API description and Rules – First to read

Contents

What is Consent?

An ‘account information service’ is an online service which provides consolidated information on payment accounts held by a payment service user with payment service providers. This will ensure that account information service providers (AISPs) can receive access to payment accounts, whilst also placing requirements on them to ensure security for users.

API endpoints

Link	Resource	Endpoints
Create consent	consents	POST /consents
Get Consent Request	consents	GET / consents /{ConsentId}
Consent status request	consents	GET / consents /{ ConsentId }/status
Delete Consent	consents	DELETE / consents /{ ConsentId }

How does it work?

Step 1: Setup Account Access Consent

- The AISP connects to UBB’s API Gateway and creates a consent resource
- This informs UBB that one of our PSUs is granting access to account and/or transaction information to an AISP
- UBB responds with an identifier for the resource, consent_Id - the intent identifier
- This step is carried out by making a POST request to /consents endpoint
- The consent resource may not include data the PSU has consented with the AISP
- The data needed for consented access to accounts will be entered on the next step, after PSU is redirected to UBB dedicated page for SCA

Step 2: Consent

- The AISP requests the PSU to give its consent to authorize the AISP to deliver services enabling access to account information and to respective information access (with balance information or transactions information or both)
- The AISP redirects the PSU to the UBB authentication page
- The redirect includes the consent_Id generated in the previous step
- This allows the UBB to correlate the account-access-consent that was setup
- The UBB authenticates the PSU by two factor method
- The PSU will be able to authorize or reject the account-access-consent details in its entirety. During authorization, the PSU selects accounts and accesses that are authorized for the AISP request in the UBB's banking interface
- After PSU confirmation, the UBB updates the state of the account-access-consent resource internally to indicate that the account access consent has been authorized and activated
- Once the consent has been authorized, the PSU can get consent details and status using respective API requests provided by AISP
- UBB uses Redirect method for applying SCA and supports callback mechanism. When AISP provide header TPP-Redirect-URI value in the request, after SCA procedure the customer will be returned back to this URI.
- The consent will be valid for the next 90 days. After this, the consent get status expired and cannot be used anymore
- In case the PSU wants to stop access over his active consent, just simple request for delete consent from AISP is needed. In this request consent_Id must be provided. Note that the consent can be deleted from the same AISP used for the creating consent procedure.

What is AIS?

An ‘account information service’ is an online service which provides consolidated information on payment accounts held by a payment service user with payment service providers. This will ensure that account information service providers (AISP) can receive access to payment accounts, whilst also placing requirements on them to ensure security for users.

API endpoints

Link	Resource	Endpoints
Accounts	accounts	GET /accounts GET /accounts/{AccountId}
Balances	balances	GET /accounts/{AccountId}/balances
Transactions	transactions	GET /accounts/{AccountId}/transactions
Transaction Details	transactions	GET /{account-id}/transactions/{resourceId}

How does it work?

UBB has decided to develop its APIs according to the BISTRA standard. Terminology used is therefore via this standard.

Step 1: Request Account Information

- The process begins with a PSU consenting to an AISP accessing their account information

Step 2: Request Data

- This is carried out by making a GET request the relevant resource
- The unique AccountId(s) that are valid for the consent will be returned with a call to GET /accounts
- This will always be the first call once an AISP has a valid access token
- The PSU could get information over balance, list of transactions or details for particular transaction. In any case in the interface between AISP and UBB the request must contain a valid consent_Id

What is PIS?

Under PSD2, a ‘payment initiation service’ is an online service which accesses a user’s payment account to initiate the transfer of funds on their behalf with the user’s authentication.

API endpoints

Link	Resource	Endpoints
Payment	domestic-credit-transfers-bgn domestic-budget-transfers-bgn sepa-credit-transfers cross-border-transfers	POST /{payment-product} GET /{payment-product}/{paymentId} GET /{payment-product}/{paymentId}/status

How does it work?

UBB has decided to develop its APIs according to the BISTRA standard. Terminology used is therefore via this standard.

Step 1: Setup Payment Order

- By PSU request, the PISP connects to UBB’s API Gateway and creates a payment-order resource
- This informs UBB that one of its PSUs intends to make a payment-order
- UBB responds with an identifier for the payment-order resource, the PaymentId, which is the intent identifier
- This step is carried out by making a POST request to the payment-order resource

Step 2: Authentication and Authorization

- The PISP redirects the PSU to the ASPSP to be identified and to authorize the initiated payment transaction from the designated payment account
- The redirect includes the PaymentId generated in the previous step
- UBB authenticates the PSU by two factor method
- This allows UBB to correlate the payment account from the resource created, with PSU access over this account
- The PSU selects the payment resource and must authorize the execution of the payment by two factor method
- UBB updates the state of the payment order resource internally to indicate that the payment has been authorized
- Once the payment is authorized, the PSU can get payment order details

- UBB uses Redirect method for applying SCA and supports callback mechanism. When PISP provide header TPP-Redirect-URI value in the request, after SCA procedure the customer will be returned back to this URI.

Step 3: Get Payment-Order/Details Status

- The PISP can check the status of the payment-order with the payment_Id resource identifier
- This is carried out by making a GET request to the payment-order resource

UBB Specific rules for PIS – read carefully first

In order to receive and execute the Payment order UBB will apply the rules as follow:

1. UBB has the right to refuse to execute a transfer for an amount exceeding BGN 30,000 or its equivalent in foreign currency, since at this stage the BISTRA Standard does not support the method of applying a transfer declaration. According to the local regulation, when ordering a transfer for an amount in BGN equivalent to over BGN 30,000, it is required to submit a declaration of origin of funds (under Art. 4, para. 7 and under Art. 6, para. 5, item 3 of the LMML).

2. Due to system requirement, UBB requires the following character table:
   - In case of domestic payments (local currency BGN) in the body of request the text is in CAPITAL LETTERS only. Other allowed symbols are: 0…9, space ( ), / \ ; & = % \ - * + : and dot(.). Cyrillic symbols in CAPITAL LETTERS are allowed as well.
   - In case of SEPA or Cross-border payments the text must be in LATIN CAPITAL LETTERS only. Other allowed symbols are space ( ), minus (–) and dot (.)
   - UBB supports Remittance information Unstructured field with limit value of 70 characters text. If the value is longer, you will receive error response.

   This requirements are applied to existing electronic channels (Online and Mobile banking) and not an obstacle over TPP's.

   Example 1. Request Body – with correct values


```
"instructedAmount": {"currency": "BGN", "amount": "123.50"},
"purposeCode": "GOVT",
"debtorAccount": {"iban": "BG94BANK12341234567890"},
"creditorName": "RECIEVER TD NRA SOFIA",
"creditorAccount": {"iban": "BG47BNBG96668123456789"},
"budgetPaymentDetails":{
    "regulatoryReportType": "1",
    "taxPayerId": "9904281234",
    "taxPayerType": "EGN",
    "paymentCategory": "110000",
    "fromDate": "20180101",
    "endDate": "20181231"},
"ultimateDebtor": "NAME SURNAME FAMILY",
"remittanceInformationUnstructured": "TAX PAYMENT "
```

Example 2. Request Body – with 3 incorrect values in red


```
"instructedAmount": {"currency": "BGN", "amount": "123.50"},
"purposeCode": "GOVT",
"debtorAccount": {"iban": "BG94BANK12341234567890"},
"creditorName": "Receiver TD NRA SOFIA",
"creditorAccount": {"iban": "BG47BNBG96668123456789"},
"budgetPaymentDetails":{
    "regulatoryReportType": "1",
    "taxPayerId": "9904281234",
    "taxPayerType": "EGN",
    "paymentCategory": "110000",
    "fromDate": "20180101",
    "endDate": "20181231"},
"ultimateDebtor": "Name SURNAME FAMILY",
"remittanceInformationUnstructured": "Tax payment information exceeds 70 characters text "
```


3. Currency account rules.

   As UBB does not support "currency convertor", you are not able to initiate payment from/to UBB account in Non BGN currency with API domestic-credit-transfer-BGN and domestic-budget-transfer-BGN.

   - From BGN account you can initiate API as follows:
     - API domestic-credit-transfer-BGN and domestic-budget-transfer-BGN to UBB accounts in BGN
     - API domestic-credit-transfer-BGN and domestic-budget-transfer-BGN to Non UBB accounts (other Bulgarian banks)
     - API SEPA credit transfer
     - API Crossborder-credit-transfer

   - From Non BGN account you can initiate API as follows:
     - API SEPA credit transfer
     - API Crossborder-credit-transfer

4. In case of sepa-credit-transfers.

   TPP is responsible to send payment details, according to SEPA requirements. UBB will execute this payment with fees and value date, according to the rules for cross-border transfer unless the payment details are not matching with SEPA requirements (both participant IBAN's to be from EU countries, currency must be "EUR", the charges must be shared, service level must be SEPA).

What is CoF?

The purpose of the COF API is it gives a TPP the possibility to check if the necessary funds are available on the payment account of the payer (eg. before execution a payment initiation). The bank has to respond with Y/N upon request from TPP whether there are sufficient funds on IBAN. In order to using the service, the customer needs to give the consent over the payment account with access to balance, transactions or both.

API endpoints

Link	Resource	Endpoints
Funds Confirmation	funds-confirmation	POST /funds-confirmations

How does it work?

UBB has decided to develop its APIs according to the BISTRA standard. Terminology used is therefore via this standard.

Step 1: Agree Funds Confirmation

- This process begins with a PSU committing to give consent, to their ASPSP to respond to confirmation of funds requests from the CBPII

Step 2: Setup Funds Confirmation Consent

- The procedure for setup Funds confirmation Consent is identical with Account Access Consent setup. UBB requires a consent over access to account for balance, transactions or both. This will grant CBPII to consume UBB Funds Confirmation API
- The CBPII connects to UBB’s API Gateway and creates a consent resource
- The CBPII redirects the PSU to the UBB Authentication page
- The redirect includes the consent_Id generated in the previous step
- This allows the UBB to correlate the account-access-consent that was setup
- The UBB authenticates the PSU by two factor method
- During authentication, the PSU selects accounts and accesses that are authorized for the CBPII request in the UBB's banking interface
- After PSU confirmation, UBB updates the state of the account-access-consent resource internally to indicate that the account access consent has been authorized and activated
- The consent will be valid for the next 90 days. After this term the consent get status expired and cannot be used anymore

Step 3: Confirm Funds

- The CBPII connects to UBB’s API Gateway and creates a funds-confirmation resource

- This informs UBB that the CBPII would like to receive confirmation of the availability of funds for the designated payment account
- UBB responds with a simple ‘yes’ (true) or ‘no’ (false) answer and not with a statement of the account balance (Boolean) for the resource
- This step is carried out by making a POST request to the /funds-confirmations endpoint, under an authorization code grant
- The setup payload will include these fields - which describe the data that the PSU has consented with the CBPII:
- Amount - the amount to be confirmed available
- Consent_Id - an Id that relates the request to a funds-confirmation-consent, and specific account with the ASPSP. This Id must match the intent identifier

What is SCA, scaRedirect link and Callback Redirect?

- SCA is abbreviation of Strong Customer Authentication. The elements of strong customer authentication categorized as ‘knowledge’ (something only the user knows), such as length or complexity, for the elements categorized as ‘possession’ (something only the user possesses), such as algorithm specifications, key length and information entropy, and for the devices and software that read elements categorized as ‘inherence’ (something the user is) such as algorithm specifications, biometric sensor and template protection features. The requirements of strong customer authentication apply to payments initiated or consent created by the user, regardless of whether the user is a natural person or a legal entity.
- scaRedirect link is element of the API’s response in cases of Consent creation or Payment initiation. This link should be used from TPP in order to redirect automatically the user to UBB application where SCA method can be applied.
- Callback Redirect functionality is supported by UBB with aim to return the user in his initial application, after SCA procedure. There are two requirements for successful back redirection of the user. First one is TPP to provide callback redirection link in API request. The second one is related to the customer, who needs to close the SCA application with proper Logout or Exit procedure.
  For example if the API request header element TPP-Redirect-URI is presented with value “TPPurlApp.com”, the customer will be redirected back to TPPurlApp.com.

Support

If you have any questions about our PSD2 APIs please check first the FAQ.
If not, you can always contact PSD2@UBB.BG.

What APIs are available?

The following APIs are available:
• AIS (Account Information Service)
• PIS (Payment Initiation Service)
• CIS (Confirmation of funds)
Further information regarding our APIs can be found after login.

Are the APIs secure?

Security is important to us. The sandbox and the live environment meet the latest security standards. We've employed the OAuth 2.0 consent model to enable clients to grant access to third party service providers (your application) in a clear and secure manner.

How do I access the technical documentation?

Once you have registered you will be able to log in to the Developer Portal to access the technical documentation of our APIs.

What can be done in the sandbox?

Our sandbox provides an opportunity for developers to test their app before building in production. The sandbox is a safe, secure environment where you can test your apps functionality and test our API integrations with your apps.

I am getting a 400 (unauthorized) HTTP status code. What went wrong?

- Request received without a mandatory parameter
- Request received with bad parameters (invalid format) values
- Request received with execution-date in past
- Request received with amount<=0
- Request received with currency other than EUR
- Request received with same account for sender and receiver
- Request received with Incorrect Beneficiary account OR incorrect structured communication
- Request received with invalid execution-date (weekend e.g.)

I am getting a 401 (unauthorized) HTTP status code. What went wrong?

- Request received with invalid values for Oauth2 Security related parameters
- Request received with invalid access_token

I am getting a 404 (RESOURCE_UNKNOWN) HTTP status code. What went wrong?

- In case of **Get Consent** Request: The consent is not in “valid” status. The reasons can be different:
  - The consent is not activated by the customer yet. SCA procedure is required.
  - The consent is in “expired” status.
  - The consent is deleted by the customer

  We recommend in the flow first to see the current status by using API Consent status request.
- In case of **GET /accounts**/{AccountId}/balances or GET /accounts/{AccountId}/transactions
  - The account is not consented for balance or transactions access
  - There are no transactions over account for the period specified
  - Account is blocked for some reason
- In case of **POST payments** request
  - The account is not allowed to be debited
  - Insufficient funds over account
  - The payment amount exceeds 30 000 BGN or its equivalent in foreign currency