



## Open Banking API manual

---

Version 1.0  
November 2020

## Versions

Version	Date	Changes
1.0	November, 2020	First edition

## Table of Contents

Versions.....	2
Table of Contents .....	3
1 Introduction .....	4
2 Definitions .....	5
3 Registration .....	6
4 Security.....	7
4.1 Authorization.....	7
4.1.1 Authorization Code Grant (Strong customer authentication – SCA).....	7
4.1.2 Access Token Request (Long term renewable token) .....	9
4.1.3 Refreshing an Access Token .....	10
4.1.4 Client Credentials Grant (Short term non-renewable token) .....	12
5 AIS and PIIS .....	14
5.1 Authorization for AIS and PIIS operations .....	14
5.2 Refreshing Access Token.....	14
5.3 Get Account List .....	15
5.4 Get Account Details.....	17
5.5 Get Transaction History .....	20
5.6 Get Transaction Details.....	24
5.7 Check Balance .....	27
6 PIS .....	30
6.1 Standard Payment Initiation .....	30
6.1.1 Step1: Obtain Access Token .....	30
6.1.2 Step2: Initiate payment.....	30
6.1.3 Step3: Payment Authorization (redirect to TrustPay) .....	34
6.1.4 Step4: Getting token that will be used to submit the payment.....	34
6.2 Standard Payment Submission .....	34
6.3 Payment Order Status .....	35

## 1 Introduction

Based on EU Directive on Payment Services (PSD 2) effective from 13. 1. 2018 TrustPay is publishing this Open Banking API to enable Third Party Payment Service Providers (TPPSP) to get access, after client's consent, to payment account information and/or to initiate payment orders from payment accounts of TrustPay clients.

PSD2 defines three types of services, that should be made available for TPPSPs:

- Payment Initiation Services (PIS)
- Account Information Services (AIS)
- Payment Instrument Issuer Service (PIIS)

To become a TPPSP that can connect to TrustPay payment accounts, the organization has to be a licensed TPPSP. Subsequently, it needs to undertake a registration process to obtain the necessary credentials to be able to connect and communicate with TrustPay servers and use the Open Banking API. The registration process and the available API calls are described in the next sections.

## 2 Definitions

**“AIS”** – Account Information Service, means an online service to provide consolidated information on one or more payment accounts held by the payment service user with either another payment service provider or with more than one payment service provider

**“PIS”** – Payment Initiation Service, a service to initiate a payment order at the request of the payment service user with respect to a payment account held at another payment service provider;

**“PIIS”** – payment service providers issuing card-based payment instruments

**“PSU”** – Payment Service User, means a natural or legal person making use of a payment service in the capacity of payer, payee, or both.

**“TPPSP”** – Third Party Payment Service Provider, means either AIS, PISP or PIIS.

### 3 Registration

To obtain the Open Banking API access credentials to Open Banking API the TPPSP organization has to undertake a registration process during which TrustPay will check if the organization does meet the requirements for this service to be enabled.

The basic requirements to get an access to Open Banking API:

- The organization needs to have an authorization to provide at least one of the following payment services as referred to in Annex 1 of PSD2 Directive, depending on the character of an requested access to Open Banking API:
  - o Issuing of payment instruments and/or acquiring of payment transactions
  - o Payment initiation services
  - o Account information services
- The organization must meet the terms and conditions
- Provide the TPPSP's eIDAS based certificate (QWAC/QSEAL) conforming to ETSI standard (ETSI TS 119 495).

To apply for the Open Banking API access credentials please contact our support department via email ([support@trustpay.eu](mailto:support@trustpay.eu)). Our colleagues will ask you to submit the necessary data about your organization and your license.

Once your request to access the Open Banking API has been approved, TrustPay technical support will contact you regarding the integration and provide you the following via a secure channel:

- TPPSP ID
- Secret Key

TrustPay also enables for TPPSPs to test their implementation of TrustPay's Open Banking API in a test environment. To obtain access to test environment TrustPay issues upon request test credentials for TPPSP. Test API endpoints are stated in each section of this document.

## 4 Security

TLS v1.2 with AEAD cipher suite has to be used to access the Open Banking API. Two-way TLS with both client and server certificates is required.

The certificates have to be eIDAS based certificates (QWAC/QSEAL) conforming to ETSI standard (ETSI TS 119 495).

To establish the TLS connection, the process is as follows:

1. Client tries to access the API HTTP endpoint
2. Server presents the server certificate
3. Client verifies the server certificate
4. Client sends the client certificate to the server
5. Server verifies the client certificate and credentials

All the data in the certificate will be validated during registration, including the roles of the PSP specified in the certificate. The PSP will be able to access only the APIs that correspond to the roles specified in the certificate.

### 4.1 Authorization

In addition to presenting the client certificate and establishing secure TLS connection, the client needs to authorize using OAuth 2.0 (RFC 6749).

The available OAuth calls are detailed in this chapter.

#### 4.1.1 Authorization Code Grant (Strong customer authentication – SCA)

Based on Section 4.1 - Authorization Code Grant of OAuth Framework (RFC 6749).

Can be used to obtain authorization code. This code can then be used to obtain long term access token.

This section documents how to create a URL, to which PSU needs to be redirected. On this URL the PSU will authenticate and then will be redirected back to the PSP to the defined redirect URI.

Test endpoint: <https://api.test.trustpay.eu/psd2/oauth2/authorize>

Production endpoint: <https://api.trustpay.eu/psd2/oauth2/authorize>

**Request:**

Parameter	Description	Format	Required
<b>response_type</b>	Value must be set to "code"	Varchar(32)	Yes
<b>client_id</b>	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
<b>redirect_uri</b>	The URL to which the PSU will be redirected after the authentication process. Query part of the URL can be present and will be preserved. Fragment part of the URL can not be present.	Varchar(2048)	Yes
<b>scope</b>	The scope of the access request (AIS, PIS, PIIS). If PIS is used, the authorization code will result in non-renewable access token.	Varchar(64)	No
<b>state</b>	An opaque value used by the client to maintain between the request and callback. This value is included when redirecting to <code>redirect_uri</code> . This parameter should be used for preventing cross-site request forgery.	Varchar(256)	No
<b>trustpay_oid</b>	Order ID. Used to bind the authorization code with prepared payment order. Has to be used with scope PIS.	Varchar(32)	No

**Example:**

GET

```
https://api.test.trustpay.eu/psd2/oauth2/authorize?response_type=code&client_id=123
&state=xyz&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom HTTP/1.1
```

Host: api.test.trustpay.eu

**Response:**

Parameter	Description	Format
<b>code</b>	The authorization code generated by TrustPay authorization server and it has to be exchanged for Access Token. This code expires in 10 minutes.	Varchar(256)
<b>state</b>	The exact value received from the TPPSP in the request.	Varchar(256)



**Example:**

HTTP/1.1 302 Found

Location: <https://client.example.com/?code=abc&state=xyz>

#### 4.1.2 Access Token Request (Long term renewable token)

Based on Section 4.1.3 - Access Token Request of OAuth Framework (RFC 6749).

Can be used to exchange authorization code for access token that is valid for 30 minutes, but can be refreshed for 90 days.

Test endpoint: <https://api.test.trustpay.eu/psd2/oauth2/token>Production endpoint: <https://api.trustpay.eu/psd2/oauth2/token>**Request:**

Parameter	Description	Format	Required
<b>grant_type</b>	Value must be set to "authorization_code".	Varchar(32)	Yes
<b>code</b>	The Authorization code obtained previously	Varchar(256)	Yes
<b>redirect_uri</b>	Must be same as in the request for authorization code.	Varchar(2048)	Yes
<b>client_id<sup>1</sup></b>	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
<b>client_secret<sup>1</sup></b>	The Secret Key obtained during the TPPSP registration process	Varchar(256)	Yes

<sup>1</sup> if preferred, these parameters can be replaced with basic HTTP authentication scheme (Authorization header)**Example:**POST <https://api.test.trustpay.eu/psd2/oauth2/token> HTTP/1.1Host: [api.test.trustpay.eu](https://api.test.trustpay.eu)

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

```
grant_type=authorization_code&code=abc
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom&client_id=123
&client_secret=secret
```

**Response:**

Parameter	Description	Format
<b>access_token</b>	The Access Token to be used for further API calls	Varchar(256)
<b>expires_in</b>	Duration of received token (the default validity of Access Token is 30 minutes)	Int
<b>token_type</b>	Type of token, value is always "Bearer"	Varchar(32)
<b>refresh_token</b>	The Token to be used to generate new Access Token during the 90 days period	Varchar(256)

**Example:**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

```
{
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "token_type": "Bearer",
  "expires_in": 1800,
  "refresh_token": "tGzv3J0kF0XG5Qx2TlKWIA"
}
```

### 4.1.3 Refreshing an Access Token

Based on Section 6 - Refreshing an Access Token of OAuth Framework (RFC 6749).

Can be used to refresh previously generated access token.

Test endpoint: <https://api.test.trustpay.eu/psd2/oauth2/token>

Production endpoint: <https://api.trustpay.eu/psd2/oauth2/token>

**Request:**

Parameter	Description	Format	Required
<b>grant_type</b>	Value must be set to "refresh_token"	Varchar(32)	Yes
<b>refresh_token</b>	The Refresh Token obtained from the previous Access Token request	Varchar(256)	Yes
<b>client_id<sup>1</sup></b>	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
<b>client_secret<sup>1</sup></b>	The Secret Key obtained during the TPPSP registration process	Varchar(256)	Yes

<sup>1</sup> if preferred, these parameters can be replaced with basic HTTP authentication scheme (Authorization header)

**Example:**

POST https://api.test.trustpay.eu/psd2/oauth2/token HTTP/1.1

Host: api.test.trustpay.eu

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

grant\_type=refresh\_token&refresh\_token=tGzv3J0kF0XG5Qx2T1KWIA  
&client\_id=123&client\_secret=secret

**Response:**

Parameter	Description	Format
<b>access_token</b>	The Access Token to be used for further API calls	Varchar(256)
<b>expires_in</b>	Duration of received token (the default validity of Access Token is 30 minutes)	Int
<b>token_type</b>	Type of token, value is always "Bearer"	Varchar(32)
<b>refresh_token</b>	A new Refresh Token to be used to generate new Access Token during the 90 days period	Varchar(256)

**Example:**

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

Cache-Control: no-store

Pragma: no-cache

```
{
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
```

```

    "token_type": "Bearer",
    "expires_in": 1800,
    "refresh_token": "tGzv3J0kF0XG5Qx2T1KWIA"
  }

```

#### 4.1.4 Client Credentials Grant (Short term non-renewable token)

Based on Section 4.4 - Client Credentials Grant of OAuth Framework (RFC 6749).

This token is needed to initiate a standard payment. This access token is valid 30 minutes and can't be renewed.

Test endpoint: <https://api.test.trustpay.eu/psd2/oauth2/token>

Production endpoint: <https://api.trustpay.eu/psd2/oauth2/token>

##### Request:

Parameter	Description	Format	Required
<b>grant_type</b>	Value must be set to "client_credentials"	Varchar(32)	Yes
<b>scope</b>	The scope of the access request. Only PIS is currently allowed.	Varchar(64)	No
<b>client_id<sup>1</sup></b>	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
<b>client_secret<sup>1</sup></b>	The Secret Key obtained during the TPPSP registration process	Varchar(256)	Yes

<sup>1</sup> if preferred, these parameters can be replaced with basic HTTP authentication scheme (Authorization header)

##### Example:

POST <https://api.test.trustpay.eu/psd2/oauth2/token> HTTP/1.1

Host: [api.test.trustpay.eu](https://api.test.trustpay.eu)

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

grant\_type=client\_credentials&client\_id=123&client\_secret=secret

**Response:**

Parameter	Description	Format
<b>access_token</b>	The Access Token to be used for further API calls	Varchar(256)
<b>expires_in</b>	Duration of received token (the default validity of Access Token is 30 minutes)	Int
<b>token_type</b>	Type of token, value is always "Bearer"	Varchar(32)

**Example:**

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

```
{
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "token_type": "Bearer",
  "expires_in": 1800
}
```

## 5 AIS and PIIS

The Account Information Services (AIS) and Payment Instrument Issuer Services (PIIS) are passive operations. This means that these operations are not able to change the balance on the PSU's TrustPay account.

TrustPay supports the following API requests for AIS and PIIS:

- Get Account List (AIS)
- Get Account Details (AIS)
- Get Transaction History (AIS)
- Get Transaction Details (AIS)
- Check Balance (PIIS)

### 5.1 Authorization for AIS and PIIS operations

For the TPPSP to be able to send API requests to TrustPay servers a PSU consent and a Access Token needs to be granted to the TPP. The TPP uses the granted access token linked to a specific PSU to process API requests.

The received Access Token via this method can be used for 90 days to access the linked accounts. After this period the PSU has to provide a new consent and a new Access Token is granted the access of TPPSP again.

The process of obtaining the Access Token:

**Step1:** The TPPSP redirects the PSU to the URL constructed according to section [Authorization Code Grant \(Strong customer authentication – SCA\)](#). There the PSU is requested to log in to TrustPay Internet Banking using Strong Customer Authentication. Once the PSU has logged in, a page with the PSU consent is displayed where a statement is presented informing the PSU about the specific permissions he is granting to TPPSP. If the PSU owns multiple TrustPay accounts, he can choose via a multi-choice option which accounts these permissions should be granted for. The PSU is requested to Submit the PSU consent.

**Step2:** After the successful submission of PSU consent TrustPay redirects the PSU to TPPSP's Redirect URL .

**Step3:** TPPSP needs to exchange the authorization code provided in the response for an Access Token.

### 5.2 Refreshing Access Token

The default validity of the Access token is 30 minutes. Once the token has expired but the PSU consent is not older than 90 days, TPPSP can request a new Access Token without the need of the PSU to be present and to authorize this action. For this action the TPPSP has to use the Refresh Token received in the previous Access Token request. See [Refreshing an Access Token](#) for details.

### 5.3 Get Account List

This request returns a response with a list of all TrustPay accounts together with some more details about the accounts. The response contains all the accounts to which the PSU has granted an access to during the Authorization process and Access token creation.

#### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/GetAccountList>

Production endpoint: <https://api.trustpay.eu/psd2/GetAccountList>

Header	Description	Format	Required
<b>Authorization</b>	The Access Token received in the previous steps	Varchar(265)	Yes

Request example:

```
using System;
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public dynamic GetAccountList(string token)
{
    string url = "https://api.trustpay.eu/psd2/GetAccountList";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
    };

    JavaScriptSerializer serializer = new JavaScriptSerializer();

    var responseData = client.UploadData(url, new byte[0]);
    var response =
    serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

    return response;
}
```

**Response:**

<b>Name</b>	<b>Description</b>	<b>Format</b>
<b>AccountId</b>	ID of account assigned by TrustPay	Numeric(10)
<b>IBAN</b>	IBAN (empty if account is not ibanized)	Varchar(34)
<b>AccountName</b>	Account name	Varchar(35)
<b>AccountOwnerName</b>	Account owner name	Varchar(70)
<b>AccountType</b>	Account type	"Individual", "Merchant"
<b>CanCreateInternalOrder</b>	Indicates whether it is possible for current disponent to create internal orders on this account	Boolean
<b>CanCreateBankWireOrder</b>	Indicates whether it is possible for current disponent to create bank wire orders on this account	Boolean
<b>CurrencyCode</b>	Currency of the account	Char(3)
<b>AccountingBalance</b>	Accounting balance	Numeric(13,2) en-US format
<b>DisposableBalance</b>	Disposable balance (empty if unlimited)	Numeric(13,2) en-US format
<b>FeeBalance</b>	Fee balance	Numeric(13,2) en-US format
<b>MinimalBalance</b>	Minimal balance	Numeric(13,2) en-US format



Response example:

```
{
  "AccountList": [
    {
      "AccountId": 2107347927,
      "IBAN": "SK109952000002107347927",
      "AccountName": "John Doe EUR",
      "AccountOwnerName": "John Doe",
      "AccountType": "Merchant",
      "CanCreateInternalOrder": false,
      "CanCreateBankWireOrder": true,
      "CurrencyCode": "EUR",
      "AccountingBalance": "15684.05",
      "DisposableBalance": "15654.05",
      "FeeBalance": "0.00",
      "MinimalBalance": "30.00"
    },
    {
      "AccountId": 2107138076,
      "IBAN": "",
      "AccountName": "John Doe USD",
      "AccountOwnerName": "John Doe",
      "AccountType": "Individual",
      "CanCreateInternalOrder": false,
      "CanCreateBankWireOrder": true,
      "CurrencyCode": "USD",
      "AccountingBalance": "121696.89",
      "DisposableBalance": "121410.89",
      "FeeBalance": "256.00",
      "MinimalBalance": "30.00"
    }
  ]
}
```

## 5.4 Get Account Details

This request returns a response with the details of a specific TrustPay account. To receive a successful response to this request the PSU must have provided a consent for the specific account during the Authorization process.

### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/GetAccountDetails>

Production endpoint: <https://api.trustpay.eu/psd2/GetAccountDetails>

Header	Description	Format	Required
<b>Authorization</b>	The Access Token received in the previous steps	Varchar(265)	Yes

Parameter	Description	Format	Required
<b>AccountId</b>	Desired account ID (ID of account assigned by TrustPay)	Numeric(10)	Yes

Request example:

```
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public dynamic GetAccountDetails(string token)
{
    string url = "https://api.trustpay.eu/psd2/GetAccountDetails";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
        var request = new
        {
            AccountId = "2107000000"
        };

        JavaScriptSerializer serializer = new JavaScriptSerializer();

        var requestData = Encoding.UTF8.GetBytes(serializer.Serialize(request));
        var responseData = client.UploadData(url, requestData);
        var response =
        serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

        return response;
    }
}
```

**Response:**

<b>Name</b>	<b>Description</b>	<b>Format</b>
<b>AccountId</b>	ID of account assigned by TrustPay	Numeric(10)
<b>IBAN</b>	IBAN (empty if account is not ibanized)	Varchar(34)
<b>AccountName</b>	Account name	Varchar(35)
<b>AccountOwnerName</b>	Account owner name	Varchar(70)
<b>AccountType</b>	Account type	"Individual", "Merchant"
<b>CanCreateInternalOrder</b>	Indicates whether it is possible for current disponent to create internal orders on this account	Boolean
<b>CanCreateBankWireOrder</b>	Indicates whether it is possible for current disponent to create bank wire orders on this account	Boolean
<b>CurrencyCode</b>	Currency of the account	Char(3)
<b>AccountingBalance</b>	Accounting balance	Numeric(13,2) en-US format
<b>DisposableBalance</b>	Disposable balance (empty if unlimited)	Numeric(13,2) en-US format
<b>FeeBalance</b>	Fee balance	Numeric(13,2) en-US format
<b>MinimalBalance</b>	Minimal balance	Numeric(13,2) en-US format

Response example:

```
{
  "AccountDetails": {
    "AccountId": "2107347927",
    "IBAN": "SK109952000002107347927",
    "AccountName": "John Doe EUR",
    "AccountOwnerName": "John Doe",
    "AccountType": "Merchant",
    "CanCreateInternalOrder": false,
    "CanCreateBankWireOrder": true,
    "CurrencyCode": "EUR",
    "AccountingBalance": "15684.05",
    "DisposableBalance": "15654.05",
    "FeeBalance": "0.00",
    "MinimalBalance": "30.00"
  }
}
```

## 5.5 Get Transaction History

This request returns all the transaction history with transaction details for a selected account within given date range. The request also provides an option to filter data and set amount of returned items.

### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/GetTransactionHistory>

Production endpoint: <https://api.trustpay.eu/psd2/GetTransactionHistory>

Header	Description	Format	Required
<b>Authorization</b>	The Access Token received in the previous steps	Varchar(265)	Yes

Parameter	Description	Format	Required
<b>Filter</b>	Transaction history filter (see explained below)	-	Yes
<b>Paging</b>	Paging information (see explained below)	-	Yes

Request/Filter:

Name	Description	Format	Required
<b>AccountId</b>	Account ID (ID of account assigned by TrustPay)	Numeric(10)	Yes
<b>DateFrom</b>	Date from	Varchar(10)	Yes
<b>DateTo</b>	Date to	Varchar(10)	Yes
<b>Type</b>	Statement type	"Any", "Internal", "External"	No
<b>Direction</b>	Transaction direction	"Any", "Credit", "Debit"	No
<b>CounterAccount</b>	Counter account	Varchar(34)	No
<b>AmountFrom</b>	Amount from	Numeric(13,2) en-US format	No
<b>AmountTo</b>	Amount to	Numeric(13,2) en-US format	No
<b>ClientReference</b>	Client reference	Varchar(256)	No
<b>TransactionId</b>	TrustPay Transaction ID	Numeric(10)	No
<b>PaymentId</b>	TrustPay Payment ID	Numeric(10)	No
<b>OrderId</b>	TrustPay Order ID	Numeric(10)	No

Request/Paging:

Name	Description	Format	Required
<b>Page</b>	Number of page to show (starts from zero)	Numeric(10)	Yes
<b>PageSize</b>	Number of results on the page	Numeric(5)	Yes

## Request example:

```
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public dynamic GetTransactionHistory(string token)
{
    string url = "https://api.trustpay.eu/psd2/GetTransactionHistory";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
        var request = new
        {
            Filter = new
            {
                AccountId = 2107000000,
                DateFrom = "2017-08-01",
                DateTo = "2017-09-01",
            },
            Paging = new
            {
                Page = 0,
                PageSize = 10
            }
        };

        JavaScriptSerializer serializer = new JavaScriptSerializer();

        var requestData = Encoding.UTF8.GetBytes(serializer.Serialize(request));
        var responseData = client.UploadData(url, requestData);
        var response =
        serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

        return response;
    }
}
```

## Response:

Name	Description	Format
<b>Transactions</b>	Details of selected transactions	see below "Response/TransactionDetails"
<b>TotalCount</b>	Count of all transactions on the account	Numeric(10)
<b>Paging</b>	Paging info	See above "Request/Paging"

## Response/TransactionDetails:

Name	Description	Format
<b>TransactionId</b>	Transaction ID	TrustPay Transaction ID (unique ID used for any enquiries)
<b>PaymentId</b>	Payment ID	TrustPay Payment ID
<b>OrderId</b>	Order ID	TrustPay Order ID
<b>Date</b>	Transaction date	Varchar(10)
<b>CounterAccount</b>	Counter account	Varchar(34)
<b>CounterAccountName</b>	Counter account name	Varchar(70)
<b>Description</b>	Transaction description	Varchar(140)
<b>PayerReference</b>	Payer's reference	Varchar(140)
<b>ClientReference</b>	Client's reference	Varchar(500)
<b>Amount</b>	Transaction amount	Numeric(13,2) en-US format
<b>Currency</b>	Transaction currency	Char(3)
<b>RefundedAmount</b>	Refunded amount	Numeric(13,2) en-US format

Response example:

```
{
  "Transactions": [
    {
      "TransactionId": 652804,
      "PaymentId": 325843,
      "OrderId": 125897,
      "Date": "2017-08-17",
      "CounterAccount": "SK369952000002107842741",
      "CounterAccountName": "Account name",
      "Description": "description",
      "PayerReference": "e2e",
      "ClientReference": "1234567890",
      "Amount": 123.45,
      "Currency": "EUR",
      "RefundedAmount": 0.00
    },
    {
      "TransactionId": 652773,
      "PaymentId": 325831,
      "OrderId": 110876,
      "Date": "2017-08-14",
      "CounterAccount": "0002725006",
      "CounterAccountName": "Account name",
      "Description": "interesting description",
      "PayerReference": "TP002517042",
      "ClientReference": null,
      "Amount": -13.37,
      "Currency": "EUR",
      "RefundedAmount": 0.00
    }
  ],
  "TotalCount": 306,
  "Paging": {
    "Page": 50,
    "PageSize": 2
  }
}
```

## 5.6 Get Transaction Details

This request returns a response with the details of a specific transaction. To receive a successful response the provided TransactionID must belong to an account to which the PSU has granted a mandate to during the Authorization process.

### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/GetTransactionDetails>

Production endpoint: <https://api.trustpay.eu/psd2/GetTransactionDetails>



Header	Description	Format	Required
<b>Authorization</b>	The Access Token received in the previous steps	Varchar(265)	Yes

Parameter	Description	Format	Required
<b>TransactionId</b>	TrustPay Transaction ID (unique ID used for any enquiries)	Numeric(10)	No <sup>1</sup>
<b>PaymentId</b>	TrustPay Payment ID	Numeric(10)	No <sup>1</sup>
<b>OrderId</b>	TrustPay Order ID	Numeric(10)	No <sup>1</sup>

1 – At least one of the fields must be provided. If multiple fields are provided status 400 (Bad Request) will be returned.

Request example:

```
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public dynamic GetTransactionDetails(string token)
{
    string url = "https://api.trustpay.eu/psd2/GetTransactionDetails";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
        var request = new
        {
            TransactionId = "123456789"
        };

        JavaScriptSerializer serializer = new JavaScriptSerializer();

        var requestData = Encoding.UTF8.GetBytes(serializer.Serialize(request));
        var responseData = client.UploadData(url, requestData);
        var response =
        serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

        return response;
    }
}
```

**Response:**

Name	Description	Format
<b>TransactionDetails</b>	Details of requested transaction	see below "Response/TransactionDetails"
<b>Xml</b>	Xml order (urn:iso:std:iso:20022:tech:xsd:pain.001.001.05)	Varchar(MAX)

## Response/TransactionDetails:

Name	Description	Format
<b>TransactionId</b>	Transaction ID	TrustPay Transaction ID (unique ID used for any enquiries)
<b>PaymentId</b>	Payment ID	TrustPay Payment ID
<b>OrderId</b>	Order ID	TrustPay Order ID
<b>Date</b>	Transaction date	Varchar(10)
<b>CounterAccount</b>	Counter account	Varchar(34)
<b>CounterAccountName</b>	Counter account name	Varchar(70)
<b>Description</b>	Transaction description	Varchar(140)
<b>PayerReference</b>	Payer's reference	Varchar(140)
<b>ClientReference</b>	Client's reference	Varchar(500)
<b>Amount</b>	Transaction amount	Numeric(13,2) en-US format
<b>Currency</b>	Transaction currency	Char(3)
<b>RefundedAmount</b>	Refunded amount	Numeric(13,2) en-US format

Response example:

```
{
  "TransactionDetails": {
    "TransactionId": 654731,
    "PaymentId": 326219,
    "OrderId": 125897,
    "Date": "2017-08-28",
    "CounterAccount": "SK469952000002107553501",
    "CounterAccountName": "Account name",
    "Description": "description",
    "PayerReference": "NOTPROVIDED",
    "ClientReference": null,
    "Amount": 4328717.29,
    "Currency": "EUR",
    "RefundedAmount": 0.00
  },
  "Xml": "<Document xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
    xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"
    xmlns=\"urn:iso:std:iso:20022:tech:xsd:pain.001.001.05\">...</Document>"
}
```

## 5.7 Check Balance

This request serves for checking whether the disposable balance of the specific TrustPay account is sufficient for a transaction amount and currency stated in the API request. The response returns a TRUE or FALSE value, while TRUE means that there is a sufficient balance.

The currency has to equal the currency of the specific TrustPay account.

### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/CheckBalance>

Production endpoint: <https://api.trustpay.eu/psd2/CheckBalance>

Header	Description	Format	Required
<b>Authorization</b>	The Access Token received in the previous steps	Varchar(265)	Yes

Name	Description	Format	Required
<b>AccountId</b>	Desired account ID (ID of account assigned by TrustPay)	Numeric(10)	Yes
<b>Amount</b>	Transaction amount	Numeric(13,2) en-US format	Yes
<b>Currency</b>	Transaction currency	Char(3)	Yes

Request example:

```
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public dynamic CheckBalance(string token)
{
    string url = "https://api.trustpay.eu/psd2/CheckBalance";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
        var request = new
        {
            AccountId = "123456789",
            Amount = 57.40
            Currency = "EUR"
        };

        JavaScriptSerializer serializer = new JavaScriptSerializer();

        var requestData = Encoding.UTF8.GetBytes(serializer.Serialize(request));
        var responseData = client.UploadData(url, requestData);
        var response = serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

        return response;
    }
}
```

Response:

Name	Description	Format
<b>CheckBalance</b>	The parameter confirm whether there is a sufficient disposable balance on the account or not	Boolean

Response example:

```
{  
  "CheckBalance": true  
}
```

## 6 PIS

This section of the API describes the Payment Initiation Service (PIS) based on which the TPPSP can submit payment orders from the PSU 's TrustPay account.

The whole payment initiation process consist of two main steps, which are Payment Initiation and a separate Payment Submission.

### 6.1 Standard Payment Initiation

TrustPay supports submitting payment orders via standard pain.001.001.03 ISO messages. This process consists of 3 steps:

#### 6.1.1 Step1: Obtain Access Token

As the first step TPPSP should obtain an Access Token for the communication with TrustPay. This is a short-term access token which is needed to verify the TPPSP in further communication. This access token is valid 30 minutes and can't be renewed. After expiration, a new Access Token has to be requested. See [Client Credentials Grant \(Short term non-renewable token\)](#) for details.

#### 6.1.2 Step2: Initiate payment

With the obtained Access Token the TPPSP can initiate a payment from the PSU 's TrustPay account.

To initiate a payment the TPPSP has to populate the Debtor IBAN field, the IBAN of the TrustPay account that should be debited. In case the TPPSP does not have this IBAN number, they can obtain it via AIS/GetAccountList services (described above) or they can ask the PSU to enter it on the TPPSP 's environment.

To initiate a payment the TPPSP should send a request with pain.001.001.03 ISO 20022 standard message.

#### **Request:**

Test endpoint: <https://api.test.trustpay.eu/psd2/CreateOrder>

Production endpoint: <https://api.trustpay.eu/psd2/CreateOrder>

Header	Description	Format	Required
<b>Authorization</b>	The Access Token received in the previous steps	Varchar(265)	Yes

Parameter	Description	Format	Required
<b>Xml</b>	Xml order (urn:iso:std:iso:20022:tech:xsd:pain.001.001.03)	Varchar(MAX)	Yes

Following payment types are available:

- Slovak domestic payment – [pain.001.001.03 example](#)
- Czech domestic payment – [pain.001.001.03 example](#)
- SEPA payment – [pain.001.001.03 example](#)
- Foreign Payment – [pain.001.001.03 example](#)

The description of XML fields:

Name	Description	Format	Required
<b>AbaCode</b>	ABA routing number	Numeric(9)	For US orders
<b>Address</b>	Address	Varchar(60)	No
<b>Amount</b>	Ordered amount (exactly 2 decimal places)	Numeric(13,2) en-US format	Yes
<b>BsbCode</b>	Bank State Branch code	Numeric(6)	For Australian orders
<b>CityCode</b>	City code	Varchar(35)	Yes
<b>ClientReference</b>	Client reference enclosed with '<Strd><AddtlRmtInf>CREF:' and '</AddtlRmtInf></Strd>'. If not filled, also omit the XML tags.	Varchar(30) excluding XML tags	No
<b>CorrespondentBank</b>	Correspondent bank	Varchar(35)	No
<b>CreationDateTime</b>	Order creation date time	UTC, ISO 8601 format, sortable format - 'yyyy-MM-ddTHH:mm:ss'	Yes

<b>CreditorAccount</b>	Creditor account IBAN for <IBAN></IBAN> tag and other for <Othr><Id></Id></Othr>	Varchar(34) excluding XML tags	Yes
<b>CreditorAccountWithTags</b>	Creditor account including either <IBAN></IBAN> or <Othr><Id></Id></Othr> tags	Varchar(34) excluding XML tags	Yes
<b>CreditorAddress</b>	Creditor address	Other country: Varchar(35) / SEPA: Varchar(70)	Other country: Yes / SEPA: No
<b>CreditorBankAddress</b>	Creditor bank street	Varchar(35)	Yes
<b>CreditorBankBic</b>	Creditor bank	Varchar(11)	Yes
<b>CreditorBankCity</b>	Creditor bank city	Varchar(35)	Yes
<b>CreditorBankCountry</b>	Creditor bank country	Char(2)	Yes
<b>CreditorBankName</b>	Creditor bank name	Varchar(35)	Yes
<b>CreditorCity</b>	Creditor city	Varchar(35)	Foreign payment: Yes / SEPA: No
<b>CreditorCountry</b>	Creditor country	Char(2)	Foreign payment: Yes / SEPA: No
<b>CreditorName</b>	Creditor name	Varchar(35)	Yes
<b>Currency</b>	Currency code	Char(3)	Yes
<b>DebtorAccount</b>	Debtor account	Numeric(10)	Yes
<b>DebtorName</b>	Debtor name	Varchar(35)	Yes
<b>Description</b>	Description	Varchar(140)	No
<b>EndToEndReference</b>	End to end reference	Varchar(35)	Yes <sup>1</sup>
<b>MessageId</b>	Message ID	Varchar(35)	Yes
<b>RequestedExecutionDate</b>	Requested execution date (must be today or greater)	Varchar(10)	Yes
<b>SymbolConstant</b>	Constant symbol	Numeric(4)	No



<b>SymbolSpecific</b>	Specific symbol	Numeric(10)	No
<b>SymbolVariable</b>	Variable symbol	Numeric(10)	No
<b>TransitCode</b>	Transit code	Numeric(9) starting with 0	For Canadian orders

<sup>1</sup> Fill in NOTPROVIDED if you do not need to use end to end reference

Request example:

```
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public long CreateOrder(string token)
{
    string url = "https://api.trustpay.eu/ApiBanking/CreateOrder";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
        var request = new
        {
            Xml = "... "
        };

        JavaScriptSerializer serializer = new JavaScriptSerializer();

        var requestData = Encoding.UTF8.GetBytes(serializer.Serialize(request));
        var responseData = client.UploadData(url, requestData);
        var response =
        serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

        return response["OrderId"];
    }
}
```

Response:

Name	Description	Format
<b>OrderId</b>	ID of the prepared payment. Has to be filled to parameter trustpay_oid when doing SCA in next step.	Varchar(32)

Response example:

```
{  
  "OrderId": "123456"  
}
```

### 6.1.3 Step3: Payment Authorization (redirect to TrustPay)

TPPSP redirects the PSU to the URL obtained by following the guide in section [Authorization Code Grant \(Strong customer authentication – SCA\)](#). It is important to fill the order ID from previous step into the trustpay\_oid parameter. Also the scope parameter has to contain the value PIS.

The PSU is then requested to enter their TrustPay login credentials and to authorize the payment using SCA. The authorization is only possible if the specific PSU has active permission to the specific Debtor Account requested in the previous request. If the specific PSU is not permitted to authorize the payment an error message will be displayed and the PSU will be redirected to TPPSP.

After successful authorization of payment order, the PSU is redirected to URL specified in SCA redirection URL along with an authorization code (bound with the initiated payment).

### 6.1.4 Step4: Getting token that will be used to submit the payment

Authorization code from step 3 needs to be converted to a token that can be used to submit the payment. This is done by following the documentation in section [Access Token Request \(Long term renewable token\)](#).

Since the scope PIS had to be specified and trustpay\_oid parameter had to be used, the resulting token will not be refreshable and is bound to the prepared order.

## 6.2 Standard Payment Submission

The TPPSP has 30 minutes to send the Payment Submission request after the authorization has been successfully authorized.

### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/PaymentSubmission>

Production endpoint: <https://api.trustpay.eu/psd2/PaymentSubmission>

Header	Description	Format	Required
--------	-------------	--------	----------

<b>Authorization</b>	The Access Token received in the previous step (section 6.1.4.)	Varchar(265)	Yes
----------------------	---	--------------	-----

### 6.3 Payment Order Status

This API request enables the TPPSP to obtain the current status of a submitted order. TPPSP can check only the status of an order that was submitted with the same TPPSP ID.

#### Request:

Test endpoint: <https://api.test.trustpay.eu/psd2/OrderStatus>

Production endpoint: <https://api.trustpay.eu/psd2/OrderStatus>

Header	Description	Format	Required
<b>Authorization</b>	The short term non-renewable token. See <a href="#">Client Credentials Grant (Short term non-renewable token)</a>	Varchar(265)	Yes

Parameter	Description	Format	Required
<b>OrderId</b>	Id of the created payment order	Varchar(32)	Yes

## Request example:

```
using System.Net;
using System.Text;
using System.Web.Script.Serialization;

public dynamic GetTransactionDetails(string token)
{
    string url = "https://api.trustpay.eu/psd2/OrderStatus";
    using (var client = new WebClient())
    {
        client.Headers = new WebHeaderCollection
        {
            "Authorization: bearer " + token,
            "Content-Type: text/json"
        };
        var request = new
        {
            OrderId = "123456789"
        };

        JavaScriptSerializer serializer = new JavaScriptSerializer();

        var requestData = Encoding.UTF8.GetBytes(serializer.Serialize(request));
        var responseData = client.UploadData(url, requestData);
        var response =
        serializer.Deserialize<dynamic>(Encoding.UTF8.GetString(responseData));

        return response;
    }
}
```

## Response:

Name	Description	Format
<b>Status</b>	The current status of the submitted payment order. Possible values: <ul style="list-style-type: none"><li>- ACTC (AcceptedTechnicalValidation)</li><li>- ACWC (AcceptedWithChange)</li><li>- RJCT (Rejected)</li><li>- PDNG (Pending)</li><li>- ACSP (AcceptedSettlementInProgress)</li><li>- ACSC (AcceptedSettlementCompleted)</li></ul>	

## Response example:

```
{
  "Status": "ACSP"
}
```