Content

References

This document represents Banca Transilvania's implementation of the PSD2 initiative following The Berlin Group's guidelines. For more information about these guidelines please visit https://www.berlin-group.org/nextgenpsd2-downloads Implementation Guidelines.

Authorization procedure was updated following Implementation Guidelines- Requirements on the OAuth2 Protocol with Proof Key for Code Exchange (PKCE). For more information about PKCE please visit https://tools.ietf.org/html/rfc7636

Register as an OAuth2 client

From the response to either POST Consent or POST Payment we find scaOAuth link:

```
"_links": {
    "scaOAuth": {
      "href": "https://api.apistorebt.ro/bt/sb/oauth/.well-known/oauth-authorization-server"
```

Following the link we find a list of endpoints, amongst which:

"registration_endpoint": "https://api.apistorebt.ro/bt/sb/oauth/register"

To register as a client, call the above method with parameters:

Headers:

- Accept: application/json
- Content-Type: application/json

Body:

```
{
 "redirect_uris": ["https://google.com"], - The list of URIs the TPP will redirect the user after authentication and consent /payment consent. Only https URIs are accepted.
 "client_name":"Third Party Provider Application DEMO 1"
}
```

The response body will contain:

```
{
  "redirect_uris": [
    "https://google.com"
  ],
  "token_endpoint_auth_method": "client_secret_basic",
  "grant_types": [
    "authorization_code",
    "refresh_token"
```

```
  ],
  "response_types": [
    "code",
    "none"
  ],
  "client_id": "example_client_id"
  "client_secret": "example_client_secret"
  "client_name": "Third Party Provider Application DEMO 1",
  "subject_type": "public",
  "tls_client_certificate_bound_access_tokens": false,
  "client_id_issued_at": 1651567307,
  "client_secret_expires_at": 0,
}
```


Account Information Service

1. To access a user's Account information, a Third-Party Provider must obtain the user's consent. To start this process, call POST Init Consent, endpoint:

https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/consents

Request body should follow this format:


```
{
   "access":
         {"availableAccounts": "allAccounts"},
   "recurringIndicator": true,
   "validUntil": "2022-07-28",
   "combinedServiceIndicator": false,
   "frequencyPerDay": 4
}
```


Where:
"recurringIndicator": true, - this parameter dictates if the consent generated will be used more than once
"validUntil": "2022-07-28", - this should be a date no more than 90 days in the future
"frequencyPerDay": 4 – This field indicates the requested maximum frequency for an access without PSU involvement per day (an access without PSU involvement is an API call missing header PSU-IP-Address). If header PSU-IP-Address is sent in the API call, then the TPP can call the API an unlimited number of times per day.

For this API call, header PSU-IP-Address is mandatory – to indicate user involvement, which is required for initiation calls.

2. From the response body TPP will receive a unique **consentID** and information about the authorization procedure:

Response Header: ASPSP-SCA-Approach →REDIRECT
"scaOAuth": { "href": https://api.apistorebt.ro/bt/sb/oauth/.well-known/oauth-authorization-server …}

- At this point a registered TPP will already have client id and client secret, as per OAuth2 standard.

3. TPP will redirect the user to BT authentication page. A redirect URL should be formed using this example:

https://apistorebt.ro/auth/realms/psd2-sb/protocol/openid-connect/auth?

client_id=example_client_Id

&redirect_uri=https://google.com

&response_type=code

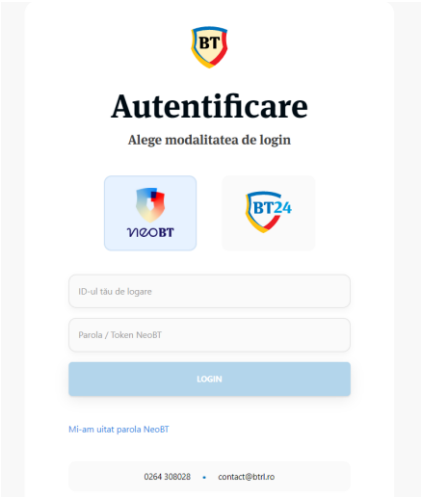
&scope=AIS: example_consentId

&state=state123test

&nonce=nonce123test

&code_challenge=4vw1eoEvbd9vdGcrINTeSKydqczt_3cxCuyaQZcNnkk

&code_challenge_method=S256



For testing purposes use any LoginID/Password.

4. User will use BT internet banking credentials to login and give consent on the list of accounts and actions (read transactions list, balances, account information) to be given to the TPP.

5. At the end of the authentication and consent flow, user is redirected to the redirect_uri mentioned above and a token will be generated for the TPP. Following Oauth2, this token will be exchanged for a Bearer code which will be used in all further calls.

Redirect URI example:

https://google.com/?state=state123test&session_state=c41a3095-b4cf-4783-90da-58367f3bbfcf&code= exampleCode

Note: Parameter “code” is exchanged for a Bearer token by calling authorization method: token_endpoint":" https://api.apistorebt.ro/bt/sb/oauth/token"

Request body should be sent in x-www-form-urlencoded format and should contain the following parameters:
• Code: exampleCode
• grant_type : authorization_code
• redirect_uri : https://google.com
• client_id: "example_client_id"
• client_secret: "example_client_secret"
• code_verifier: D_zlTiLARF0xO8wmtN4200gVuhsi6Ob3OAfZUQ2U69Z9j9EPZjbHjNbTKS-2dVgLxxzzAYVU3YHkG93m8zPtqCfnZRMMwLzKsyJPQ0NBXtaecuozXJeUQkOIzOkh4Y_X

Note: According to https://datatracker.ietf.org/doc/html/rfc7636, the code_verifier must have a minimum length of 43 characters and a maximum length of 128 characters

Authorization server will return a token:
• access_token: exampleToken - will be used in all following API calls:
• token_type: bearer
• refresh_token: exampleRefreshToken
• expires_in: 3599
• (…)

Using ConsentID and token the following methods can be called:

Consent information

1. Consent details: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/consents/{consentId} - Returns the content of an account information consent object.
2. Consent status: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/consents/{consentId}/status - Can check the status of an account information consent resource.
3. Revoke consent: DELETE https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/consents/{consentId} - Deletes a given consent.

Account information

1. Account list: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/accounts - returns the list of accounts for which the user has given consent
2. Account details: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/accounts/{account-id} - returns details about one of the accounts for which the user has given consent.
3. Transaction details: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/accounts/{account-id}/transactions/{ transactionId } - Reads transaction details from a given transaction addressed by "transactionId" on a given account addressed by "account-id".
4. Accounts balances: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/accounts /{account-id}/balances - Reads account data from a given account addressed by "account-id".
5. Transactions list for an account: GET https://api.apistorebt.ro/bt/sb/bt-psd2-aisp/v2/accounts /{account-id}/transactions – reads account data from a given account addressed by "account-id".

To this api call two additional query parameters were added to support pagination:

- Limit = 10 how many transactions per page. If this parameter is missing, the default and maximum value is 100 transactions per page.
- Page = 1 page number. If this parameter is missing, the default value is 1st page.

7-day limit between dateFrom and dateTo was removed.

dateFrom is still limited to maximum 120 days in the past.

Payment Service

User case 1: PSU knows IBAN code of the account used to make a payment.

In this case the IBAN should be sent in request body

1. To start the process of initiating a payment, call **POST Payment**.
   For RON payment:

   https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/ron-payment

Example of a request body (containing only the mandatory parameters) for a valid RON payment a TPP can use to test BT APIs:


```
{ "debtorAccount": {
"iban": "RO98BTRLRONCRT0ABCDEFGHI"
 },
 "instructedAmount": {
"currency": "RON",
"amount": "50" },
"creditorAccount": {
 "iban": " RO98BTRLEURCRT0ABCDEFGHI " },
"creditorName": "Creditor Name",
"debtorId": " J123456",
 "endToEndIdentification": "TPP Reference",
"remittanceInformationUnstructured": "Merchant reference" }
```


For Other currency payment:

https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/other-currency-payment

Example of a request body (containing only the mandatory parameters) for a valid other-currency payment a TPP can use to test BT APIs:


```
{ "debtorAccount": {
"iban": "RO98BTRLEURCRT0ABCDEFGHI "
 },
 "instructedAmount": {
"currency": "EUR",
"amount": "1" },
"creditorAccount": {
 "iban": " RO98BTRLEURCRT0ABCDEFGHJ " },
"creditorName": "Creditor Name",
"endToEndIdentification": "TPP Test Reference",
"remittanceInformationUnstructured": "detalii plata",
"creditorAgent": "BTRLRO22",
"creditorAgentName": "BT",
"creditorAddress": {
 "country":"Romania"

}
```


For Sandbox environment testing please use the parameters from this example.

For this API call, header PSU-IP-Address is mandatory – to indicate user involvement, which is required for initiation calls.

Response will contain:

- paymentId: 1658664d6a2d98ed70b129376ared
- "_links": { "scaOAuth": {

"href": "https://api.apistorebt.ro/bt/sb/oauth/.well-known/oauth-authorization-server"

At this point a registered TPP will already have client id and client secret, as per OAuth2 standard.

2. TPP will redirect the user to BT authentication page. A redirect URL should be formed using this example:

https://apistorebt.ro/auth/realms/psd2-sb/protocol/openid-connect/auth?

client_id= example_client_id

&redirect_uri=https://google.com

&response_type=code

&scope=PIS:1658664d6a2d98ed70b129376ared

&state=state123test

&nonce=nonce123test

&code_challenge=4vw1eoEvbd9vdGcrINTeSKydqczt_3cxCuyaQZcNnkk

&code_challenge_method=S256

3. User will use BT internet banking credentials to login and give consent on the payment (with the payment information mentioned above: amount, currency, debtorAccount, etc)

4. At the end of the authentication and consent flow, user is redirected to redirect_uri and a token will be generated for the TPP. Following Oauth2, this token will be exchanged for a Bearer code which will be used in all further calls.

Redirect URI example: https://google.com/?state=state123test&session_state=c41a3095-b4cf-4783-90da-58367f3bbfcf&code= exampleCode

Note: Parameter “code” is exchanged for a Bearer token by calling authorization method: token_endpoint":" https://api.apistorebt.ro/bt/sb/oauth/token"

Request body should be sent in x-www-form-urlencoded format and should contain the following parameters:

• Code: exampleCode
• grant_type : authorization_code
• redirect_uri : https://google.com
• client_id: example_client_id
• client_secret: example_client_secret
• code_verifier: D_zlTiLARF0xO8wmtN4200gVuhsi6Ob3OAfZUQ2U69Z9j9EPZjbHjNbTKS-2dVgLxxzzAYVU3YHkG93m8zPtqCfnZRMMwLzKsyJPQ0NBXtaecuozXJeUQkOIzOkh4Y_X

Note: According to https://datatracker.ietf.org/doc/html/rfc7636, the code_verifier must have a minimum length of 43 characters and a maximum length of 128 characters

Authorization server will return a token:
• access_token: exampleToken - will be used in all following API calls:
• token_type: bearer
• refresh_token: exampleRefreshToken
• expires_in: 3599
• (…)

Using PaymentID and token the following methods can be called:
a. Payment details: GET https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/ron-payment/{paymentId} – returns details about a payment initiated (amount, currency, credit account)
b. Payment status: GET https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/ron-payment/{paymentId}/status – returns the payment's status.

User case2: PSU has multiple accounts and wants to choose from which one to make the payment.

In this case Cont Debit IBAN should not be sent in request body, and user will choose an account from authorization page, after login.

1. To start the process of initiating a payment, call **POST Payment**.

For RON payment:

https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/ron-payment

Example of a request body (containing only the mandatory parameters) for a valid RON payment a TPP can use to test BT APIs:

```
{
 "instructedAmount": {
"currency": "RON",
"amount": "50" },
"creditorAccount": {
```

"iban": " RO98BTRLEURCRT0ABCDEFGHI " },
"creditorName": "Creditor Name",
"debtorId": " J123456",
"endToEndIdentification": "TPP Reference",
"remittanceInformationUnstructured": "Merchant reference" }

For Other currency payment:

https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/other-currency-payment

Example of a request body (containing only the mandatory parameters) for a valid other-currency payment a TPP can use to test BT APIs:

```
{
"instructedAmount": {
"currency": "EUR",
"amount": "1" },
"creditorAccount": {
"iban": " RO98BTRLEURCRT0ABCDEFGHJ " },
"creditorName": "Creditor Name",
"endToEndIdentification": "TPP Test Reference",
"remittanceInformationUnstructured": "detalii plata",
"creditorAgent": "BTRLRO22",
"creditorAgentName": "BT",
"creditorAddress": {
"country":"Romania"
}
```

For Sandbox environment testing please use the parameters from this example.

For this API call, header PSU-IP-Address is mandatory – to indicate user involvement, which is required for initiation calls.

Response will contain

- paymentId: b2f94b747333b46b5c54d484ccred
- "_links": { "scaOAuth": {

"href": "https://api.apistorebt.ro/bt/sb/oauth/.well-known/oauth-authorization-server"

- At this point a registered TPP will already have client id and client secret, as per OAuth2 standard.

2. TPP will redirect the user to BT authentication page. A redirect URL should be formed using this example:

https://apistorebt.ro/auth/realms/psd2-sb/protocol/openid-connect/auth?

client_id= example_client_id

&redirect_uri=https://google.com

&response_type=code

&scope=PIS:1658664d6a2d98ed70b129376ared

&state=state123test

&nonce=nonce123test

&code_challenge=4vw1eoEvbd9vdGcrINTeSKydqczt_3cxCuyaQZcNnkk

&code_challenge_method=S256

3. User will use BT internet banking credentials to login and give consent on the payment (with the payment information mentioned above: amount, currency, etc) and choose debtorAccount from a list of accounts.

4. At the end of the authentication and consent flow, user is redirected to redirect_uri and a token will be generated for the TPP. Following Oauth2, this token will be exchanged for a Bearer code which will be used in all further calls.

Redirect URI example:
https://google.com/?state=state123test&session_state=c41a3095-b4cf-4783-90da-58367f3bbfcf&code= exampleCode

Note: Parameter “code” is exchanged for a Bearer token by calling authorization method: token_endpoint":" https://api.apistorebt.ro/bt/sb/oauth/token"

Request body should be sent in x-www-form-urlencoded format and should contain the following parameters:

- Code: exampleCode
- grant_type : authorization_code
- redirect_uri : https://google.com
- client_id: example_client_id
- client_secret: example_client_secret
- code_verifier: D_zlTiLARF0xO8wmtN4200gVuhsi6Ob3OAfZUQ2U69Z9j9EPZjbHjNbTKS-2dVgLxxzzAYVU3YHkG93m8zPtqCfnZRMMwLzKsyJPQ0NBXtaecuozXJeUQkOIzOkh4Y_X

Note: According to https://datatracker.ietf.org/doc/html/rfc7636, the code_verifier must have a minimum length of 43 characters and a maximum length of 128 characters

Authorization server will return a token:
• access_token: exampleToken - will be used in all following API calls:
• token_type: bearer
• refresh_token: exampleRefreshToken
• expires_in: 3599
• (…)

Using PaymentID and token the following methods can be called:
a. Payment details: GET https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/ron-payment/{paymentId} – returns details about a payment initiated (amount, currency, credit account)
b. Payment status: GET https://api.apistorebt.ro/bt/sb/bt-psd2/v2/payments/ron-payment/{paymentId}/status – returns the payment's status.

Types of payments accepted through Payment Services:
a. Intra-bank payments in RON;
b. Inter-banks payments in RON;
c. External intra-bank payment in RON;
d. Intra-bank payments in EUR or other currencies;
e. Inter-bank payments in EUR or other currencies;

Funds confirmation

Returns confirmation of the availability of funds for a specific account.

1. To obtain access to funds availability information a Third-Party Provider must obtain the user's consent.

Request body should follow this format:


```
{
"account": { "iban": " RO98BTRLEURCRT0ABCDEFGHI " },
"cardNumber": "1234567891234",
"cardExpiryDate": "2022-12-31",
"cardInformation": "MyMerchant Loyalty Card",
"registrationInformation": "Your contract Number 1234 with MyMerchant is completed with the registration with your bank."
}
```

Where: "iban": " RO98BTRLEURCRT0ABCDEFGHI " - mandatory parameter and represents the user's account number.

For this API call, header PSU-IP-Address is mandatory – to indicate user involvement, which is required for initiation calls.

2. From the response body TPP will receive a unique consentID and information about the authorization procedure:

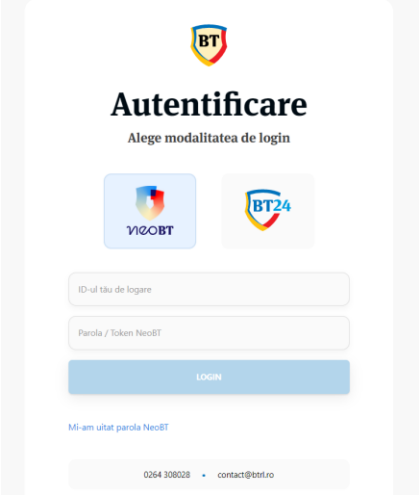
Response Header: ASPSP-SCA-Approach →REDIRECT

```
"scaOAuth": {
    "href": "https://api.apistorebt.ro/bt/sb/oauth/.well-known/oauth-authorization-server"
```

- At this point a registered TPP will already have client id and client secret, as per OAuth2 standard.

3. TPP will redirect the user to BT authentication page. A redirect URL should be formed using this example:

https://apistorebt.ro/auth/realms/psd2-sb/protocol/openid-connect/auth?
client_id= example_client_id
&redirect_uri=https://google.com
&response_type=code
&scope=PIISP: example_consentID
&state=state123test
&nonce=nonce123test
&code_challenge=4vw1eoEvbd9vdGcrINTeSKydqczt_3cxCuyaQZcNnkk
&code_challenge_method=S256



For testing purposes use any LoginID/Password.

4. User will use BT internet banking credentials to login and give consent on the account number.

5. At the end of the authentication and consent flow, user is redirected to the redirect_uri mentioned above and a token will be generated for the TPP. Following Oauth2, this token will be exchanged for a Bearer code which will be used in all further calls.

Redirect URI example:

https://google.com/?state=state123test&session_state=c41a3095-b4cf-4783-90da-58367f3bbfcf&code= exampleCode

Note: Parameter “code” is exchanged for a Bearer token by calling authorization method: token_endpoint":" https://api.apistorebt.ro/bt/sb/oauth /token"

Request body should be sent in x-www-form-urlencoded format and should contain the following parameters:

- Code: exampleCode
- grant_type : authorization_code
- redirect_uri : https://google.com
- client_id: example_client_id
- client_secret: example_client_secret
- code_verifier: D_zlTiLARF0xO8wmtN4200gVuhsi6Ob3OAfZUQ2U69Z9j9EPZjbHjNbTKS-2dVgLxxzzAYVU3YHkG93m8zPtqCfnZRMMwLzKsyJPQ0NBXtaecuozXJeUQkOIzOkh4Y_X

Note: According to https://datatracker.ietf.org/doc/html/rfc7636, the code_verifier must have a minimum length of 43 characters and a maximum length of 128 characters

Authorization server will return a token:

- access_token: exampleToken - will be used in all following API calls:
- token_type: bearer
- refresh_token: exampleRefreshToken
- expires_in: 3599
- (…)

Using ConsentID and token the following methods can be called:

Consent information

1. Consent details: GET https://api.apistorebt.ro/bt/sb/bt-psd2-piisp/v2/consents/confirmation-of-funds/{consentId} - Returns the content of an account information consent object.
2. Consent status: GET https://api.apistorebt.ro/bt/sb/bt-psd2-piisp/v2/consents/confirmation-of-funds/{consentId}/status - Can check the status of an account information consent resource.
3. Revoke consent: DELETE https://api.apistorebt.ro/bt/sb/bt-psd2-piisp/v2/consents/confirmation-of-funds /{consentId} - Deletes a given consent.

Confirmation of funds

POST https://api.apistorebt.ro/bt/sb/bt-psd2-piisp/v2/funds-confirmations

Request body should follow this format:

```
{
 "account": { "iban": " RO98BTRLEURCRT0ABCDEFGHI " },
"instructedAmount": {
"currency": "EUR",
"amount": "100"
}
 }
```

Token expiration:

In case access_token expires, a new token can be generated using refresh_token by calling token_endpoint: https://api.apistorebt.ro/bt/sb/oauth/token

Request body (in x-www-form-urlencoded format) should contain:

- Refresh_token= exampleRefreshToken
- grant_type= refresh_token
- redirect_uri= https://google.com
- client_id: example_client_id
- client_secret: example_client_secret
- code_verifier: D_zlTiLARF0xO8wmtN4200gVuhsi6Ob3OAfZUQ2U69Z9j9EPZjbHjNbTKS-2dVgLxxzzAYVU3YHkG93m8zPtqCfnZRMMwLzKsyJPQ0NBXtaecuozXJeUQkOIzOkh4Y_X

Note: According to https://datatracker.ietf.org/doc/html/rfc7636, the code_verifier must have a minimum length of 43 characters and a maximum length of 128 characters

Authorization server will return another token:

- access_token: exampleToken - **will be used in all following API calls:**
- token_type: bearer
- refresh_token: exampleRefreshToken
- expires_in: 3599
- (…)