API description and Rules – First to read

Contents

API description and Rules – First to read 1
What is Consent? 2
API endpoints 2
How does it work? 2
What is AIS? 4
API endpoints 4
How does it work? 4
What is PIS? 6
Single payments API endpoints 6
How does it work? 6
Bulk payments API endpoints 7
How does it work? 7
UBB Specific rules for PIS – read carefully first 8
List of supported functionalities in UBB Open banking channel according to PSD2 requirements. Version Jan.2026 11
Payments 11
Bulk Payments 12
Account information 12
What is SCA, scaRedirect link and Callback Redirect and App to App redirect? 14
Support 15
I am getting a 400 (Invalid or Bad request) HTTP status code. What went wrong? 15
I am getting a 400 (unauthorized) HTTP status code. What went wrong? 15
I am getting a 401 (unauthorized) HTTP status code. What went wrong? 16
I am getting a 404 (RESOURCE_UNKNOWN) HTTP status code. What went wrong? 16
What happened with the payment ordered by application – payment statuses 16

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 180 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.
- In any case in the interface between AISP and UBB the request must contain a valid consent_Id
- The PSU could get information over balance. UBB provides 4 balance types
  - Expected balance
  - Opening balance
  - Closing balance
  - Interim Available balance

- The PSU could get list of transactions. UBB apply the following specifics in the transaction list report:
  - Paginated account report. The information over account movement is provided by blocks contains up to 1000 records.
    PSU can access next 1000 records using provided link NEXT in the _links array
    PSU can access previous 1000 records using provided link PREVIOUS in the _links array
    PSU can access first 1000 records using provided link FIRST in the _links array
    In case of no movements in the block the response contains empty value {}

  - Remittance information contains the type of operation e.g. Card transaction, Transfer, Transaction Fee etc. In order to obtain the detailed information fulfilled from the debtor side, the API for transaction details must be used.

- Details for particular transaction. The purpose of the API to obtain the full details and remittance information send by debtor side. It is good practice to combine this and transaction list API using the unique transaction 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.

Single payments API endpoints

Link	Resource	Endpoints
payments	sepa-credit-transfers cross-border-transfers	POST /{payment-product} GET /{payment-product}/{paymentId} GET /{payment-product}/{paymentId}/status

How does it work?

UBB developed 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 with dynamic linking code 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

Bulk payments API endpoints

UBB supports BULK payments functionality with related APIs according to the BISTRA standard.

Link	Resource	Endpoints
bulk-payments	sepa-credit-transfers	POST /{payment-product} GET /{payment-product}/{paymentId}/status

How does it work?

The bulk payment is array containing several single payments. All payments are executed from one debit account. In one bulk only one type of single payments can be presented. All single payments must be in one currency. After successful execution of the bulk payment, the customer is able to see all single transactions in related AIS APIs. UBB developed its APIs according to the BISTRA standard.

Due to a complexity and various scenarios in order of bulk payments usage we provide the related information below:

Remark 1: What types of bulk payments is supported

UBB supports 2 types of bulk payments:

- sepa-credit-transfers
- sepa-credit-transfers toward budget accounts

Remark 2: The volume of single payments in an array of one bulk payment

The volume depends from payment type of the bulk payment. Here you can find the recommended volumes:

- sepa-credit-transfers – up to 75 single transactions in one bulk

Remark 3: What happens in case of error API response

The whole bulk payment is rejected and none of the single payments is accepted for further processing.

Remark 4: The specifics when SCA procedure is applied

When the bulk-payments API request is accepted, the SCA_redirect link will lead the customer to UBB SCA page for bulk payment confirmation.
On this page the customer can see or must do:

- The total count of single transactions in the bulk payment
- The total amount of fees due for the bulk payment
- The total amount of the bulk payment
- In case total amount of the bulk payment exceeded equivalent of 30 000 BGN, the customer is asked to upload the necessary additional documents related to Bulgarian regulation for payment services. This is the reason UBB to support bulk-payments SCA by web-based solution. SCA for bulk payment via mobile device is not supported
- In case of error returns during authorization by SCA, the whole bulk payment is rejected and none of the single payments is executed.

Remark 5: Timeout for bulk-payment processing

The maximum time processing is 120 seconds. This timeout is valid for both – API processing and SCA confirmation procedure. Please use recommended volumes for bulk payments initiation.

Remark 6: Get bulk-payment Status

- The PISP can check the status of the bulk payment with the payment_Id resource identifier
- This is carried out by making a GET request to the bulk-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 execution of a transfer for an amount exceeding BGN 30,000 or its equivalent in foreign currency. 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 in the body of request the text is in CAPITAL LETTERS only.
     - Other allowed symbols are: 0…9, space (), / \ = % \ - * + and dot(.). Please be in mind the double slash combination // is forbidden.

- 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 0…9, 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"},
"debtorAccount": {"iban": "BG94BANK12341234567890"},
"creditorName": "RECIEVER TD NRA SOFIA",
"creditorAccount": {"iban": "BG47BNBG96668123456789"},
"ultimateDebtor": "EIK-123456789-442100",
"remittanceInformationUnstructured": "TAX PAYMENT "

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

"instructedAmount": {"currency": "BGN", "amount": "123.50"},
"debtorAccount": {"iban": "BG94BANK12341234567890"},
"creditorName": "Receiver TD NRA SOFIA",
"creditorAccount": {"iban": "BG47BNBG96668123456789"},
"ultimateDebtor": "Eik-123456789-442100",
"remittanceInformationUnstructured": "Tax payment information exceeds 70 characters text "

3. In case of sepa-credit-transfers.

   TPP is responsible to send payment details, according to SEPA requirements:

   - Both parties' debtor and creditor account must be presented with IBAN
   - Both participant IBAN's to be from EU countries,
   - Currency must be "EUR",
   - The charges must be shared (SHAR),
   - Service level must be SEPA.

   UBB supports the requirement to calculate creditor bank BIC from creditor account IBAN. It means that is not necessary the "creditor agent" element to be presented in the API request.

4. **SEPA Instant** transfer

   Using the same API of sepa-credit-transfers.

   TPP is responsible to send payment details, according to SEPA requirements:

   - Both parties' debtor and creditor account must be presented with IBAN
   - Both participant IBAN's to be from EU countries,
   - Currency must be "EUR",
   - The charges must be shared (SHAR),
   - Service level must be **INST**.

UBB supports the requirement to calculate creditor bank BIC from creditor account IBAN. It means that is not necessary the “creditor agent” element to be presented in the API request.

5. Cutoff time for SEPA and Cross-border transfers.
   In order of execution of cross-border, single SEPA or bulk SEPA transfer, PSU must keep the rule for cutoff time. Such payments will be accepted by API request 24/7, but must be signed and authorized by SCA procedure in the appropriate cutoff time 08:00 – 15:30 (BG time zone). The payments authorized out of this interval will be rejected or not completed. It will be a good practice the TPP to remind the customer that SEPA and Cross-border transfer are subject of cutoff times policy.

List of supported functionalities in UBB Open banking channel according to PSD2 requirements. Version Jan.2026

Payments

1.1. SEPA Credit transfer toward accounts in and out of UBB

Supported cases

- Debtor account presented in the API interface - supported
- Debtor account not presented in the API interface - supported
- SEPA payment system - When API field serviceLevel provided with value SEPA
- **SEPA Instant payment system** - When API field **serviceLevel** provided with value **INST**
- SEPA payment to the budget accounts using the field "ultimateDebtor"
  Example: "ultimateDebtor" : "EIK-123456789-442100",
- Payment with future value date – Will be executed by SEPA payment system

1.2. Credit transfer in other currencies between accounts in UBB

Supported cases

- Debtor account presented in the API interface - supported
- Debtor account not presented in the API interface - supported
- Payment with future value date – When debtor and creditor accounts are the same currencies as the instructed currency

1.3. Credit transfer in other currencies toward accounts out of UBB

Supported cases

- Debtor account presented in the API interface - supported
- Debtor account not presented in the API interface - supported
- By SWIFT payment system
- By TARGET payment system - When API field serviceLevel provided with value URGP
- Payment with future value date – When debtor and creditor accounts are the same currencies as the instructed currency

1.4. Get payment status and payment details

Supported cases

- Payment status of specific transfer can be provided by transaction ID
- Basic payment details and status of specific transfer can be provided by transaction ID after the final customer confirmation

1.5. Payment status codes

Supported status codes:

RCVD – The payment instruction is received and waiting for signing from the final customer
ACCP – The payment instruction is received and signed from the final customer, but the transfer is not completed yet
ACSC – The payment instruction is processed after signing from the final customer and the transfer is completed successfully
RJCT – The payment instruction is rejected and not processed

Empty “” – The payment instructions were not accepted from the bank

Bulk Payments

2.1. Bulk SEPA credit transfer

Supported cases

- Between accounts in and out of UBB
- Toward budget accounts in and out of UBB
- Debtor account presented in the API interface

Account information

3.1. Consent creation

Supported cases

Accounts is not mandatory in the consent request

Valid period 180 days

3.2. Account details

Supported fields information for all accounts consented

"iban": "BG58UBBS800210xxxxx7511",
"bban": "800231xxxxx511",
"currency": " EUR",
"name": "NAME Company LTD ",
"product": "CURRENT ACCOUNT WITH OVERDRAFT"

3.3. Account list

Supported fields information for all accounts consented

"iban": "BG58UBBS80021xxxxx7511",
"currency": "BGN",
"name": "NAME Company LTD ",
"product": "CURRENT ACCOUNT WITH OVERDRAFT"

3.4. Read Balances

Supported fields information for specific account consented

"iban": "BG58UBBS80021xxxxx7511",
"name": "NAME Company LTD ",
"product": "CURRENT ACCOUNT WITH OVERDRAFT"
"balanceAmount": { "currency": " EUR", "amount": "66710.75" },
"balanceType": "openingBooked",
"balanceType": "closingBooked",
"balanceType": "interimAvailable",
"balanceType": "expected",

3.5. Read transaction list

Supported pagination by 1000 records for page

Supported fields:

transactionId": "206958394534334.060002",
entryReference": "FT2424103BCWFVGZ",
bookingDate": "2025-08-28",
valueDate": "2025-08-28",
transactionAmount": { "currency": " EUR", "amount": "-11.06" },
creditorName": "RECEIVER NAME",
creditorAccount": { "iban": "BG83TBIB931010xxxx0600" },
debtorName": "DEBTOR NAME ",
remittanceInformationUnstructured": "OUTGOING TRANSFER",
proprietaryBankTransactionCode": "244",

3.6. Read transaction details

Supported fields:

"transactionId": "206958394534334.060002",
"entryReference": "FT2424103BCWFVGZ",
"bookingDate": "2025-08-28",
"valueDate": "2025-08-28",
"transactionAmount": { "currency": "EUR", "amount": "-11.06" },
"exchangeRate": [
{ "currencyFrom": "EUR",
"rateFrom": "1.000",
"currencyTo": "EUR",
"rateTo": "1.000" }
],
"creditorName": "RECEIVER NAME OTHER BANK",
"creditorAccount": { "iban": "BG83TBIB931010xxxx0600" },
"ultimateCreditor": "RECEIVER NAME OTHER BANK",
"debtorName": " DEBTOR NAME ",
"ultimateDebtor": "ULTIMATE DEBTOR NAME ",
"remittanceInformationUnstructured": "TEST 5 NEXT CREDITOR NAME UNKNOWN NEUR11.06",

What is SCA, scaRedirect link and Callback Redirect and App to App 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.

Example:
"scaRedirect": {
"href": "https://ebb.ubb.bg/log.aspx?paymentId=FT2424nnnnnWT2&redirectBack=https://google.bg" }

- 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.
- UBB supports scaRedirect method for online and mobile devices based on Android, HMS and iOS operating systems.

Applying SCA via UBB Online banking

Supported cases

- The SCA procedure follows the access rights stored in UBB Online solution
- Multiple SCA is allowed to all types of initiated transfers
- SCA methods supported via SMS, Hardware token, Software token (mobile), Push method

Applying SCA via UBB Mobile banking

Supported cases

- The SCA procedure follows the access rights stored in UBB Mobile application
- SCA methods supported via Software token (biometric / pin code)
  Multiple SCA is not supported in Mobile App yet

Support

If you have any questions about our PSD2 APIs please check first the FAQ.

If not, you can always contact PSD2_Support@ubb.bg.

What APIs are available?

The following APIs are available:

- AIS (Account Information Service)
- PIS (Payment Initiation Service)

  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 (Invalid or Bad request) HTTP status code. What went wrong?

- Mandatory header element Content-Type is not presented with value "application/json"
- Mandatory header element X-Request-ID is not presented
- Mandatory header element Date is not presented in the next format: "Fri, 19 Aug 2022 11:42:22 GMT"

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}

- The account is not consented for balance or transactions access
- Account is blocked or unavailable for some reason

- In case of **POST payments** request

- The ordering account is not allowed to be debited
- Insufficient funds over ordering account
- The payment amount exceeds 30 000 BGN or its equivalent in foreign currency
- In case of internal UBB transfer credit account currency not match with transaction currency

What happened with the payment ordered by application – payment statuses

UBB supports variety of payments statuses in the response of the API GET payment status. The statuses are equal and valid for single and bulk payments

- With payment status "**RCVD**", UBB confirms correct receiving the payment order initiated by TPP
- UBB supports status "**ACCP**" in case of SEPA or cross-border payment. The status means that SEPA or cross-border payment **waits** the last technical bank procedure **to be finally processed**. The status appears only after successful SCA procedure for payment signing.
- UBB supports status "**ACSC**" when the payment is successfully signed and processed by the payment system.
- UBB response with status "**RJCT**" when the payment is successfully signed, but rejected by payment system.
- If UBB return the response with empty status "". It means the payment was not signed timely for further processing. In this case the payment is already invalid and should be reordered again.