



Open Banking API manual

Version 1.2
December 2024



Versions

Version	Date	Changes
1.0	November, 2020	First edition
1.1	March, 2024	Added account selection screen to Payment Initiation Services
1.2	December, 2024	- The expiry of PSU consent (validity period of Refresh Token) changed from 90 to 180 days - "Check Balance" API request added for PIS providers



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 finby).....	34
6.2 Standard Payment Submission	34
6.3 Payment Order Status.....	35
6.4 Check Balance	37



1 Introduction

Based on EU Directive on Payment Services (PSD 2) effective from 13. 1. 2018 Trust Pay a.s., with its registered office at Za Kasárňou 1, 831 03, Bratislava, Company ID: 36 865 800 (hereinafter referred to as "**finby**") is publishing this Open Banking API to enable Third Party Payment Service Providers (TPPSP) to access payment account information and initiate payment orders from payment accounts of finby 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 finby payment accounts, the organization has to undertake a registration process to obtain the necessary credentials to be able to connect and communicate with finby 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 finby 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@finby.com). 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, finby technical support will contact you regarding the integration and provide you the following via a secure channel:

- TPPSP ID
- Secret Key



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
response_type	Value must be set to "code"	Varchar(32)	Yes
client_id	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
redirect_uri	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
scope	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
state	An opaque value used by the client to maintain between the request and callback. This value is included when redirecting to redirect_uri. This parameter should be used for preventing cross-site request forgery.	Varchar(256)	No
trustpay_oid	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
code	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)
state	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 180 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
grant_type	Value must be set to "authorization_code".	Varchar(32)	Yes
code	The Authorization code obtained previously	Varchar(256)	Yes
redirect_uri	Must be same as in the request for authorization code.	Varchar(2048)	Yes
client_id ¹	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
client_secret ¹	The Secret Key obtained during the TPPSP registration process	Varchar(256)	Yes

¹ 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=authorization_code&code=abc
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom&client_id=123
&client_secret=secret



Response:

Parameter	Description	Format
access_token	The Access Token to be used for further API calls	Varchar(256)
expires_in	Duration of received token (the default validity of Access Token is 30 minutes)	Int
token_type	Type of token, value is always "Bearer"	Varchar(32)
refresh_token	The Token to be used to generate new Access Token during the 180 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
grant_type	Value must be set to "refresh_token"	Varchar(32)	Yes
refresh_token	The Refresh Token obtained from the previous Access Token request	Varchar(256)	Yes
client_id¹	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
client_secret¹	The Secret Key obtained during the TPPSP registration process	Varchar(256)	Yes

¹ 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=tGzv3J0kF0XG5Qx2TlKWIA
&client_id=123&client_secret=secret

Response:

Parameter	Description	Format
access_token	The Access Token to be used for further API calls	Varchar(256)
expires_in	Duration of received token (the default validity of Access Token is 30 minutes)	Int
token_type	Type of token, value is always "Bearer"	Varchar(32)
refresh_token	A new Refresh Token to be used to generate new Access Token during the 180 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"  
}
```

4.1.4 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
grant_type	Value must be set to "client_credentials"	Varchar(32)	Yes
scope	The scope of the access request. Only PIS is currently allowed.	Varchar(64)	No
client_id ¹	The TPPSP ID obtained during the TPPSP registration process	Varchar(256)	Yes
client_secret ¹	The Secret Key obtained during the TPPSP registration process	Varchar(256)	Yes

¹ 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=client_credentials&client_id=123&client_secret=secret



Response:

Parameter	Description	Format
access_token	The Access Token to be used for further API calls	Varchar(256)
expires_in	Duration of received token (the default validity of Access Token is 30 minutes)	Int
token_type	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 finby account.

finby 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 finby 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 180 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 finby 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 finby 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 finby 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 180 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 finby 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
Authorization	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:

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



Response example:

```
{
  "AccountList": [
    {
      "AccountId": 2107347927,
      "IBAN": "SK1099520000002107347927",
      "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 finby 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
Authorization	The Access Token received in the previous steps	Varchar(265)	Yes



Parameter	Description	Format	Required
AccountId	Desired account ID (ID of account assigned by finby)	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:

Name	Description	Format
AccountId	ID of account assigned by finby	Numeric(10)
IBAN	IBAN (empty if account is not ibanized)	Varchar(34)
AccountName	Account name	Varchar(35)
AccountOwnerName	Account owner name	Varchar(70)
AccountType	Account type	"Individual", "Merchant"
CanCreateInternalOrder	Indicates whether it is possible for current disponent to create internal orders on this account	Boolean
CanCreateBankWireOrder	Indicates whether it is possible for current disponent to create bank wire orders on this account	Boolean
CurrencyCode	Currency of the account	Char(3)
AccountingBalance	Accounting balance	Numeric(13,2) en-US format
DisposableBalance	Disposable balance (empty if unlimited)	Numeric(13,2) en-US format
FeeBalance	Fee balance	Numeric(13,2) en-US format
MinimalBalance	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
Authorization	The Access Token received in the previous steps	Varchar(265)	Yes

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



Request/Filter:

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

Request/Paging:

Name	Description	Format	Required
Page	Number of page to show (starts from zero)	Numeric(10)	Yes
PageSize	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
Transactions	Details of selected transactions	see below "Response/TransactionDetails"
TotalCount	Count of all transactions on the account	Numeric(10)
Paging	Paging info	See above "Request/Paging"



Response/TransactionDetails:

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



Response example:

```
{
  "Transactions": [
    {
      "TransactionId": 652804,
      "PaymentId": 325843,
      "OrderId": 125897,
      "Date": "2017-08-17",
      "CounterAccount": "SK3699520000002107842741",
      "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
Authorization	The Access Token received in the previous steps	Varchar(265)	Yes

Parameter	Description	Format	Required
TransactionId	finbyTransaction ID (unique ID used for any enquiries)	Numeric(10)	No ¹
PaymentId	finbyPayment ID	Numeric(10)	No ¹
OrderId	finbyOrder ID	Numeric(10)	No ¹

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

Response/TransactionDetails:

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



Response example:

```
{
  "TransactionDetails": {
    "TransactionId": 654731,
    "PaymentId": 326219,
    "OrderId": 125897,
    "Date": "2017-08-28",
    "CounterAccount": "SK4699520000002107553501",
    "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 finby 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 finby account.

Request:

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

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

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



Name	Description	Format	Required
AccountId	Desired account ID (ID of account assigned by finby)	Numeric(10)	Yes
Amount	Transaction amount	Numeric(13,2) en-US format	Yes
Currency	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
CheckBalance	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 finby 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

finby 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 finby. 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 finby account.

To initiate a payment the TPPSP should send a request with pain.001.001.03 ISO 20022 standard message. The TPPSP has to populate the Debtor IBAN field, the IBAN of the finby account that should be debited. There are three possibilities:

- The TPPSP does already have this IBAN number
- The TPPSP can obtain it via AIS/GetAccountList services (described above)
- The TPPSP can fill the string NOTPROVIDED into the debtor IBAN field, in which case after the redirect and authorization in step 3, the PSU can chose their account in the account selector provided by finby

Request:

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

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



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

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



		MM-ddTHH:mm:ss'	
CreditorAccount	Creditor account IBAN for <IBAN></IBAN> tag and other for <Othr><Id></Id></Othr>	Varchar(34) excluding XML tags	Yes
CreditorAccountWithTags	Creditor account including either <IBAN></IBAN> or <Othr><Id></Id></Othr> tags	Varchar(34) excluding XML tags	Yes
CreditorAddress	Creditor address	Other country: Varchar(35) / SEPA: Varchar(70)	Other country: Yes / SEPA: No
CreditorBankAddress	Creditor bank street	Varchar(35)	Yes
CreditorBankBic	Creditor bank	Varchar(11)	Yes
CreditorBankCity	Creditor bank city	Varchar(35)	Yes
CreditorBankCountry	Creditor bank country	Char(2)	Yes
CreditorBankName	Creditor bank name	Varchar(35)	Yes
CreditorCity	Creditor city	Varchar(35)	Foreign payment: Yes / SEPA: No
CreditorCountry	Creditor country	Char(2)	Foreign payment: Yes / SEPA: No
CreditorName	Creditor name	Varchar(35)	Yes
Currency	Currency code	Char(3)	Yes
DebtorAccount	Debtor account	Numeric(10)	Yes
DebtorName	Debtor name	Varchar(35)	Yes
Description	Description	Varchar(140)	No
EndToEndReference	End to end reference	Varchar(35)	Yes ¹
MessageId	Message ID	Varchar(35)	Yes



RequestedExecutionDate	Requested execution date (must be today or greater)	Varchar(10)	Yes
SymbolConstant	Constant symbol	Numeric(4)	No
SymbolSpecific	Specific symbol	Numeric(10)	No
SymbolVariable	Variable symbol	Numeric(10)	No
TransitCode	Transit code	Numeric(9) starting with 0	For Canadian orders

¹ 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
------	-------------	--------



OrderId	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 finby)

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 finby_oid parameter. Also the scope parameter has to contain the value PIS.

The PSU is then requested to enter their finby 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) by adding the following parameters to the Redirect URL:

Name	Description	Format
AuthorizationID	The ID assigned to the successful authorization. This ID should be used by TPPSP to submit the payment	

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
Authorization	The Access Token received in the previous steps	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
Authorization	The Access Token received in the previous steps	Varchar(265)	Yes

Parameter	Description	Format	Required
OrderId	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
Status	The current status of the submitted payment order. Possible values: - ACTC (AcceptedTechnicalValidation) - ACWC (AcceptedWithChange) - RJCT (Rejected) - PDNG (Pending) - ACSP (AcceptedSettlementInProgress) - ACSC (AcceptedSettlementCompleted)	

Response example:

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



6.4 Check Balance

This request serves for checking whether the disposable balance of the specific finby 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 finby account.

Request:

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

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

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

Name	Description	Format	Required
AccountId	Desired account ID (ID of account assigned by finby)	Numeric(10)	Yes
Amount	Transaction amount	Numeric(13,2) en-US format	Yes
Currency	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
CheckBalance	The parameter confirm whether there is a sufficient disposable balance on the account or not	Boolean

Response example:

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