCallbackUrl**.

В теле запроса-вызова метода передаётся следующая структура данных.

```
{
  data: {
    user_id: string;           // Уникальный идентификатор пользователя получателя платежа

    sender: string;           // Никнейм отправителя платежа
    payment_id: string;       // Идентификатор платежа
    amount: number;          // Сумма платежа
    currency: string;        // Валюта платежа EUR, RUB, USD
    message: string;         // Сообщение от отправителя платежа (чаевых)
    date: string;            // Дата и время платежа в формате ISO8601
    additional_data: string;  // Дополнительные параметры в формате «параметр=значение»

    transaction_id: string;   // Идентификатор транзакции оплаты
    transaction_status_code: number; // Код статуса платежа:
                                // -1 — транзакция отклонена (DECLINED)
                                // 1 — транзакция в обработке (PROCESSING)
                                // 2 — транзакция выполнена успешно (COMPLETED)
    transaction_status_text: string; // Текст статуса платежа
  }
}
```

### 6.3. Уведомление об изменении информации о пользователе

Уведомление об изменении информации о пользователе отправляется во внешнюю систему посредством вызова **UserDataChangedCallbackUrl**. В теле запроса-вызова метода передаётся следующая структура данных.

```
{
  data: {
    user_id: string;           // Уникальный идентификатор пользователя
    email_confirmed: boolean; // Подтверждён ли адрес электронной почты пользователем
    email: string;            // e-mail пользователя
    nickname: string;         // Никнейм пользователя
    limits: [                 // Массив объектов со значениями максимальной и минимальной суммы платежа (чаевых)
      {                       // для каждой установленной валюты по одному элементу массива на валюту
        currency: string;     // EUR, RUB, USD
        min: number;          // Значение минимальной суммы платежа (чаевых)
        max: number;          // Значение максимальной суммы платежа (чаевых)
      }
    ]
  }
}
```

**Примечание.** В текущей реализации уведомления отправляются только по факту подтверждения пользователем адреса электронной почты.