DenizBank AG

Developer Guide

Version: 2.4

External Document

Disclaimer:

AG assumes no liability for claims or damage of any type caused by using this website, unless DenizBank AG has demonstrably caused such damage deliberately or through its gross negligence, and only to the proportional extent of its contribution to the damage caused. In particular, unless it is culpable or grossly negligent, DenizBank AG shall not be liable for any and all damage connected with interruptions to the user's hardware and software required for displaying the homepage or through the failure to establish a connection with DenizBank AG. Furthermore, DenizBank AG assumes no liability for damage caused through any and all delays or stray connections for which it

Version History

Version	*Date*	*Author*	*Changes*
1.0	01.09.2019	Open Banking Dept.	First version
1.1	01.09.2022	Open Banking Dept.	Flow changes, changes on service details
2.0	29.05.2023	Open Banking Dept.	Adaptation of Berlin Group Standard v.1.3.12
2.1	13/06/2023	Open Banking Dept.	Balance type details are appended
2.2	23/06/2023	Open Banking Dept.	Reference to Fortuna Documentation is added to explain backend processes
2.3	10/07/2023	Open Banking Dept.	Additional two methods added for authorization as OAuth approach
2.4	08/02/2024	Open Banking Dept.	Periodic and future payments ,delete payments and refresh token methods added.

Table of Contents

List of Tables

No table of figures entries found.

Table of Figures

No table of figures entries found.

1. Introduction

This documentation provided by Intertech AS as service provider of Denizbank AG is based on the Berlin Group Standard. Deviations are marked accordingly. Work in progress.

Glossary

Acronym	Description
ASPSP	Account Servicing Payment Service Provider; Provides and maintains customer accounts from which payments can be made.
PISP	Payment Initiation Service Provider; Initiates a payment order at the request of the user, from a payment account held at another payment services provider.
PIS	Payment Initiation Services
AISP	Account Information Service Provider
AIS	Account Information Service
TPP	Third Party Provider; Executes the services defined by PSD2 on behalf of a PSU.
PSU	Payment Service User; Could be a natural or legal person under PSD2 legislation.
API	Application Program Interface
SCA	Strong Customer Authentication
PSD	Payment Service Directive

Use cases supported for the core services

Use case	Service	Role of the TPP	Support optional	PSU directly involved
Initiation of a single payment	Payment initiation service	PISP	no	yes
Initiation of a future dated single payment	Payment initiation service	PISP	yes	yes
Initiation of a regular payment	Payment initiation service	PISP	yes	yes
Establish account information consent	Account information service	AISP	no	yes
Get list of reachable accounts	Account information service	AISP	no	no
Get account details of the list of accessible accounts	Account Information service	AISP	no	no
Get balances for a given account	Account information service	AISP	no	no
Get transaction information for a given account	Account information service	AISP	no	no
Get balances for a given card account	Account information service	AISP	yes	no
Get a confirmation on the availability of funds	Funds confirmation service	PIISP	no	no

2. Summary

The NextGenPSD2 Framework Version 1.3.12 offers a modern, open, harmonized and interoperable set of Application Programming Interfaces (APIs) as the safest and most efficient way to provide data securely. The NextGenPSD2 Framework reduces XS2A complexity and costs, addresses the problem of multiple competing standards in Europe and, aligned with the goals of the Euro Retail Payments Board, enables European banking customers to benefit from innovative products and services ('Banking as a Service') by granting TPPs safe and secure (authenticated and authorized) access to their bank accounts and financial data.

The possible Approaches are:

- Redirect SCA Approach
- **OAuth2 SCA Approach** * Denizbank AG supports this approach
- Decoupled SCA Approach
- Embedded SCA Approach without SCA method
- Embedded SCA Approach with only one SCA method available
- Embedded SCA Approach with Selection of a SCA method

Not every message defined in this API definition is necessary for all approaches. Furthermore, this API definition does not differ between methods which are mandatory, conditional, or optional. Therefore, for a particular implementation of a Berlin Group PSD2 compliant API it is only necessary to support a certain subset of the methods defined in this API definition.

Please have a look at the implementation guidelines if you are not sure which message has to be used for the approach you are going to use.

General Remarks Related to the Open Api Specification:

- This API definition is based on the Implementation Guidelines of the Berlin Group PSD2 API. It is not a replacement in any sense. The main specification is (at the moment) always the Implementation Guidelines of the Berlin Group PSD2 API.
- This API definition contains the RESTful-API for requests from the PISP and AISP to the ASPSP.
- This API definition contains the messages for all different approaches defined in the Implementation Guidelines.
- There are several predefined types which might occur in payment initiation messages, but are not used in the standard JSON messages in the Implementation Guidelines. Therefore, they are not used in the corresponding messages in this file either. We added them for the convenience of the user. If there is a payment product, which needs these fields, one can easily use the predefined types. But the ASPSP need not to accept them in general.

- We omit the definition of all standard HTTP header elements (mandatory/optional/conditional) except they are mentioned in the Implementation Guidelines. Therefore, the implementer might add these in his own realization of a PSD2 compliant API in addition to the elements defined in this file.

- Our system has no support of session at the XS2A interface for AIS and PIS Services.

General Remarks on Data Types

The Berlin Group definition of UTF-8 strings in context of the PSD2 API has to support at least the following characters

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : ( ) . , ' +

Space

3. API Access Methods

API Access Methods	*Endpoints/Resources*	*Mthd*	*Cond.*	*Supported?*
Accounts	Accounts	GET	M	Supported
Accounts	accounts?withBalance	GET	O	Not yet Supported
Accounts	accounts/{account-id}	GET	M	Supported
Accounts	accounts/{account- id}?withBalance	GET	O	Not yet Supported
Accounts	accounts/{account-id}/balances	GET	M	Supported
Accounts	accounts/{account-id}/transactions	GET	M	Supported
Accounts	accounts/{account-id}/transactions?withBalance	GET	O	Not yet Supported
Accounts	accounts/{account-id}/transactions/{transaction Id}	GET	O	Not yet Supported
payments	payments/{payment-product}	POST	M	Supported
payments	payments/{payment-product}/{paymentId}	GET	M	Supported
payments	payments/{payment-product}/{paymentId}/status	GET	M	Supported
payments	bulk-payments/{payment- product}	POST	O	Not yet Supported
payments	bulk-payments/{payment-product}/{paymentId}	GET	M	Not yet Supported
payments	bulk-payments/{payment-product}/{paymentId}/status	GET	M	Not yet Supported
payments	periodic- payments/{payment- product}	POST	O	Supported
payments	periodic-payments/{payment-product}/{paymentId}	GET	M	Supported
payments	periodic-payments/{payment-product}/{paymentId}/status	GET	M	Supported
payments	{payment-service}/{payment-product}/{paymentId}/authorisations	POST	M	Not yet Supported

payments	{payment-service}/{payment-product}/{paymentId}/authorisations	GET	M	Not yet Supported
payments	{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}	PUT	C	Not Supported
payments	{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}	GET	M	Not yet Supported
payments	{payment-service}/{payment-product}/{paymentId}	DELETE	O	Not yet Supported
payments	{payment-service}/{payment-product}/{paymentId}/cancel lation-authorisations	POST	O	Not yet Supported
payments	{payment-service}{payment-product}/{paymentId}/cancel lation-authorisations	GET	O	Not yet Supported
payments	{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations/{authorisatio nId}	PUT	C	Not Supported
payments	{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations/{authorisationId}	GET	M	Not yet Supported
card accounts	card-accounts	GET	O	Not yet Supported
card accounts	card-accounts/{account-id}	GET	O	Not yet Supported
card accounts	card-accounts/{account- id}/balances	GET	O	Not yet Supported
card accounts	card-accounts/{account-id}/transactions	GET	M	Not yet Supported
Consents	Consents	POST	M	Supported
Consents	consents/{consentId}	GET	M	Supported
Consents	consents/{consentId}	DELETE	M	Supported
Consents	consents/{consentId}/status	GET	M	Supported
Consents	consents/{consentId}/authorisations	POST	M	Not yet Supported

Consents	consents/{consentId}/authorisations	GET	M	Not yet Supported
Consents	consents/{consentId}/authorisations/{authorisationId}	PUT	C	Not Supported
Consents	consents/{consentId}/authorisations/{authorisationId}	GET	M	Not yet Supported
Consents	oauth/.well-known/oauth-authorization-server	GET	M	Supported
Consents	authorize?response_type=code&client_id={client_id}&scope={scope}&state={state}&redirect_uri={redirect_uri}&code_challenge_method={code_challenge_method}&code_challenge={code_challenge}	GET	M	Supported
Consents	token?client_id={client_id}&grant_type=authorisation_code&code={code}&code_verifier={code_verifier}&redirect_uri={redirect_uri}	GET	M	Supported
confirmation-of-funds	funds-confirmations	POST	M	Supported
signing-baskets	signing-baskets	POST	O	Not yet Supported
signing-baskets	signing-baskets/{basketId}	GET	O	Not yet Supported
signing-baskets	signing-baskets/{basketId}	DELETE	O	Not yet Supported
signing-baskets	signing- baskets/{basketId}/status	GET	O	Not yet Supported

Table 1: End Points

Conditions:

M: Mandatory

O: Optional

C: Mandatory for Embedded SCA Approach, Conditional for other approaches

Consent

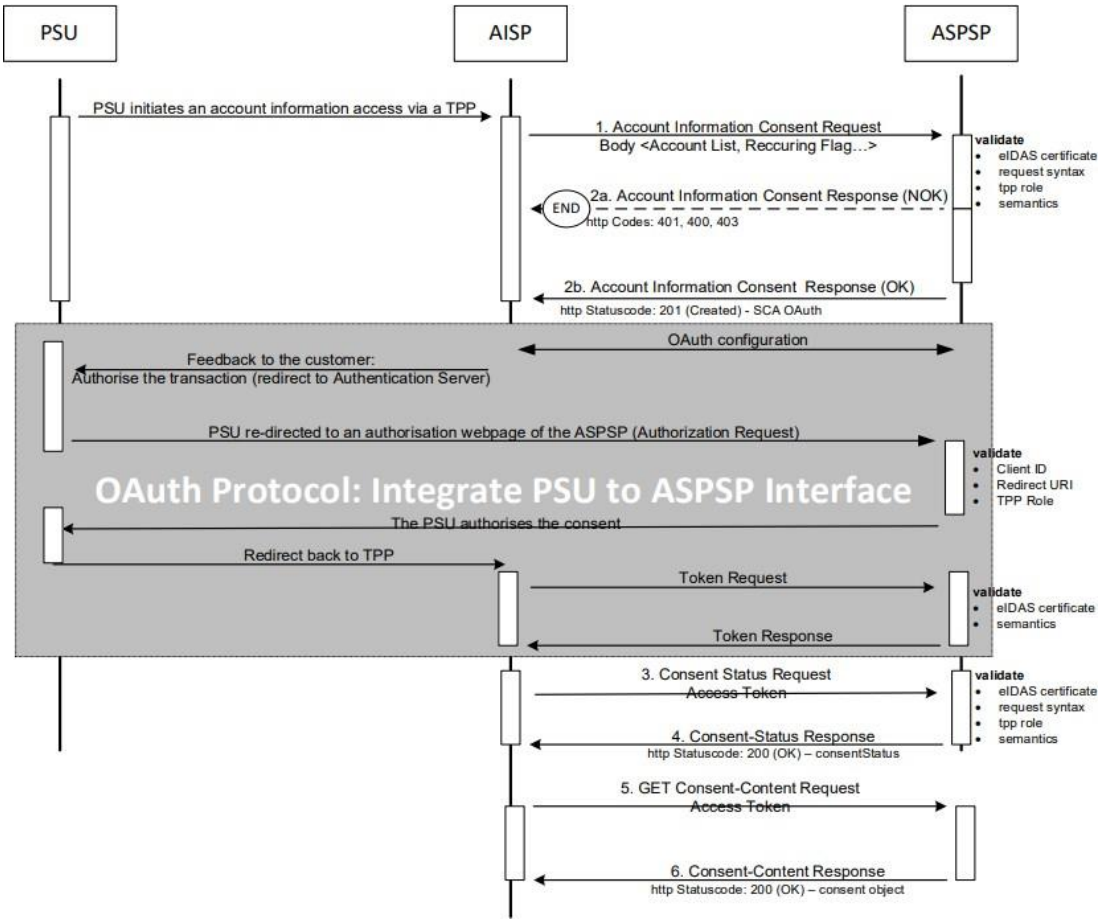
If TPP does not have a valid consent, TPP can create by the "Create Consent" method. TPP can verify whether the existing consent is valid with the "Get Consent Status" method or TPP can query the consent status in more detail with the "Get Consent Details" method.

Then If TPP has a valid consent for PSU and after getting PSU approvals, TPP calls “Get Access Token” method to get token. Consent ID is required only for AIS and Confirmation of Funds requests.

The PSU will have to authorize the access in an authorization page of the ASPSP (bank). If the authorization is successful, he/she will be redirected back to the TPP page. The consent is valid until the ‘validUntil’ date expires. This request will generate a unique consent ID.

We provide services in order to create consent, delete consent, get status of consent and get consent.

Below schema works for all consent methods. There is integration with Konsentus to validate QWAC certificate.



Consent Flow Sequence Diagram

Create Consent

This method creates a consent, granting access rights to dedicated accounts of a given PSU. Use POST to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/consents

Prod Endpoint: https://openbanking.denizbank.at/api/v1/consents

Request:

Request parameter:

Request Header:

Field	*Data Type*	*Condition*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Redirect-URI	String	M	Represents URI link that TPP wants to register for redirection after successful completion of OAuth2 flow
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
PSU-IP-Address	String	M	Represents PSU IP. It should be set only if there is PSU involvement.
Content-Type	String	M	application/json

Request Body:

Attribute	*Type*	*Condition*	*Description*
Access	Account Access	M	Requested access services.
recurringIndicator	Boolean	M	true, if the consent is for recurring access to the account data false, if the consent is for one access to the account data
validUntil	ISODate	M	This parameter is defining a valid until date (including the mentioned date) for the

			requested consent. The content is the local ASPSP date in ISODate Format, e.g. 2017-10-30. Future dates might get adjusted by ASPSP. If a maximal available date is requested, a date in far future is to be used: "9999-12-31". In both cases, the consent object to be retrieved by the GET Consent Request will contain the adjusted date.
frequencyPerDay	Integer	M	This field indicates the requested maximum frequency for an access without PSU involvement per day. For a one-off access, this attribute is set to "1". The frequency needs to be greater equal to one. If not otherwise agreed bilaterally between TPP and ASPSP, the frequency is less equal to 4.
combinedService Indicator	Boolean	M	If true indicates that a payment initiation service will be addressed in the same "session". Only false is supported.

Sample Request:

```
{
  "access": {
    "accounts": [],
    "balances": [],
    "transactions": []
  },
  "recurringIndicator": true,
  "validUntil": "2022-10-10",
  "frequencyPerDay":1,
  "combinedServiceIndicator": true
}
```

Response Body:

Field	*Data Type*	*Description*
consentID	String	The unique identifier for a certain consent
consentStatus	String	Authentication status of the consent

_links	Object	"scaOAuth": The link refers to a JSON document specifying the Oauth details of the ASPSP's 16uthorization server.

Response Header:

Field	*Data Type*	*Condition*	*Description*
Content-Type	String	M	application/json
location	String	M	location of the created resource
X-Request-ID	String	M	Request ID

Sample Response:

```
{
    "consentId": "955040f4-1c0e-43e3-9557-5e41d8959ffa",
    "consentStatus": "received",
    "_links": {
        "scaOAuth": {
            "href": "https://openbankinguat.denizbank.at/oauth/.well-known/oauth-authorization-server"
        }
    }
}
```

DELETE Consent

This method terminates the addressed consent. PSU may request to delete consent via TPP whenever he/she wants. Use DELETE to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/consents/{ConsentId}

Prod Endpoint: https://openbanking.denizbank.at/api/v1/consents/{ConsentId}

Path Parameters

Attribute	*Type*	*Description*
consentId	String	Contains the resource-ID of the consent to be deleted.

Request:

Request parameter: consentID, string, The unique identifier for a certain consent.

Request Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization

Request Body: No request body

Response Body: No response body

Response Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

GET Consent Details

Returns the content of an account information consent object. This is returning the data for the TPP especially in cases, where the consent was directly managed between ASPSP and PSU. Use GET to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/consents/{ConsentId}

Prod Endpoint: https://openbanking.denizbank.at/api/v1/consents/{ConsentId}

Request:

Path Parameters

Attribute	*Type*	*Description*
consentId	String	ID of the corresponding consent object as returned by an Account Information Consent Request

Request Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization

Request Body: No request body

Response Body:

There is "access" field which includes Account Access array information. This array includes "accounts", "balances", "transactions".

Field	*Data Type*	*Description*
Access.Accounts.iban	String	International Bank Account Number

Access.Balances.iban	String	International Bank Account Number
Access.transactions.iban	String	International Bank Account Number
recurringIndicator	Boolean	true-- if the consent is for recurring access to the account data. false, if the consent is for one access to the account data
validUntil	String	This parameter is requesting a valid until date for the requested consent. ISO-Date Format
frequencyPerDay	Integer	This field indicates the requested maximum frequency for an access without PSU involvement per day. For a one-off access, this attribute is set to 1. The frequency needs to be greater equal to one. According to PSD2 rules, TPP can make request without PSU involvement minimum 4 times in 24hr.
lastActionDate	String	Last action date of the consent either through the XS2A interface or the PSU/ASPSP interface.
consentStatus	String	**Received**: PSU started the process of giving consent (the process is not completed) The consent data have been received and are technically correct. The data is not authorized yet **Expired**: Consent is no longer valid due to expire date **Valid**: PSU confirm and complete the process of giving consent. The consent is accepted and valid for GET account data calls and others as specified in the consent object

Response Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
Content-Type	String	M	application/json

If Consent is deleted or not exits, service will return an error message (`"code": "RESOURCE_UNKNOWN"`), `"Consent` not found".

Sample Response: Valid Consent


```
{
    "access": {
        "accounts": [
            {
                "iban": "DE48500307004207019550"
```

```
            }
        ],
        "balances": [
            {
                "iban": "DE48500307004207019550"
            }
        ],
        "transactions": [
            {
                "iban": "DE48500307004207019550"
            }
        ]
    },
    "validUntil": "2023-08-10T00:00:00",
    "recurringIndicator": true,
    "frequencyPerDay": 1,
    "lastActionDate": "2023-07-10T00:41:39",
    "consentStatus": "valid"
}
```


Sample Response: Expired Consent


```
{
    "access": {
        "accounts": [],
        "balances": [],
        "transactions": []
    },
    "recurringIndicator": true,
    "validUntil": "2022-06-30T00:00:00",
    "frequencyPerDay": 1,
    "lastActionDate": "0001-01-01T00:00:00",
    "consentStatus": "expired"
}
```


GET Consent Status

Can check the status of an account information consent resource. Use GET to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/consents/{ConsentId}/status

Prod Endpoint: https://openbanking.denizbank.at/api/v1/consent/{ConsentId}/status

Request:

Path Parameters

Attribute	*Type*	*Description*
consentId	String	The consent identification assigned to the created resource.

Request Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal

Request Body: No request body.

Response Body:

Field	*Data Type*	*Description*
consentStatus		**Received**: PSU started the process of giving consent (the process is not completed) **Expired**: Consent is no longer valid due to expire date **Valid**: PSU confirm and complete the process of giving consent.

Response Header:

Field	*Data Type*	*Cond*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:

```
{
    "consentStatus": "valid"
}
```

Get Authorization Server MetaData

Defines authorization server metadata that an OAuth 2.0 client can use to obtain the information needed to interact with an OAuth 2.0 authorization server, including its endpoint locations. Use GET to access this method.

Test Endpoint:

https://openbankinguat.denizbank.at/oauth/.well-known/oauth-authorization-server

Prod Endpoint:

https://openbanking.denizbank.at/oauth/.well-known/oauth-authorization-server

Request:

Request Parameter:

Request Header: No request header.

Request Body: No request body.

Response:

Response Body:

Field	*Data Type*	*Description*
issuer	String	The authorization server's issuer identifier, which is a URL that uses the "https" scheme and has no query or fragment components.
authorization_endpoint	String	URL of the authorization server's token endpoint
token_endpoint	String	URL of the authorization server's token endpoint
response_types_supported	Array of String	JSON array containing a list of the OAuth 2.0 "response_type" values that this authorization server supports

Response Header:

Field	*Data Type*	*Cond*	*Description*
Content-Type	String	M	application/json

Sample Response:

```
{
    "issuer": "https://openbankinguat.denizbank.at",
    "authorization_endpoint": "https://openbankinguat.denizbank.at/authorize",
    "token_endpoint": "https://openbankinguat.denizbank.at/token",
    "response_types_supported": [
        "code",
        "code token"
    ]
}
```

Get Authorization Request

While routing to the Authorization endpoint, the following query parameters in the scope of OAuth2 are added. We expect query parameter values as encoded in the Authorization endpoint.(For example "AIS%3a" instead of "AIS:") Use GET to access this method.

Test Endpoint:

https://openbankinguat.denizbank.at/authorize?response_type=code&client_id={client_id}&scope={scope}&state={state}&redirect_uri={redirect_uri}&code_challenge_method={code_challenge_method}&code_challenge={code_challenge}

Prod Endpoint:

https://openbanking.denizbank.at/authorize?response_type=code&client_id={client_id}&scope={scope}&state={state}&redirect_uri={redirect_uri}&code_challenge_method={code_challenge_method}&code_challenge={code_challenge}

This endpoint is directed to the Internet banking login page. After the customer logs in and approves on the consent page, within the scope of OAuth2, the code, state response query parameters are directed to the redirect uri of the customer as shown below..

Sample URL: (The part right after "code=" changes according to the request)

https://www.TPPRedirectURL.com/?code=ZR7c1zdRrxww1YAzK0x6D8lVNHxOXI1HDMTl4FqUhn1ruK9mwvZI4mqQjDvv8UD1yhwbK4ra8sH0nB4d32g_17nCf4tg4jWXwiynHYWRwxSpYSID_2Dv8jkieUbUa1M8pdqvdu_bNmfFFTFj7-4UmllGKZNG3nYCJq7K5Vnsc___yM5SacVUQb9n0Lm9sBvOl0;YKVLy6Aa-QWoVWr0wLHacA2

Request:

Query Parameter:

Field	*Data Type*	*Description*
client_id	String	Client ID value which you can get on your API Portal profile page by creating application
response_type	String	"code" is required as response type.
scope	String	PIS: The scope is the reference to the payment resource in the Form "PIS:<paymentId>". AIS: The scope is the reference to the consent resource for account access in the Form "AIS:<consentId>"
state	String	A dynamical value set by the TPP and used to prevent XSRF attacks.
redirect_uri	String	the exact uri of the TPP where the OAuth2 server redirected the user agent to for this particular transaction
code_challenge_method	String	Code verifier transformation method, is "S256" or "plain". "S256" is recommended by this specification.
code_challenge	String	PKCE challenge according to cryptographic RFC 7636 used to prevent code injection attacks.

Request Header: No request header.

Request Body: No request body.

Response:

Response Parameter:

Field	*Data Type*	*Description*
Location:	String	Redirect URI of TPP
code	String	Authorization code
state	String	Same value as for the request.

Response Header:

Field	*Data Type*	*Cond*	*Description*
Content-Type	String	M	text/html

POST Access Token

To use services that require authorization, an access token must be obtained. The duration of the access token is 1 hour for consent requests and 1 day for payment requests. (Durations are parametric and can be updated by Business Units). Once the access token expires it can be renewed via the refresh token method that explained below. Use POST to access this method.

https://www.TPPRedirectURL.com/?code=ZR7c1zdRrxww1YAzK0x6D8lVNHxOXI1HDMTl4FqUhn1ruK9mwvZI4mqQjDvv8UD1yhwbK4ra8sH0nB4d32g_17nCf4tg4jWXwiynHYWRwxSpYSID_2Dv8jkieUbUa1M8pdqvdu_bNmfFFTFj7-4UmlIGKZNG3nYCJq7K5Vnsc___yM5SacVUQb9n0Lm9sBvOl0;YKVLy6Aa-QWoVWr0wLHacA2 => After PSU confirm the access grant for TPP, browser will be redirect to TPP's page. Highlighted part of URL is the code for creating access token. The URL to be used in the GetAccess token method is generated with OAuth2 in the Create Consent step. (Redirect_uri field).

Test Endpoint:

https://openbankinguat.denizbank.at/token?client_id={client_id}&grant_type=authorization_code&code={code}&code_verifier={code_verifier}&redirect_uri={redirect_uri}

Prod Endpoint:

https://openbanking.denizbank.at/token?client_id={client_id}&grant_type=authorization_code&code={code}&code_verifier={code_verifier}&redirect_uri={redirect_uri}

Request:

Query Parameter:

Field	*Data Type*	*Description*
client_id	String	Client ID value which you can get on your API Portal profile page by creating application
grant_type	String	"authorization_code" is required as response type.
code	String	Authorization code from the authorization response

code_verifier	String	PKCE verifier according to cryptographic RFC 7636 (https://tools.ietf.org/html/rfc7636) used to prevent code injection attacks
redirect_uri	String	the exact uri of the TPP where the OAuth2 server redirected the user agent to for this particular transaction

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal

Request Body: No request body.

Response:

Response Body:

Field	*Data Type*	*Description*
access_token	String	Access Token bound to the scope as requested in the authorisation request and confirmed by the PSU. This info is used all AIS/PIS requests except Payment Initiation.
token_type	String	Token Type:Bearer
expires_in	String	The lifetime of the access token in seconds
scope	String	Consent Id or Payment Id
refresh_token	String	Refresh Token Information. Used to obtain a new access token when the access token expires.

Response Header:

Field	*Data Type*	*Cond*	*Description*
Content-Type	String	M	application/json

Sample Response:


```
{
    "access_token": "78yYPYSmQCHHv754eC0IMmAwWBJPNy-i4eInKZ-
NSNNkpaMUYBMfC3bYgIlEjNBP4QoTZ83gWSnQjyWuIEBweqleUIOdOs17WzNt1w-
xRIEma7RD8e46V1_Kk_e2NeNHKnFDqStxjvwwWzQQqOETv4w27hhK9EQKolrWPL5MzCxFSSjtRx_y4sB3_Qv2-
AF20;71nRiOFv1ohvEfuLZJDgJA2",
    "token_type": "Bearer",
    "expires_in": 2633822,
    "scope": "a09ae616-a928-485c-958c-b49fb6351308",
    "refresh_token": "hqXkQGbRm1XaN-DJKD0ljdScMIfE9v1L6s3PfdvA8wzvy_zzbXgqXeFQ-
UiQRcCQOz0av6G0SBV5nQ2Ku1o7dvHGMP_KVy0ZwUtFfAf4b8vWIBwOkeNnnhFXgkGP5Pbh0;96y4799Jlc9Uk
Q5hoBL4Jg2"
}
```

POST Access Token via Refresh Token

It ensures the renewal of the expired access token. The renewal token is valid until the "validUntil" date specified by the customer in consent requests. The validity period for payment requests is 50 days. (This period is parametric and can be changed by business units.) Access tokens cannot be obtained again for deleted consent and payment requests. If the refresh token has expired, service will be failed.

Test Endpoint:

https://openbankinguat.denizbank.at/token?client_id={client_id}&grant_type=refresh_token&refresh_token={refresh_token}&redirect_uri={redirect_uri}

Prod Endpoint:

https://openbanking.denizbank.at/token?client_id={client_id}&grant_type=refresh_token&refresh_token={refresh_token}&redirect_uri={redirect_uri}

Request:

Query Parameter:

Field	*Data Type*	*Description*
client_id	String	Client ID value which you can get on your API Portal profile page by creating application
grant_type	String	"refresh_token" is required as response type.
refresh_token	String	Refresh token from the access token endpoint response.
redirect_uri	String	the exact uri of the TPP where the OAuth2 server redirected the user agent to for this particular transaction

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal

Request Body: No request body.

Response:

Response Body:

Field	*Data Type*	*Description*
access_token	String	Renewed Access Token
token_type	String	Token Type:Bearer
expires_in	String	The lifetime of the renewed access token in seconds
scope	String	Consent Id or Payment Id
refresh_token	String	Refresh Token Information. Used to obtain a new access token when the access token expires.

Response Header:

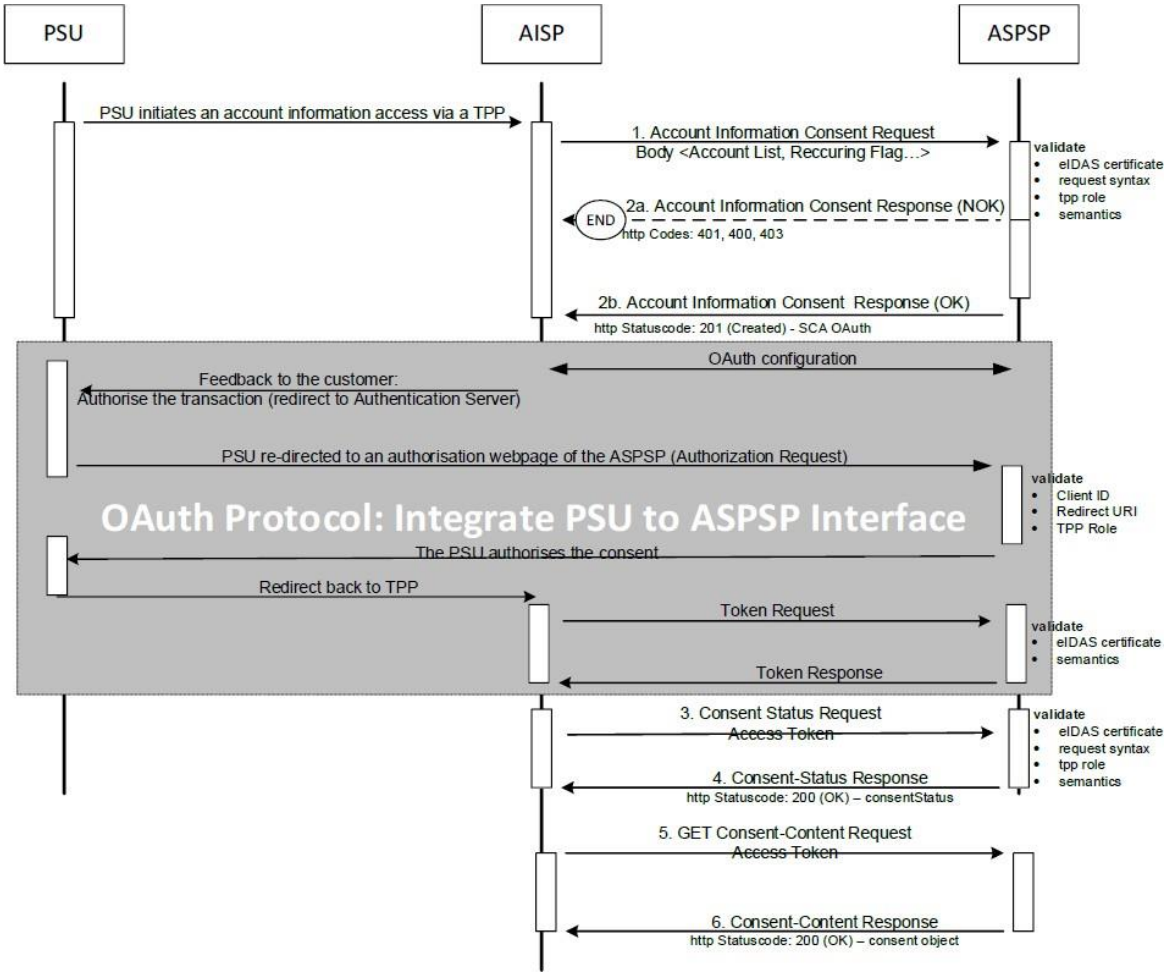
Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json

Sample Response:


```
{
  "access_token": "78yYPYSmQCHHv754eC0IMmAwWBJPNy-i4eInKZ-
NSNNkpaMUYBMfC3bYgIlEjNBP4QoTZ83gWSnQjyWuIEBweqleUIOdOs17WzNt1w-
xRIEma7RD8e46V1_Kk_e2NeNHKnFDqStxjvwwWzQQqOETv4w27hhK9EQKolrWPL5MzCxFSSjtRx_y4sB3_Qv2-
AF20;71nRiOFv1ohvEfuLZJDgJA2",
  "token_type": "Bearer",
  "expires_in": 2633822,
  "scope": "a09ae616-a928-485c-958c-b49fb6351308",
 "refresh_token": "x_gD1DeI53gLuqdqz1F9auPOWjXZZDyHGBgDyUdjoer7_yieLH-
dUxtJ3QgfH0eLXTmTw_xqLgyj41BPzXy1gA-
Ki45gavx_ICWFlycnPtgbCYuTaNvSo_Id3gGwlB2W0;qscuy5ZPbC7MuqQAHe5yNw2"

}
```

Account Information Services

Below schema works for all AIS methods. There is integration with Konsentus to validate QWAC certificate. There is request limit for all AIS methods four per day.



AIS Flow Sequence Diagram

GET Account List

Reads a list of bank accounts, with balances where required. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system. Use GET to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/accounts

Prod Endpoint: https://openbanking.denizbank.at/api/v1/accounts

Request:

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Consent-ID	String	M	Represents PSU's consent
PSU-IP-Adress	String	C	IP Address field between PSU and TPP
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization

Request Body: No request body.

Response Body:

Field	*Data Type*	*Description*
Accounts.resourceId	String	This is the data element to be used in the path when retrieving data from a dedicated account.
Accounts.iban	String	International Bank Account Nu mber
Accounts.currency	String	ISO 4217 Alpha 3 currency code
Accounts.name	String	Name of the account given by the bank or the PSU in online-banking
Accounts.product	String	Product name of the bank for this account, proprietary definition.

Response Header:

Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:


```
{
    "accounts": [
        {
            "resourceId": "a1pDNXlubnpCVEtjV3RmblNxNTB1UT09",
            "iban": "AT161965004010251550",
            "currency": "EUR",
            "name": "TEST",
            "product": "MGBA"
        }
    ]
}
```

GET Account Detail

Reads details about an account, with balances where required. Use GET to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/accounts/{accountId}

Prod Endpoint: https://openbanking.denizbank.at/api/v1/accounts/{accountId}

Request:

Path Parameters

Attribute	*Type*	*Description*
account-id	String	This identification is denoting the addressed account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Request Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Consent-ID	String	M	Represents PSU's consent
PSU-IP-Adress	String	C	IP Address field between PSU and TPP
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization

Request Body: No request body.

Response Body:

Field	*Data Type*	*Description*

Account.resourceId	String	This is the data element to be used in the path when retrieving data from a dedicated account.
Account.iban	String	International Bank Account Number
Account.currency	String	ISO 4217 Alpha 3 currency code
Account.name	String	Name of the account given by the bank or the PSU in online-banking
Account.product	String	Product name of the bank for this account, proprietary definition.

Response Header:

Field	*Data Type*	*Cond*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:


```
{
    "account": {
        "resourceId": "a1pDNXlubnpCVEtjV3RmblNxNTB1UT09",
        "iban": "AT161965004010251550",
        "currency": "EUR",
        "name": "TEST",
        "product": "MGBA"
    }
}
```


GET Account Balances

Reads account data from a given account addressed by "account-id". Use GET to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/accounts/{accountId}/balances

Prod Endpoint: https://openbanking.denizbank.at/api/v1/accounts/{accountId}/balances

Request:

Path Parameters

Attribute	*Type*	*Description*
account-id	String	This identification is denoting the addressed account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party.
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal.
Consent-ID	String	M	Represents PSU's consent.
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization
PSU-IP-Address	String	C	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.

Request Body: No request body.

Response:

Field	*Data Type*	*Description*
account.iban	String	International Bank Account Number
account.currency	String	ISO 4217 Alpha 3 currency code
balances.balanceAmount.currency	String	ISO 4217 Alpha 3 currency code

balances.balanceAmount.amount	Number	The amount given with fractional digits, where fractions must be compliant to the currency definition. The decimal separator is a dot.
balances.balanceType	Balance Type	We provide "interimBooked" balance type.
balances.referenceDate	String	"2022-08-21T21:38:50.6309513+03:00" Reference date of the balance

Response Header:

Field	*Data Type*	*Cond*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:


```
{
    "account": {
        "iban": "AT161965004010251550",
        "currency": "EUR"
    },
    "balances": [
        {
            "balanceAmount": {
                "currency": "EUR",
                "amount": 52241.86
            },
            "balanceType": "interimBooked",
            "referenceDate": "2023-07-10T12:46:43.3302097+02:00"
        }
    ]
}
```


GET Transaction List

Reads account transaction data from a given account addressed by "account-id". Use GET to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/accounts/{accountId}/transactions

Prod Endpoint: https://openbanking.denizbank.at/api/v1/accounts/{accountId}/transactions

Request:

Request parameters:

Path Parameters

Attribute	*Type*	*Description*
account-id	String	This identification is denoting the addressed account. The account-id is retrieved by using a "Read Account List" call. The account-id is the "resourceId" attribute of the account structure. Its value is constant at least throughout the lifecycle of a given consent.

Query Parameters

Attribute	*Type*	*Description*
dateFrom	Format date-time(as date-time in RFC3339)	Starting date (inclusive the date dateFrom) of the transaction list,
dateTo	Format date-time(as date-time in RFC3339)	End date (inclusive the data dateTo) of the transaction list
bookingStatus	String	Permitted code is "booked".

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Consent-ID	String	M	Represents PSU's consent
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization
PSU-IP-Address	String	C	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. It shall be contained if and only if this request was actively initiated by the PSU.

Request Body: No request body.

Response:

Field	*Data Type*	*Description*
Account	Account Reference	Identifier of the addressed account.
Transactions	Account Report	JSON based account report. This account report contains transactions resulting from the query parameters.

Response Header:

Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

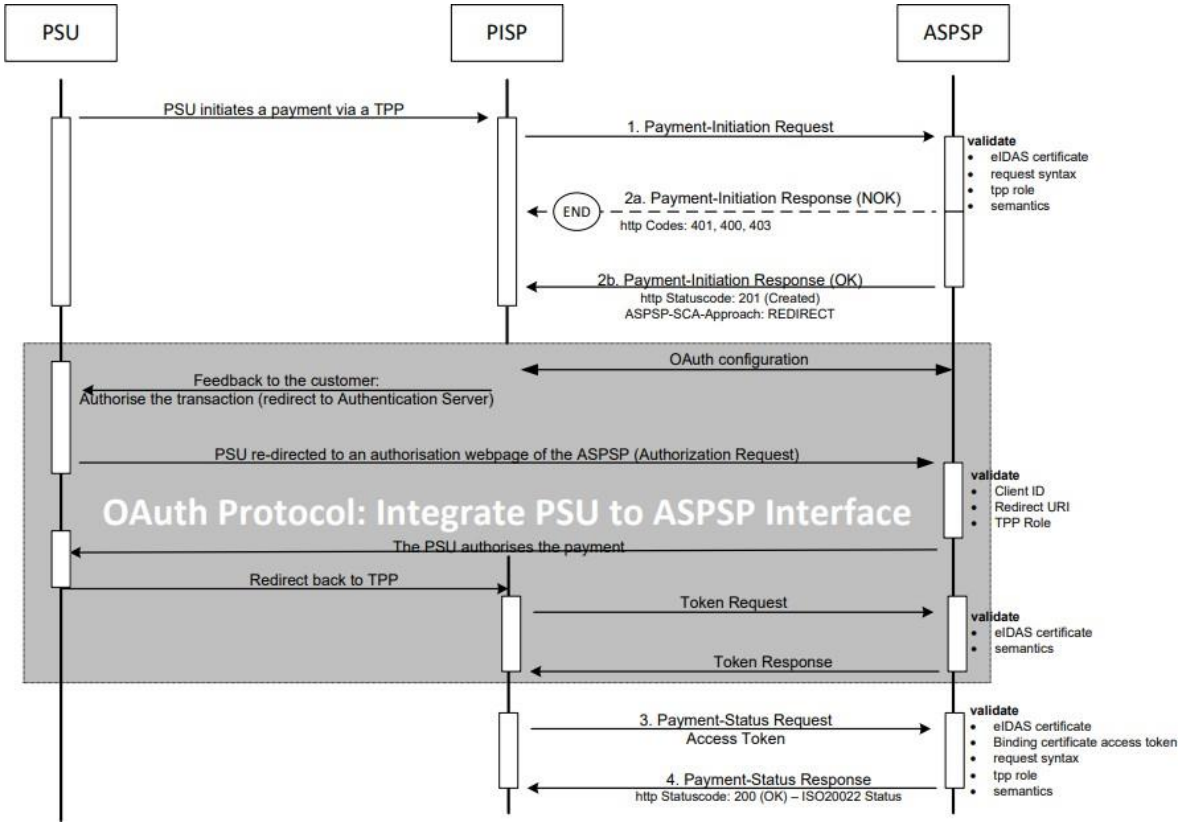
Sample Response:


```
{
    "account": {
        "iban": "AT161965004010251550",
        "currency": "EUR"
    },
    "transactions": {
        "booked": [
            {
                "transactionId": "477",
                "channel": "Other",
                "bookingDate": "2023-07-10T23:16:04",
                "transactionAmount": {
                    "currency": "EUR",
                    "amount": -10
                },
                "remittanceInformationUnstructured": "www.amazon.de",
                "balanceAfterTransaction": {
                    "currency": "EUR",
                    "amount": 1000
                }

            },
            {
                "transactionId": "538",
                "channel": "Other",
                "bookingDate": "2023-07-11T01:45:16",
                "transactionAmount": {
                    "currency": "EUR",
                    "amount": -8
                },
                "remittanceInformationUnstructured": "www.amazon.de",
                "balanceAfterTransaction": {
                    "currency": "EUR",
                    "amount": 1000
                }

            }
        ],
        "_links": {
            "account": {
                "href": "https://openbankinguat.denizbank.at/api/v1/accounts/a1pDNXlub
npCVEtjV3RmblNxNTB1UT09"
            }
        }
    }
}
```

Payment Initiation Services

The Payment Initiation Request is followed by implicit start of the Authorization Process. This is followed by a redirection to the ASPSP SCA authorization site. A status request might be requested after initiation.

Below schema works for all PIS methods. There is integration with Konsentus to validate QWAC certificate.



PIS Flow Sequence Diagram

Payment Initiation

Creates a payment initiation request. Use POST to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/payments/{paymentProduct}

Prod Endpoint: https://openbanking.denizbank.at/api/v1/payments/{paymentProduct}

Request:

Path Parameters

Attribute	*Type*	*Description*
payment-product	String	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The following payment products are supported: sepa-credit-transfers, cross-border-credit-transfers

Request Header:

Field	***Data Type***	***Cond.***	*Path Parameters*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained.It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
TPP-Redirect-URI	String	M	Represents URI link that TPP wants to register for redirection after successful completion of OAuth2 flow
PSU-IP-Address	String	M	Represents PSU IP. It should be set only if there is PSU involvement.
Content-Type	String	M	application/json

Request Body:

Field	*Data Type*	*Cond.*	*Description*
DebtorAccount.iban	String	M	International Bank Account Number
DebtorAccount.currency	String	O	ISO 4217 Alpha 3 currency code
instructedAmount.amount	Decimal	M	Payment amount
instructedAmount.currency	String	M	ISO 4217 Alpha 3 currency code
creditorAccount.iban	String	M	International Bank Account Number
creditorAccount.currency	String	O	ISO 4217 Alpha 3 currency code
creditorName	String	M	Name of the creditor if a "Debited" transaction
creditorAdress	String	M	Address of Bank
remittanceInformationUnstructured	String	M	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts receivable system
endToEndIdentification	string	O	Customer reference field in online banking.
requestedExecutionDate	DateTime	O	If contained, payment transaction will executed in requested execution date.

Sample Request:


```
{
  "debtorAccount": {
    "iban": "AT161965004010251550",
    "currency": "EUR"
  },
  "instructedAmount": {
    "amount": 8,
    "currency": "EUR"
  },
  "creditorAccount": {
    "iban": "DE02512207000906409427",
    "currency": "EUR"
  },
  "creditorName": "Jean",
  "creditorAddress": {
    "streetname": "street",
    "townname": "city",
    "postcode": 15150,
    "country": "BE"
  },
  "remittanceInformationUnstructured": www.amazon.de,
  "endToEndIdentification": "Test",
  "requestedExecutionDate": "2024-01-01"
}
```


Response Body:

Field	*Data Type*	*Description*
paymentId	String	Identification of the generated payment initiation.
transactionStatus	String	The transaction status is filled with codes of the ISO 20022 data table
_links.scaOAuth.href	String	The link refers to a JSON document specifying the OAuth details of the ASPSP's authorisation server

Response Header:

Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json
location	String	M	location of the created resource
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:


```
{
    "paymentId": "d402b203-9ad9-4fa3-8f00-f78fce7e74fc",
    "transactionStatus": "RCVD",
    "_links": {
        "scaOAuth": {
            "href": "https://openbankinguat.denizbank.at/oauth/.well-known/oauth-authorization-server"
        }
    }
}
```


Periodic Payment Initiation

Creates a periodic payment initiation request. Use POST to access this method.

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/periodic-payments/{paymentProduct}

Prod Endpoint: https://openbanking.denizbank.at/api/v1/periodic-payments/{paymentProduct}

Request:

Path Parameters

Attribute	*Type*	*Description*
payment-product	String	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The following payment product is supported: sepa-credit-transfers

Request Header:

Field	***Data Type***	***Cond.***	*Path Parameters*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained.It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
TPP-Redirect-URI	String	M	Represents URI link that TPP wants to register for redirection after successful completion of OAuth2 flow
PSU-IP-Address	String	M	Represents PSU IP. It should be set only if there is PSU involvement.
Content-Type	String	M	application/json

Request Body:

Field	*Data Type*	*Cond.*	*Description*
DebtorAccount.iban	String	M	International Bank Account Number
DebtorAccount.currency	String	O	ISO 4217 Alpha 3 currency code
instructedAmount.amount	Decimal	M	Payment amount
instructedAmount.currency	String	M	ISO 4217 Alpha 3 currency code
creditorAccount.iban	String	M	International Bank Account Number
creditorAccount.currency	String	O	ISO 4217 Alpha 3 currency code
creditorName	String	M	Name of the creditor if a "Debited" transaction
creditorAdress	String	M	Address of Bank
remittanceInformationUnstructured	String	M	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts receivable system
endToEndIdentification	string	O	Customer reference field in online banking.
frequency	String	M	Frequency of periodic payment
startDate	DateTime	M	The first applicable day of execution
endDate	DateTime	O	Last applicable day of execution
dayOfExecution	String	M	Day of execution based on frequency. For example: - "Monthly" frequency: "01" means first day of month. "Weekly" frequency: "01" means Monday.
executionRule	String	O	Indicates whether the regular payment will be set following or preceding if it falls on a holiday or weekend. Only following is supported.

Sample Request:


```
{
  "debtorAccount": {
    "iban": "AT161965004010251550",
    "currency": "EUR"
  },
  "instructedAmount": {
    "amount": 8,
    "currency": "EUR"
  },
  "creditorAccount": {
    "iban": "DE02512207000906409427",
    "currency": "EUR"
  },
  "creditorName": "Jean",
  "creditorAddress": {
    "streetname": "street",
    "townname": "city",
    "postcode": 15150,
    "country": "BE"
  },
  "remittanceInformationUnstructured": www.amazon.de,
  "endToEndIdentification": "Test",
  "startDate": "2024-01-01",
  "endDate": "2024-03-01",
  "frequency": "Monthly",
  "dayOfExecution": "02",
  "executionRule": "following"
}
```


Response:

Field	*Data Type*	*Description*
paymentId	String	Identification of the generated payment initiation.
transactionStatus	String	The transaction status is filled with codes of the ISO 20022 data table
_links.scaOAuth.href	String	The link refers to a JSON document specifying the OAuth details of the ASPSP's authorisation server

Response Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
location	String	M	location of the created resource

Sample Response:


```
{
    "paymentId": "d402b203-9ad9-4fa3-8f00-f78fce7e74fc",
    "transactionStatus": "RCVD",
    "_links": {
        "scaOAuth": {
            "href": "https://openbankinguat.denizbank.at/oauth/.well-known/oauth-authorization-server"
        }
    }
}
```


GET Payment Detail

Returns the content of a payment object. Use GET to access this method.

Test Endpoint:

https://openbankinguat.denizbank.at/api/v1/{payment-service}/{paymentProduct}/{paymentId}

Prod Endpoint:

https://openbanking.denizbank.at/api/v1/{payment-service}/{paymentProduct}/{paymentId}

Request:

Path Parameters

Attribute	*Type*	*Description*
payment-service	String	The possible values are "payments" and "periodic-payments"
payment-product	String	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The following payment products are supported: sepa-credit-transfers, cross-border-credit-transfers For periodic "payments sepa-credit-transfers" is supported as in online banking.
paymentId	String	Resource identification of the generated payment initiation resource.

Request Header:

Field	***Data Type***	***Cond.***	***Description***
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal

Request Body; No request body.

Response:

Field	*Data Type*	*Description*
creditorAccount.iban	String	International Bank Account Number
creditorAccount.currency	String	ISO 4217 Alpha 3 currency code
creditorAddress.streetName	String	Country code of Bank
creditorAddress.buildingNumber	String	Building number of Bank
creditorAddress.townName	String	Town name of Bank
creditorAddress.postCode	String	Post code of Bank
creditorAddress.country	String	Country code of Bank
creditorName	String	Party to which an amount of money is due. Name by which a party is known and which is usually used to identify that party
debtorAccount.iban	String	International Bank Account Number
debtorAccount.currency	String	ISO 4217 Alpha 3 currency code
instructedAmount.currency	String	ISO 4217 Alpha 3 currency code
instructedAmount.amount	Number	The amount given with fractional digits, where fractions must be compliant to the currency definition. The decimal separator is a dot.
remittanceInformationUnstructured	String	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts receivable system
transactionStatus	String	The transaction status is filled with codes of the ISO 20022 data table
frequency	String	Frequency of periodic payment

startDate	DateTime	The first applicable day of execution
endDate	DateTime	Last applicable day of execution
dayOfExecution	String	Day of execution based on frequency. For example: - "Monthly" frequency: "01" means first day of month. - "Weekly" frequency: "01" means Monday.
executionRule	String	Indicates whether the regular payment will be set following or preceding if it falls on a holiday or weekend. Only following is supported.
requestedExecutionDate	DateTime	If contained, payment transaction will executed in requested execution date.
endToEndIdentification	String	Customer reference field in online banking.

Response Header:

Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:


```
{
    "creditorAccount": {
        "iban": "DE02512207000906409427",
        "currency": "EUR"
    },
    "creditorAddress": {
        "streetName": "street",
        "buildingNumber": "",
        "townName": "city",
        "postCode": "15150",
        "country": "BE"
    },
    "creditorName": "Jean",
    "debtorAccount": {
        "iban": "AT161965004010251550",
        "currency": "EUR"
    },
    "instructedAmount": {
        "currency": "EUR",
        "amount": 10
    },
    "remittanceInformationUnstructured": "www.amazon.de",
    "transactionStatus": "ACSC",
    "startDate": "2024-01-01 ",
    "endDate": "2024-03-01",
    "frequency": "monthly",
    "dayOfExecution": "02",
    "executionRule": "following",
    "requestedExecutionDate": "0001-01-01",
    "endToEndIdentification": "Test"
}
```


GET Payment Status

Can check the status of a payment initiation. Use GET to access this method.

Test Endpoint:

https://openbankinguat.denizbank.at/api/v1/payments/{paymentProduct}/{paymentId}/status

Prod Endpoint:

https://openbanking.denizbank.at/api/v1/payments/{paymentProduct}/{paymentId}/status

Request:

Path Parameters

Attribute	*Type*	*Description*
payment-service	String	The possible values are "payments" and "periodic-payments"
payment-product	String	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT).The following payment products are supported: sepa-credit-transfers,cross-border-credit-transfers For periodic "payments sepa-credit-transfers" is supported as in online banking.
paymentId	String	Resource Identification of the related payment.

Request Parameters:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained.It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization

Request Body : No request body.

Response:

Field	*Data Type*	*Description*
transactionStatus	String	The transaction status is filled with codes of the ISO 20022 data table

Response Header:

Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response

```
{
    "transactionStatus": "ACSC"
}
```

DELETE Payment

This method terminates the addressed future and periodic payments. Instant payments cannot be deleted. PSU may request to delete future and periodic payment via TPP whenever he/she wants. Use DELETE to access this method.

Test Endpoint:

https://openbankinguat.denizbank.at/api/v1/{paymentService}/{paymentProduct}/{paymentId}

Prod Endpoint:

https://openbanking.denizbank.at/api/v1/{paymentService}/{paymentProduct}/{paymentId}

Request:

Path Parameters

Attribute	*Type*	*Description*
paymentService	String	The possible values are "payments" and "periodic-payments"
paymentProduct	String	The addressed payment product. The following payment products are supported: sepa-credit-transfers,cross-border-credit-transfers For periodic "payments sepa-credit-transfers" is supported as in online banking.
paymentId	String	Resource Identification of the related payment.

Request Header:

Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization

Request Body: No request body

Response:

Response Header:

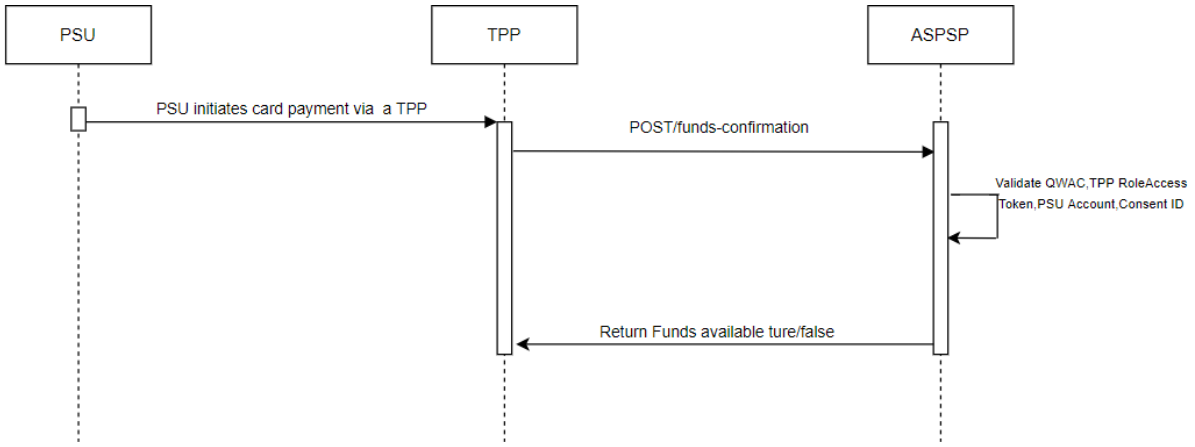
Field	*Data Type*	*Cond*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Response Body: No response body. Returns Http 204 status code when successful.

Confirmation of Funds

Creates a confirmation of funds request at the ASPSP. Use POST to access this method.

Below schema works for confirmations of funds. There is integration with Konsentus to validate QWAC certificate.



COF Flow Sequence Diagram

Test Endpoint: https://openbankinguat.denizbank.at/api/v1/funds-confirmations

Prod Endpoint: https://openbanking.denizbank.at/api/v1/funds-confirmations

Request:

Request Parameter: -

Request Header:

Field	*Data Type*	*Cond.*	*Description*
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party
TPP-Application-Code	String	M	Represents TPP's application that is provided from API portal
Authorization	String	M	AccessToken which is generated in getAccessToken method is required for authorization
Consent-ID	String	M	Represents PSU's consent

TPP-Signature-Certificate	String	M	The certificate used for signing the request, in base64 encoding. Must be contained if a signature is contained. It must be QWAC and contains role information of TPP.
Content-Type	String	M	application/json

Request Body:

Field	*Data Type*	*Cond.*	*Description*
account	String	M	PSU's account number
instructedAmount	Amount	M	Transaction amount to be checked within the funds check mechanism.
cardNumber	String	O	Card Number of the card issued by the PIISP.

Response:

Field	*Data Type*	*Description*
fundsAvailable	Boolean	Equals true if sufficient funds are available at the time of the request, false otherwise.

Response Header:

Field	*Data Type*	*Cond.*	*Description*
Content-Type	String	M	application/json
X-Request-ID	String	M	ID of the request, unique to the call, as determined by the initiating party

Sample Response:

```
{
    "fundsAvailable ": true
}
```

Response Codes

Http Status Code	*Description*
200	Success for Get Methods
201	Success for Post Methods

204	Success for Delete Methods
400	Bad Request - The request was invalid (There might be a missing header filed in the request)
401	Unauthorized- If the consent is not valid or The TPP's QWAC certificate is invalid
404	If there is no consent/account http status code return 404
500	Internal Server Error

Error Codes

Error Code	***Description***
FORMAT_ERROR	The request is not suitable.
RESOURCE_UNKNOWN	Success for Post Methods
PERIOD_INVALID	If startdate is bigger than enddate for GetAccountTransactions method
MISSING_HEADER_FIELD	Required value is missing in header field.
USED_AUTHENTICATION_CODE	Authentication code is already used previously
REQUEST_LIMIT_EXCEEDED	When the TPP trigger methods more than the number of allowed frequencies per day.
REQUIRED_METHOD_PARAMETERS	When the parameters are empty for Get Token method
INVALID_VALIDUNTIL	When validuntil value is older than current day
INVALID_URL	When URL does not belong to application for create consent and create payment methods.
PSU_CREDENTIALS_INVALID	If the account is not match for AIS methods
CONSENT_INVALID	If there is no consent or consent has not been approved yet for account methods
CONSENT_EXPIRED	When the expire date of consent passed.
PAYMENT_INVALID	If there is no payment
TOKEN_INVALID	When token could not verify
TOKEN_EXPIRED	When expire date of token passed
MISSING_CLIENT	If client ID or client Secret do not exist
INVALID_AUTH_CODE	When the code value could not be authenticated
INVALID_APPLICATIONCODE	If the application code does not exist

CERTIFICATE_NOT_VALID	QWAC of TPP is invalid
UNAUTHORIZED_TPP_ROLE	The role information of TPP is not suitable
FREQUENCY_INVALID	If the frequency field is other than weekly and monthly
DAYOFEXECUTION_INVALID	If sent other than numbers 1 to 7 for weekly frequency or If sent other than numbers 1 to 31 for monthly frequency
PERIODIC_STARTDATE_INVALID	If the start date is past or not sent
PERIODIC_ENDDATE_INVALID	If the end date is submitted and is less today than or less than the start date
EXECUTIONRULE_INVALID	When execution rule sent outside "following"
MONTHOFEXECUTION_NOT_SUPPORTED	When month of execution is sent
REQUESTEDEXECUTIONDATE_INVALID	If the requested execution date is submitted and is less today than today
SERVICENAME_INVALID	If payments are made to the { payment -service} variable as payments for a periodic payment or {payment-service} variable periodic-payments for a future or one-time payment