



PSD2 API Specification and Documentation

Author	AS "Reģionālā investīciju banka"
Status	Approved
Document version	1.1
Last modified	31.07.2019.

Content

General	3
Link to an API	4
Types of Service Requests Available in API	4
Structure of Description of API Methods	4
Standard Response Format	5
List of Permitted Symbols for the Document	7
Information on the External Payment Code (Necessary for the SEPA Payment Service)	7
1. Log in to the System	8
1.1. Receive a Security Token	8
1.2. Redirect the Customer to the Internet Banking Authentication Page	9
1.3. Establish Account Information Consent	10
1.4. Refresh a Session (Refreshed Token)	10
2. Get Account Details of the List of Accessible Accounts	11
3. Get Balances for a Given Account	13
4. Get Transaction Information for a Given Account	14
5. Get a Confirmation on the Availability of Funds	18
6. Initiation of a Single SEPA Payment	19
7. SEPA Payment Signing	21
History of Changes	22

General

As of June 01, 2019, AS "Reģionālā investīciju banka" (hereinafter –RIB), observing PSD2 requirements, opens access to the RIB Open Banking platform for Payment Initiation and Account Information Service Providers (hereinafter – Third Parties).

The RIB Open Banking platform can be accessed by Third Parties which have received the following:

- ✓ a licence issued by financial supervision institutions;
- ✓ QWAC (Qualified Website Authentication Certificate) which complies with PSD2 Regulatory Technical Standards (RTS).

The Open Banking platform ensures a possibility of sharing data or initiating activities between applications by means of API, which connects the RIB infrastructure to Third-Party solutions.

This document contains a description of RIB API and a manual on connecting to the Open Banking platform. API has been developed in accordance with the "Berlin Group NextGenPSD2 XS2A Framework" standard, adapting it to the RIB Internet Banking architecture.

RIB API uses the HTTPS protocol, GET and POST methods as an access mechanism. JSON is used as a data format. A standard HTTP status code is returned at any request.

Both input and output parameters must comply with the camelCase register style.

Prior to connecting to the RIB Open Banking platform, you are advised to try out the test platform. More information on the use of the test platform is available [here](#).

Should any questions, issues or recommendations arise, please contact RIB by email at openbanking@ribbank.com.

Link to an API

<https://openbanking.ribbank.com:8443/api.cfm>

Types of Service Requests Available in API

#	Use case	Initiator	Initiator abbreviation
1	Initiation of a single SEPA payment	Payment Initiation Service Provider	PISP
2	Get account details of the list of accessible accounts	Account Information Service Provider	AISP
3	Get balances for a given account	Account Information Service Provider	AISP
4	Get transaction information for a given account	Account Information Service Provider	AISP
5	Get a confirmation on the availability of funds	Payment Issuer Instrument Service Provider	PIISP
6	Establish account information consent	Account Information Service Provider	AISP

Structure of Description of API Methods.

The general structure of description of API methods is provided below.

Title	Name of the API method
Description	Description of the method
URL (method)	URL method in the format: [HTTP verb] [relative URL]
Request type	Request type
URL Params	Request string parameters
Data Params	Structure of the method request body. If the method does not have the request body, this section is skipped.

Success Response

Description of the response returned in the event of successful execution of the method. If the method uses the standard response format and does not return data, this section is skipped.

Error Response

Description of the response returned in the event of an error.

Notes

Block for unclear moments, comments and discussions on the topic

Standard Response Format

Unless the description specifies otherwise, a response from API for the method is received in the following format:

Response's status:

Possible response statuses (values of the field "status"):

StatusID	Description
0	The call completed successfully – data returned
1	The call completed with errors – error returned
2	Session terminated due to timeout
3	Security error

Success Response:

Status code and data are specified in case of a successful response.

Successful response contains mandatory fields:

<u>Name of the field</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
status	Status ID	int	1	0	0	
data	Data structure of the returned response	data		Without restrictions		

Further, in describing API, only the content of the field "data" will be described as the result of the successful call.

Response HTTP code: 200

Response structure:

```
{
  "status": 0,
  "data": ...
}
```

Error Response:

Status code and data are specified in case of an unsuccessful response. Unsuccessful response contains mandatory fields:

<u>Name of the field</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
status	Status ID	int	1	1,2,3	1	
error	Data structure of the returned unsuccessful response	error		Without restrictions		

Structure of the field "error":

<u>Name of the field</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
errorId	Error ID	int	int(max)	[0-9]	3	
errorText	Localised error text	string	500	Without restrictions	One-time password is incorrect	

Response HTTP code: 200

Response structure:

```
{
  "status": 1,
  "error": {
    "errorId": -20979,
    "errorText": "Incorrect DigiPass code. Remaining attempts left: "
  }
}
```

Further, in describing API, only the content of the field "error" will be described as the result of the erroneous call.

Notes

In the event of internal server errors, API methods may return an HTTP status code different from 200.

List of Permitted Symbols for the Document

&'_*,-

()+ /0123456789.,:;?ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz`АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдеёжзийклмнопрстуфхцчшщъыьэюяĀĒĢĶĻŠŅŪŽāēċģķīļšņūž

Information on the External Payment Code (Necessary for the SEPA Payment Service)

In accordance with the Regulation of the Bank of Latvia on Compiling Information about Non-bank External Payments, when making external payments residents of the Republic of Latvia shall encode every such payment exceeding EUR 10,000 according to the applicable classifier of external payments. The external payment code must be used by residents of the Republic of Latvia when making external payments (a transfer of resident's funds to a non-resident or an increase of external deposit and a transfer of non-resident's funds to a resident or a reduction of external deposit). Each purpose of an external payment is assigned with a three-digit payment code in accordance with international standards of payment balance. External payment codes can be found on the [home page](#).

1. Log in to the System

1.1. Receive a Security Token

Title	Receive a token
Description	The method creates and returns an authentication token.
URL (method)	/getAuthToken
Request type	GET
Header Params	TPP-Redirect-URI – URL address where to the request result is returned TPP-Redirect-URI-Unsuccessful – URL address whereto the result of an unsuccessful request is returned

Success Response:

Name	Description	Data type	Max. length	Permitted values	Example	Comments
authToken	<i>Token</i>	string	100	[0-9],[A-Z], [a-z], special symbols	PSD2AUTHD79EAFF455DF42512A00E81EE3F300FELXFPR	

Example:

```
{
  "status": 0,
  "data": {
    "authToken": "PSD2AUTHD79EAFF455DF42512A00E81EE3F300FELXFPR"
  }
}
```

Error response:

Example:

```
{
  "status": 3,
  "error": {
    "errorId": -20004,
    "errorText": "Security error"
  }
}
```

Notes

After the request has been successfully fulfilled, the customer must be redirected to the Internet Banking authentication page, [see request 1.2.](#)

The standard duration of a session is 15 minutes. Except if the customer has given his/her unambiguous consent to TPP's access without using strong customer authentication SCA.

1.2. Redirect the Customer to the Internet Banking Authentication Page

Description	Redirect the customer to the authorisation page
	1. Redirect to the address specified in the URL method, specifying the token received in request 1.1 in URL parameters.
	2. The Internet Banking authentication window will be displayed to the customer.
	3. The customer must enter his/her user code and Digipass code.
	4. After the customer has successfully logged in to the Internet Banking system, the customer provides an authorisation to TPP to access the selected account information therein.
	5. After an unambiguous consent has been received from the customer in the Internet Banking system, the access token and the consent ID are returned.

URL (method) <https://ib.ribbank.com/psd2start.cfm>

URL Params:

Name	Description	Data type	Max. length	Permitted values	Example	Comments
auth_token	Token received in the request 1.1 .	string	100	0-9,A-Z, a-z, special symbols	PSD2AUTHD79EAFF455DF42512A00E81EE3F300FELXFPR	

Success Response::

Name	Description	Data type	Max. length	Permitted values	Example	Comments
access_token	Access token	string	100	[0-9],[A-Z], [a-z], special symbols	PSD2ACCESS767663F73575F6DA6976F675B8DC19BBIX2E1	
consent_id	Consent ID	string	100	[0-9],[A-Z], [a-z], special symbols	PSD2CONSENT7BA03EB39A3687B68F2A263FAD8983ABCQOGU	

Example:

consent_id=PSD2CONSENT7BA03EB39A3687B68F2A263FAD8983ABCQOGU&access_token=PSD2ACCESS767663F73575F6DA6976F675B8DC19BBIX2E1

1.3. Establish Account Information Consent

The customer gives, revokes or changes and approves his/her consent after logging in to the Internet Banking system, specifying the following:

1. accounts which TPP with the AISP role may view;
2. how often AISP is authorised to view accounts specified in clause 1.

If the customer has given his/her consent, creating a recurrent session, AISP is ensured a possibility of viewing accounts specified by the customer up to 4 times a day and up to 90 days in a row, without strong customer authorisation (SCA).

If the customer modifies his/her consent in the Internet Banking system, the current session is automatically refreshed.

1.4. Refresh a Session (Refreshed Token)

Description		Refresh an interrupted session				
URL (method)		/refreshToken				
Request type		GET				
Success Response:						
Name	Description	Data type	Max. length	Permitted values	Example	Comments
authToken	Token	string	100	[0-9],[A-Z], [a-z], special symbols	PSD2AUTHD79EAF455DF42512A00E81EE3F300FELXFPR	
consent_id	Consent ID	string	100	[0-9],[A-Z], [a-z], special symbols	PSD2CONSENT7BA03EB39A3687B68F2A263FAD8983ABCQOGU	
Success Response::						
Name	Description	Data type	Max. length	Permitted values	Example	Comments
access_token	Access token	string	100	[0-9],[A-Z], [a-z], special symbols	PSD2ACCESS767663F73575F6DA6976F675B8DC19BBIX2E1	
Example:						
<pre>{ "status": 0, "data": {</pre>						

```
    "access_token": "PSD2ACCESS767663F73575F6DA6976F675B8DC19BBIX2E1"
  }
}
```

Error response:

Example:

```
{
  "status": 3,
  "error": {
    "errorId": 7,
    "errorText": "Incorrect consent id"
  }
}
```

2. Get Account Details of the List of Accessible Accounts

Title Get account details of the list of accessible accounts

Description The method returns available customer's accounts

URL (method) /accounts

Request type GET

URL Params:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
access_token	Customer's access token (received after the customer has successfully logged in to the Internet Banking system)	string	100	0-9,A-Z, a-z, special symbols	PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6	

Example:

access_token= PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6

Success Response:

An array "accounts" is returned, containing the following fields:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
accountNumber	Customers account number	string	21	[0-9], [A-Z]	LV36RIBR00000350N0000	
accountType	Account type	int	1	1 or 2	1	1 – current account 2 – card account
accountName	Account name assigned by the customer	string	100	According to the list of permitted symbols	My card account	
Example:						
<pre>{ "status": 0, "data": { "accounts": [{ "accountNumber": " LV36RIBR00000350N0000", "accountType": 1, "accountName": "My card account" }, { "accountNumber": "LV58RIBR00400040N0000", "accountType": 2, "accountName": "" }] } }</pre>						

Error response:

Example:

```
{
  "status": 3,
  "error": {
    "errorId": 8,
    "errorText": "Role should be AISP"
  }
}
```

3. Get Balances for a Given Account

Title	Get balances for a given account
Description	The method returns the balance of the account
URL (method)	/getBalanceByAccount
Request type	GET

URL Params:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
access_token	Customer's access token (received after the customer has successfully logged in to the Internet Banking system)	string	100	0-9,A-Z, a-z, special symbols	PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6	
account	Customers account number	string	21	From customers account list	LV36RIBR00000350N0000	

Example:

Access_token= PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6&account=LV65RIBR00500010N0000&CCY

Success Response:

An array "balances" is returned, containing the following fields:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
account	Customers account number	string	21	0-9,A-Z,	LV36RIBR00000350N0000	
timestamp	Time stamp	timestamp	-	Timestamp	2019-02-21 10:28:12.461292 +02:00	
currencyName	Currency code	string	3	[0-9]	USD	
availableBalance	Available balance	decimal	20.2	[0-9],[.]	10926.69	

Example:

```
{
  "status": 0,
  "data": {
```

```
"balances": [
  {
    "account": " LV36RIBR00000350N0000",
    "timestamp": "2019-02-21 10:28:12.461292 +02:00",
    "currencyName": "USD",
    "availableBalance": 10926.69
  },
  {
    "account": "LV06RIBR00400010N0000",
    "timestamp": "2019-02-21 10:28:12.461292 +02:00",
    "currencyName": "EUR",
    "availableBalance": 0
  }
]
```

Error response:

Example:

```
{
  "status": 3,
  "error": {
    "errorId": 8,
    "errorText": "Role should be AISP"
  }
}
```

4. Get Transaction Information for a Given Account

Title	Get transaction information for a given account
Description	The method returns the customer's account report
URL (method)	/ accounts/transactions
Request type	GET

URL Params:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
access_token	Customer's access token (received after the customer has successfully logged in to the Internet Banking system)	string	100	0-9,A-Z, a-z, special symbols	PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6	
accountnumber	Customers account number	string	21	From customers account list	LV36RIBR00000350N0000	
dateFrom	Start date (filtration)	date		YYYY-MM-DD	2018-01-01	
dateTo	End date (filtration)	date		YYYY-MM-DD	2019-05-24	

Example:

Access_token=PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6&accountNumber=LV65RIBR00500010N0000&dateFrom=2018-01-01&dateTo=2019-05-24

Success Response:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
accountNumber	Customers account number	string	21	[A-Z],[0-9]	LV36RIBR00000350N0000	
accountType	Account type	int	1	1,2	1	1 – current account 2 – card account
accountName	Account name assigned by the customer	string	100	According to the list of permitted symbols	My card account	LV65RIBR00500010N0000
currencyStatements		array				See the table below "currencyStatements"

An array "currencyStatements" is returned, containing the field "currency"

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
currency	Currency code	string	3	[0-9]	EUR	

And an array "transactions" with the following fields:

Name	Description	Data type	Max. length	Permitted values	Example	Comments
description	Transaction details	string	450	According to the list of permitted symbols	(#100) 201902261170001 Outgoing transfer Jānis Bērziņš LV47PARX0000115401017 JSC CITADELE BANKA (SWIFT PARXLV22XXX) For consulting services, invoice 123	documentId ((#100)), transaction ID (201902261170001), transaction direction (Outgoing transfer), Beneficiary/ payer name, surname/ name (Jānis Bērziņš), beneficiary/ payer account (LV47PARX0000115401017), beneficiary/payer bank (JSC CITADELE BANKA (SWIFT PARXLV22XXX), transaction details (For consulting services, invoice 123)
amount	Transaction amount	decimal	20,2	[0-9], [.]	300	
timestamp	Transaction date	date		YYYY-MM-DD	2019-02-22	
direction	Transaction direction	string	3	IN, OUT	OUT	IN- incoming OUT- outgoing
id	Transaction ID	string	20	0-9	201902261170001	

Example:

```
{
  "status": 0,
  "data": {
    "currencyStatements": [
      {
        "currency": "USD",
        "transactions": [
          {
            "description": "(#635 ) 000-813142 Currency exchange Exchange 12.24 USD <= 10.00 EUR , rate -
1.2235. KONVERTĀCIJA",
            "amount": 12.24,
            "timestamp": "2018-04-18",
            "direction": "IN",
            "id": "000-813142"
          }
        ]
      }
    ],
    {
      "currency": "EUR",
      "transactions": [
```



```

    {
      "description": "(#100 ) 201902261170001 Outgoing transfer Jānis Bērziņš LV47PARX0000115401017 JSC
CITADELE BANKA (SWIFT PARXLV22XXX ) For consulting services, invoice 123",
      "amount": 300.00,
      "timestamp": "2018-04-18",
      "direction": "IN",
      "id": "201804180460002"
    },
    {
      "description": "(#635 ) 000-813142 Currency exchange Exchange 12.24 USD <= 10.00 EUR , rate -
1.2235. KONVERTĀCIJA",
      "amount": 10.00,
      "timestamp": "2018-04-18",
      "direction": "OUT",
      "id": "000-813142"
    }
  ]
},
"accountNumber": "LV58RIBR00400040N0000",
"accountType": 1,
"accountName": ""
}
}

```

Error response:

Example:

```

{
  "status": 3,
  "error": {
    "errorId": 8,
    "errorText": "Role should be AISP"
  }
}

```

5. Get a Confirmation on the Availability of Funds

Title	Get a confirmation on the availability of funds
Description	The method returns a response whether the amount of funds specified in the request is available in the customer's account
URL (method)	/CheckAvaliableAmount
Request type	GET

URL Params:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
access_token	Customer's access token (received after the customer has successfully logged in to the Internet Banking system)	string	100	0-9,A-Z, a-z, special symbols	PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6	
account	Customers account number	string	21	From customers account list	LV36RIBR00000350N0000	
CCY	Currency code	string	3	[A-Z]	EUR	
amount	Requested amount	decimal	20.2	[0-9],[.]	500.00	

Example:

Access_token= PSD2ACCESSE528008709AB4AB12146144C4B187B623RSJ6&account= LV36RIBR00000350N0000&CCY=EUR&amount=500

Success Response:

Example:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
result	Note that the requested amount is available in the customer's account	boolean		true/false	true	True – the requested amount is available in the account False – the requested amount is not available in the account

Error response:

Example:

```

{
  "status": 3,
  "error": {
    "errorId": 8,
    "errorText": "Role should be PIISP"
  }
}

```

6. Initiation of a Single SEPA Payment

Title	Initiation of a single SEPA payment
Description	The method creates a new payment
URL(method)	/getAuthToken
Request type	POST
Header Params	TPP-Redirect-URI – URL address whereto the request result is returned TPP-Redirect-URI-Unsuccessful – URL address whereto the result of an unsuccessful request is returned

Data params:

The data array "document" must be created, containing the following fields:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Mandatory</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
amount	Payment amount	decimal	Yes	20.2	[0-9],[.]	500.00	
bnfName	Beneficiary name	string	Yes	140		Jānis Bērziņš	
bnfAccount	Beneficiary's account No.	string	Yes	34	[A-Z,0-9]		
bnfAddr	Beneficiary's address	string	Yes	140	According to the list of permitted symbols	Rīga, street Kalpaka 1	
residenceId	Beneficiary's residence	string	Yes	2	ISO 3166-1 alpha-2	LV	
details	Payment details	string	Yes	450	According to the list of permitted symbols	Payment details	
paycodeId	External payment code	int	No	3	[0-9]	111	See "Information on the External Payment Code (for SEPA Payment)"

Example:

```
{
  "document":
  {
    "amount": "852.23",
    "bnfAccount": "LV52NDEA0000084953089",
    "bnfName": "Jānis Bērziņš",
    "residenceId": "LV",
    "bnfAddr": "Rīga, street Kalpaka 1",
    "details": "Payment details"
    "paycodeId": "111"
  }
}
```

Success Response:

<u>Name</u>	<u>Description</u>	<u>Data type</u>	<u>Mandatory</u>	<u>Max. length</u>	<u>Permitted values</u>	<u>Example</u>	<u>Comments</u>
authToken	Access token	string	100	0-9,A-Z, a-z, special symbols	PSD2AUTHF6C8ACD44AAD8DCF8DEDDDA72C15E319A88EQ		

Example:

```
{
  "status": 0,
  "data": {
    "authToken": "PSD2AUTHF6C8ACD44AAD8DCF8DEDDDA72C15E319A88EQ"
  }
}
```

Error Response:

An array "documentErrors" is returned, containing the following fields:

Example:

```
{
  "status": 1,
  "documentErrors": [
    {
      "fieldName": "details",
      "errorText": "Field \"details\" should not be empty."
    }
  ]
}
```

```

    ],
    "error": {
      "errorId": 20979,
      "errorText": "Couldn't create document, because it have errors. "
    }
  }
}

```

Notes: After the payment is signed, it is executed in accordance with RIB regulations, i.e., the acceptance of the payment does not guarantee its execution. The fact that the payment has been successfully sent does not mean that it has been executed!

7. SEPA Payment Signing

Title	SEPA payment signing					
Description	Redirecting the customer to the Internet Banking authentication and payment signing page					
	<ol style="list-style-type: none"> 1. Redirect to the URL address: https://ib.ribbank.com/psd2_start.cfm 2. The Internet Banking authentication and payment signing window is returned to the customer: 3. The customer must log in to the Internet Banking system and sign the payment 					
URL (method)	https://ib.ribbank.com/psd2_sign.cfm					
Request type	GET					
URL params:						
Name	Description	Data type	Max. length	Permitted values	Example	Comments
authToken	Access token	string	100	0-9,A-Z, a-z, special symbols	PSD2AUTHF6C8ACD44AAD8DCF8DEDDDA72C15E319A88EQ	
Example:						
auth_token= PSD2AUTHF6C8ACD44AAD8DCF8DEDDDA72C15E319A88EQ						

Notes: After the payment is signed, it is executed in accordance with [RIB regulations](#), i.e., the acceptance of the payment does not guarantee its execution. The fact that the payment has been successfully sent does not mean that it has been executed!

History of Changes

<u>Version No.</u>	<u>Date of approval</u>	<u>Changes</u>
1.0.	30.05.2019	Initial version
1.1.	31.07.2019	<ul style="list-style-type: none">✓ Significant changes to the initiation and signing of SEPA payment;✓ New section "Customer's Consent Management" added;✓ Minor editorial changes