

Quick Access Guide

How to get access to the Santander Open Banking API Sandbox and the productive environment?

Index:

1.	List of URLs and endpoints	2
1	Technical baseline.....	4
1.1	Certificate in Santander Sandbox.....	4
1.2	Registration of TPP-Redirect-URIs.....	4
1.3	SCA Approaches.....	5
1.3.1	DECOUPLED.....	6
1.3.2	EMBEDDED.....	6
1.3.3	REDIRECT.....	7
1.3.4	scaOAuth details (/ .well-known/)	8
1.3.5	SCA Status	9
2	Functional details	9
2.1	Consents.....	10
2.1.1	Global consents	10
2.1.2	Dedicated consents with IBAN	10
2.1.3	Online / Offline consents and recurring indicator	10
2.1.4	Frequency per day and combined services	11
2.2	Accounts	11
2.2.1	Transactions	11
2.2.2	Transactions cursor.....	11

2.2.3	Additional information.....	11
2.3	Payments.....	11
2.3.1	Payment services.....	11
2.3.2	Payment service JSON.....	11
2.3.3	Payment service XML.....	12
2.3.4	Payment status codes.....	12
2.3.5	SEPA Instant Payments.....	13
2.4	Periodic Payments (Standing Order).....	13
2.4.1	Frequency Codes.....	14
2.4.2	Execution rule and day of execution.....	14
2.4.3	Suspension of Periodic Payments.....	15
2.4.4	Get all Periodic Payments.....	15
2.4.5	Update or delete Periodic Payments.....	15
2.5	Overview of endpoints and details.....	16

1. List of URLs and endpoints

Production URL:

<https://api-cc.santander.de/scb-openapis/client>

Sandbox URL:

<https://api-sandbox-cc.santander.de/scb-openapis/sx>

Endpoint:	Method:	Resource:
Redirect-URI registration	POST	/v1/tpp_registrations/mutual_tls
Access Token	POST	/v1/oauth_matls/token
scaOAuth (Discovery Service)	GET	(link to a JSON document specifying the OAuth details) /.well-known/oauth-authorization-server
PUT Authorizations	PUT	/v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId} or /v1/consents/{consentId}/authorisations/{authorisationId}

Authorization status	GET	/v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId} or /v1/consents/{consentId}/authorisations/{authorisationId}
Consent	POST	/v1/consents
Consent details	GET	/v1/consents/{consentId}
Consent status	GET	/v1/consents/{consentId}/status
Consent delete	DELETE	/v1/consents/{consentId}
Accounts	GET	/v1/accounts
Account details	GET	/v1/accounts/{resourceId}
Account balances	GET	/v1/accounts/{resourceId}/balances
Account transactions	GET	/v1/accounts/{resourceId}/transactions
Payment	POST	/v1/payments/sepa-credit-transfers
Payment details	GET	/v1/payments/sepa-credit-transfers/{payment_id}
Payment status	GET	/v1/payments/sepa-credit-transfers/{payment_id}/status
Payment delete	DELETE	/v1/payments/sepa-credit-transfers/{payment_id}
Payment (pain format)	POST	/v1/payments/pain.001-sepa-credit-transfers
Payment details (pain format)	GET	/v1/payments/pain.001-sepa-credit-transfers/{payment_id}
Payment status (pain format)	GET	/v1/payments/pain.001-sepa-credit-transfers/{payment_id}/status
Payment delete (pain format)	DELETE	/v1/payments/pain.001-sepa-credit-transfers/{payment_id}
Periodic payment	POST	/v1/periodic-payments/sepa-credit-transfers

Periodic payment details	GET	/v1/periodic-payments/sepa-credit-transfers/{payment_id}
Periodic payment status	GET	/v1/periodic-payments/sepa-credit-transfers/{payment_id}/status
Periodic payment delete	DELETE	/v1/periodic-payments/sepa-credit-transfers/{payment_id}
Funds confirmation	POST	/v1/funds-confirmations

1 Technical baseline

1.1 Certificate in Santander Sandbox

For any kind of communication with our API you need to establish a **TLS** connection with an **eIDAS certificate**.

The only supported certificate type is **QWAC**, for production and sandbox environment.

To access the productive API, the eIDAS certificate has to be your individual certificate provided by an authorized trust centre.



The access to the Santander Sandbox is only possible using the Santander Sandbox eIDAS certificate, provided in the Santander API Market. (<https://www.santander.de/privatkunden/specials/api-market/>)
The Client-Id equivalent for this certificate is: PSDDE-SANDBOX-0006

SCA method in conjunction with OAuth means that any call to our API has to be carried out with an access token (please read also BerlinGroup NextGenPSD2 XS2A Framework, Implementation Guidelines version 1.3, chapter 13).

1.2 Registration of TPP-Redirect-URIs

From security reason it is technically required to register your TPP-Redirect-URIs inside a separate endpoint of our API. As long as the registration ID or the used TPP-Redirect-URIs doesn't change, it is not necessary to repeat this process.



Please be aware that the step of registering at least one TPP-Redirect-URI is strictly necessary before calling the first time any token, consent or payment endpoint. Otherwise your request will be answered with an error message.

Of course, it is also possible to make this one time registration request from a standard application like POSTMAN instead of your individual business application.

Example:

```
curl -X POST
--url https://api-cc.santander.de/scb-
openapis/client/v1/tpp_registrations/mutual_tls
-H 'Accept: application/json'
-H 'Content-Type: application/json'
--key '$PATH_TO_KEY_PEM'
--cert '$PATH_TO_CERT_PEM:$PASSWORD_OF_CERT'
-d '{"registeredRedirectUris": ["https://tpp-redirect.com/cb"]}'
```

Example for request body in raw JSON:

```
{
  "registeredRedirectUris": [
    https://tpp-redirect.com/cb,
    https://tpp-redirect2.com/rtd,
    https://tpp-redirect3.com/hgfddcv
  ]
}
```

Please remind that this `registeredRedirectUris` is only an example and must be given with the TPPs individual TPP-Redirect-URIs.

1.3 SCA Approaches

The authorization method is automatically derived from the information given in the POST `/consents/`, `/payments/` or `/periodic-payment/` call.

In case of adding `TPP-Redirect-Preferred = false` plus a valid PSU-ID, the SCA method EMBEDDED or DECOUPLED is automatically chosen. This is always the SCA method known and preferred by the PSU. It is not possible to change this predefined SCA method.

The **PSU-ID** is 6 to 20 digit alphanumeric. Additionally it is possible that the customer has defined his email address as alias and of course for these customers it is also possible to use this alias as PSU-ID in the Open Banking API.

Derivation of SCA method:

Request Header:		Request Body:	Response Header:
PSU-ID	TPP-Redirect-Preferred	Debtor account:*	ASPSP-SCA-Approach
not given or valid or invalid	not given	not given or valid or invalid	REDIRECT
			<i>Implicit Redirect</i>

not given or valid or invalid	true	not given or valid or invalid	REDIRECT	<i>Explicit Redirect</i>
valid	false	not given or invalid	REDIRECT	<i>Implicit Redirect</i>
not given or invalid	false	valid	REDIRECT	<i>Implicit Redirect</i>
valid	false	valid	EMBEDDED or DECOUPLED	<i>Explicit Embedded or Decoupled</i>

* in case of an allPsd2 consent request, no debtor account validation performed

1.3.1 DECOUPLED

When the ASPSP-SCA-Approach is DECOUPLED, the PSU automatically receives a push notification in the Santander Sign APP to authorize the transaction or consent request.



In case of payments initiations, it is strictly recommended to make additionally to a successful authorization status request also a payment status request. Final transaction validations are performed after the authorization and it is also possible that a "transactionStatus" is "RJCT" (rejected) when the "scaStatus" is "finalised".

1.3.2 EMBEDDED

An ASPSP-SCA-Approach = Embedded requires two additional PUT requests for final authorization.

```
PUT .../v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}
```

or

```
PUT .../v1/consents/{consentId}/authorisations/{authorisationId}
```

Request body of first PUT request:

```
{
  "psuData": {
    "password": "PSUsPASSWORD"
  }
}
```

Request body of second PUT request:

```
{
  "scaAuthenticationData": "mTAN* received by the PSU"
}
```

mTAN = 6 digit numeric

1.3.3 REDIRECT

For the REDIRECT SCA approach you need to redirect the PSU to our Santander REDIRECT SCA Portal by calling the URL for the authorization endpoint endpoint which was retrieved in the step before from the discovery service. The following parameters must be added to the URL:

```
response_type=code
client_id=PSD<xx-xx-xxxxxx>
scope=<PIS or AIS>%3A<paymentID or consentID>
state=<state>
redirect_uri=https://your redirect-URL
```

Element	Description
response_type=code	fix value
client_id=PSD<xx-xx-xxxxxx>	for production, TPPs individual Client ID; for Santander Sandbox: PSDDE-SANDBOX-0006
scope=<PIS or AIS>	fix value PIS / AIS depending to the previous payments (PIS) or consent (AIS) call
%3A	fix value
<paymentID or consentID>	paymentID or consentID received from the post payment or post consent call
state=<state>	dynamical value set by the TPP
redirect_uri=<https://your redirect-URL>	TPPs individual redirect URI. (The redirect URI must match 1:1 with an URI defined for this client_id in the redirect URI registration endpoint (see chapter 1.2). It's possible to substitute characters by regular expressions in the portal request, also when it is not allowed to use regular expressions in the registration of redirect URIs.)

Request example in production (PIS):

```
https://api.santander.de/scb-
openapis/client/oauths/password/authorize?response_type=code&client_id=
PSDDE-BAFIN-123456&scope=PIS%3A6ee7548f-3fd7-45c1-a6e8-
a378e31d8726&state=S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw&redirect_u
ri=https://tppredirect.com/cb
```

Request example in Sandbox (AIS):

```
https://api-sandbox-cc.santander.de/scb-  
openapis/sx/oauthsos/password/authorize?response_type=code&client_id=PSDD  
E-SANDBOX-0006&scope=AIS%3A3087d8e2-2eb0-4e54-9af9-32ea8c6eef02&state=  
S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw&redirect_uri=  
https://tppredirect.com/cb
```

After successful authorization by the PSU, you will be provided with the “**authorization code**” inside the redirect URL. Additionally the parameter “**state**”, as given in the SCA request URL. The “**state**” inside the redirect URL enables to link the redirect answer to the original business process. If the SCA-process was not successful for any reason (customer did not sign, the payment was not possible, or any error during the execution), the redirect will answer with an error message. In any case the SCA page will redirect back to the **redirect_uri** given from the TPP.

The grant type “**authorization code**” to receive a higher privileged “Authorization Code Access Token” is no more used in the Santander Open Banking API version 1.2. This process is going to be decommissioned in the next release of the API.



PSU credentials to be used in the Sandbox SCA portal:

User: open
Password: banking
mTAN: 123456

Example for redirect answer URL after successful SCA process:

```
https://tppredirect.com/cb?code=Q1cKn5&state==  
S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw&redirect_
```

1.3.4 scaOAuth details (/ .well-known/)

The link refers to a specification of the OAuth details of our authorisation server. To retrieve the `authorization_endpoint` URL, call the `.../.well-known/... Service`.

This authorization endpoint URL needs to be part of the SCA request URL for the redirect SCA approach.

```
GET https://api-cc.santander.de/scb-openapis/client/.well-  
known/oauth-authorization-server  
--key '$PATH_TO_KEY_PEM'  
--cert '$PATH_TO_CERT_PEM:$PASSWORD_OF_CERT'
```

Respond example in production:

```
{
```

```

"authorization_endpoint": https://api.santander.de/scb-
openapis/client/oauthsos/password/authorize
}

```

Respond example in Santander Sandbox:

```

{
"authorization_endpoint": https://api-sandbox-cc.santander.de/scb-
openapis/sx/oauthsos/password/authorize
}

```

1.3.5 SCA Status

Available SCA status are:

Code	Description	Comment
psuidentified	The PSU related to the authorisation or cancellation-authorisation resource has been identified.	First status after a successful POST payment or post consent. In case of DECOUPLED, the next step is that the PSU has to approve the transaction in the Santander Sign APP.
psuAuthenticated	The PSU related to the authorisation or cancellation-authorisation resource has been identified and authenticated.	PSU credentials are validated and the mTAN was sent to the PSU.
finalised	The SCA routine has been finalised successfully. This is a final status of the authorisation resource.	SCA process finished successfully. To ensure about the transaction status also request the payments status!
failed	The SCA routine failed. This is a final status of the authorisation resource.	SCA and transaction failed.

2 Functional details

The Santander Open Banking API is fully compliant with the obligations according the PSD2 regulation from European Commission and additionally the Regulatory Technical Standards and all published opinions from the European Banking Authority. Technical baseline for the Santander Open Banking API is the Berlin Group specification as long as this is applicable for Santander's products and services. Please remember that that Berlin Group is a technical framework to standardize open banking services. This standard is in ongoing development and the obligations from the PSD2 are only included as a part of this framework.

In principle, the Santander Open Banking API follows the Berlin Group framework and due to that is not foreseen to copy the full functional specification. The following functional description explains the chosen solution and adds some specific details.

2.1 Consents

Following example for a `/constents` request body in raw JSON:

```
{
  "access": {
    "allPsd2": "allAccounts"
  },
  "recurringIndicator": false,
  "validUntil": "2021-11-25",
  "frequencyPerDay": 1,
  "combinedServiceIndicator": false
}
```

2.1.1 Global consents

The “global consent” is supported, and the expected access element is: `"allPsd2": "allAccounts"`

2.1.2 Dedicated consents with IBAN

Creating a `/consents` request with dedicated IBAN requires the following access details:

```
"access": {
  "accounts": [
    { "iban": "DE99123456789876543210", "currency": "EUR" },
    { "iban": "DE99123456789876543210", "currency": "EUR" }
  ],
  "balances": [
    { "iban": "DE99123456789876543210", "currency": "EUR" },
    { "iban": "DE99123456789876543210", "currency": "EUR" }
  ],
  "transactions": [
    { "iban": "DE99123456789876543210", "currency": "EUR" },
    { "iban": "DE99123456789876543210", "currency": "EUR" }
  ]
},
```

2.1.3 Online / Offline consents and recurring indicator

Every `.../transactions` request with a requested period of more than 90 days requires an **online** consent. Every consent is automatically an online consent for a period of 20 minutes from posting the POST `/consents` call.

The `"recurringIndicator"`: can be true or false. A call with indicator “true” is valid for max. 90 days, and a request with indicator “false” is automatically valid for 20 minutes.

! | In case `"recurringIndicator": false`, the only allowed value for
• | `"frequencyPerDay"` is **1**

2.1.4 Frequency per day and combined services

The maximum number of offline request per day is 4. Combined services are not supported.

```
"combinedServiceIndicator": false
```

2.2 Accounts

2.2.1 Transactions

The list of transactions can be received in JSON or XML format, depending on the decision given in the Request Header `Accept: application/json` or `Accept:`

```
application/xml
```

2.2.2 Transactions cursor

Requesting the list of transactions in JSON format (Header: `Accept: application/json`) the response body contains max. 100 transactions per response. In case the result of the request has more than 100 transactions, the response body ends with a link including a cursor to request the next set of max. 100 transactions.

2.2.3 Additional information

Additional information in the `/accounts/.../transactions` endpoint:

In case of expecting a response body in JSON format (Request Header “`Accept: application/json`”), the response of `GET /accounts/{resourceId}/transactions` may contain the additional parameter “`paymentReference`” which is not defined in the Berlin Group specification. This parameter is only for additional services and has no functional aspect in the PSD2 context.

2.3 Payments

2.3.1 Payment services

Available payment services are JSON `/sepa-credit-transfers/` and XML `/pain.001-sepa-credit-transfers`. Naturally GET requests for `.../{paymentId}` or `.../{paymentId}/status` must be requested within the same payment service as the initial payment was posted.

2.3.2 Payment service JSON

Example for JSON body:

```
{
  "instructedAmount": {
    "currency": "EUR",
    "amount": "1.50"
  },
  "debtorAccount": {
```

```

    "iban": "DE66123456789876543210"
  },
  "creditorName": "Max Mustermann",
  "creditorAccount": {
    "iban": " DE99876543219876543210"
  },
  "remittanceInformationUnstructured": "Santander API"
}

```

2.3.2.1 Payment initiation without debtor account

In case of using the JSON `/sepa-credit-transfers/` payment service for initiation of transactions, the debtor account inside of the request body is optional. If the debtor account is not included, the account will be chosen by the PSU in the REDIRECT SCA process. During the login by the PSU in the REDIRECT SCA Landing Page, the possible accounts are retrieved and in case the PSU owns more than one payments account, the customer has to choose his preferred account before approving the transaction.

Due to the fact that the debtor account is a mandatory element in the `pain.001.001.03` XML scheme, it is not supported to initiate transactions without debtor account in the `/pain.001-sepa-credit-transfers/` service.

2.3.3 Payment service XML

The supported scheme for a XML body is: `pain.001.001.03`

It is very important, that if the POST payment has been done within the `/pain.001-sepa-credit-transfers/` service, all GET and DELETE calls for this paymentId must be done within the same payment service type.

2.3.4 Payment status codes

The supported payment status codes are:

Code	Name	Description
RCVD	Received	POST <code>/payments</code> initiation has been received successfully.
ACTC	Accepted Technical Validation	Interim status for EMBEDDED and REDIRECT SCA. In case of EMBEDDED, the endpoint <code>"PUT payments/.../authorisations/..."</code> (<code>updatePsuAuthentication</code>) was called successfully. Within the REDIRECT process, the PSU has entered successfully the REDIRECT SCA Portal and requested the second factor. In both cases the initiated payment has also passed a first step of the technical validations successfully.

ACCP	Accepted Customer Profile	Only for Future Dated Payments and Periodic Payments! Interim status when the PSU SCA flow was performed successfully but the payment is not yet executed. ("scaStatus": "finalised")
ACFC	Accepted Funds Checked	Final status for every one-off or Future Dated Payment when the PSU SCA flow was performed successfully and the payment was executed.
RJCT	Rejected	Execution of the payment was not successful. The PSU has rejected the payment initiation within the SCA flow or the authorization was successful but additional transaction validations performed afterwards have failed.
CANC	Cancelled	Future Dated Payments or Periodic Payments has been cancelled before execution by TPP, PSU or ASPSP.

2.3.5 SEPA Instant Payments

Santander does not differentiate between instant and classic SEPA Payments, because for Santander the SEPA Instant Payment is already the new normal. There is a special service is used, which is called: "Instant Automatic" to enable fully automatic SEPA Instant Payments. Every SEPA transaction which is initiated in the Online Banking or in the Open Banking API is validated automatically if the transaction is available for SEPA Instant Payment and in case of yes, the transaction is, by default, automatically send as SEPA Instant Payment. If the instant transaction fails, because for example the beneficiary bank is temporary not available, the transaction is automatically and immediately re-send as classic SEPA payment. If the transaction was sent as instant or classic payment can be retrieved immediately in the account transactions.

That means, every `POST /v1/payments/sepa-credit-transfers` or `/v1/payments/pain.001-sepa-credit-transfers` is automatically send as SEPA Instant Payment, if possible.

! The possible endpoints `POST /v1/payments/instant-sepa-credit-transfers` or `/v1/payments/pain.001-instant-sepa-credit-transfers` **are not used!**

Every Future Dated Payment and every Periodic Payments is always processed as classic SEPA payment! (Excluded are Future Dated Payments with a requestedExecutionDate = actual date, which are either way executed immediately.)

2.4 Periodic Payments (Standing Order)

Before continuing to the details of Periodic Payments, please consider that these are only available for the payments service `/sepa-credit-transfers/`. The execution of

Periodic Payments is different to one-off payments. They are always executed as classic SEPA and not as SEPA Instant Payments.

Example for POST `/periodic-payments` body in raw JSON format:

```
{
  "debtorAccount": {
    "iban": "DE66123456789876543210",
    "currency": "EUR"
  },
  "instructedAmount": {
    "amount": "44.11",
    "currency": "EUR"
  },
  "creditorAccount": {
    "iban": " DE66123456789876543210",
    "currency": "EUR"
  },
  "creditorName": "Max Mustermann",
  "startDate": "2021-12-15",
  "endDate": "2022-12-31",
  "frequency": "Annual",
  "remittanceInformationUnstructured": "PSD2 Standing Order"
}
```

Details & Data Type:

Attribute	Type	Condition	Description
startDate	ISODate	mandatory	Any requested date must be at least 2 days in the future
endDate	ISODate	optional	If no date is given, the standing order will remain active indefinitely
frequency	see description	mandatory	see chapter 2.4.1 Frequency Codes

2.4.1 Frequency Codes

Supported Frequency Codes: Monthly, EveryTwoMonths, Quarterly, SemiAnnual, Annual

NOT supported Frequency Codes: all other

2.4.2 Execution rule and day of execution

All Periodic Payments are processed with the **executionRule "following"**. In case that the startDate is a Saturday, the execution will be on the following Monday, start of business day. Due to the fact that the date of next execution is being calculated from the given startDate

and the periodicity, the `dayOfExecution` is not considered in the input request. Therefore it is also not reported in the GET `/periodic-payments`.

If the real day of the next execution is needed, it can be read within the “`dayOfExecution`” from the list of active standing orders using the following endpoint:

```
GET /accounts/{account-id}/transactions?bookingStatus=information
```

It must be taken into account that the `dayOfExecution` may vary after every execution. E.g. if the last execution was on Wednesday 30.06.2021 (month-end) and the frequency is “Monthly”, the `dayOfExecution` = 02 because the next execution will take place on 02.08.2021 (31.07.2021 = Saturday).

2.4.3 Suspension of Periodic Payments

Furthermore, Santander customers are enabled to suspend the execution of Periodic Payments for a certain time period of time. If this happens, the `endDate` of the standing order is being overwritten with the start date of the suspension period. The modified date will be visible the next time the list of standing orders is being called.

If the customer cancels the suspension before the start date of the suspension period has been reached, the `endDate` of the Periodic Payments is being restored to its original value. If the customer does not cancel the suspension before the start date of the suspension period has been reached, on the first day the suspension period the `startDate` of the standing order will be overwritten with the (internal) end date of the suspension period + 1 day, and the `endDate` of the standing order is restored to its original value. The `dayOfExecution` will be adjusted accordingly.

2.4.4 Get all Periodic Payments

To get a list of all Periodic Payments just call the `.../transactions` endpoint with the parameter `?bookingStatus=information`:

```
GET /accounts/{resourceId}/transactions?bookingStatus=information
```

2.4.5 Update or delete Periodic Payments

To delete a Periodic Payment call: `DELETE /v1/periodic-payments/sepa-credit-transfers/{payment_id}`. Updating of existing periodic payments is not expected, please delete and create a new one.

Deletion of Periodic Payments created outside of the API is not possible, because of missing `payment_id`.

2.5 Overview of endpoints and details

Endpoint:	Method:	Resource:	Mandatory Header:	HTTP Response code: ¹
SCA OAuth (Discovery Service)	GET	<code>/.well-known/oauth-authorization-server</code>	- X-Request-ID: <UUID>	200 OK
TPP-Activation	POST	<code>/v1/tpp_registrations/mutual_tls</code>	- Content-Type: application/json - X-Request-ID: <UUID>	200 OK
Access Token	POST	<code>/v1/oauth_matls/token</code>	- Content-Type: application/x-www-form-urlencoded - X-Request-ID: <UUID> Body: urlencode - grant_type: "client_credentials"	200 OK
PUT Authorizations	PUT	<code>/v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}</code> or <code>/v1/consents/{consentId}/authorisations/{authorisationId}</code>	Authorization: "Bearer <access_token>" Content-Type: "application/json" PSU-ID: <PSU-ID> X-Request-ID: <UUID>	200 OK
Authorization status	GET	<code>/v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}</code> or <code>/v1/consents/{consentId}/authorisations/{authorisationId}</code>	Authorization: "Bearer <access_token>" X-Request-ID: <UUID>	200 OK
Consent	POST	<code>/v1/consents</code>	Authorization: "Bearer <access_token>" Content-Type: "application/json"	201 Created

			X-Request-ID: <UUID>	
Consent details	GET	/v1/consents/{consentId}	Authorization: "Bearer <access_token>" X-Request-ID: <UUID>	200 OK
Consent status	GET	/v1/consents/{consentId}/status	Authorization: "Bearer <access_token>" X-Request-ID: <UUID>	200 OK
Consent delete	DELETE	/v1/consents/{consentId}	Authorization: "Bearer <access_token>" X-Request-ID: <UUID>	204 No Content
Accounts	GET	/v1/accounts	- Authorization: "Bearer <access_token>" - Consent-ID: <consentId> - X-Request-ID: <UUID>	200 OK
Account details	GET	/v1/accounts/{resourceId}	- Authorization: "Bearer <access_token>" - Consent-ID: <consentId> - X-Request-ID: <UUID>	200 OK
Account balances	GET	/v1/accounts/{resourceId}/balances	- Authorization: "Bearer <access_token>" - Consent-ID: <consentId> - X-Request-ID: <UUID>	200 OK
Account transactions	GET	/v1/accounts/{resourceId}/transactions	- Authorization: "Bearer <access_token>" - Consent-ID: <consentId> - Accept: "application/json" or - Accept: "application/xml" - X-Request-ID: <UUID>	200 OK
Payment	POST	/v1/payments/sepa-credit-transfers	- Authorization: "Bearer <access_token>" - Content-Type: "application/json" - PSU-IP-Address: <PSU-IP-Address> - X-Request-ID: <UUID>	201 Created

Payment details	GET	/v1/payments/sepa-credit-transfers/{payment_id}	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	200 OK
Payment status	GET	/v1/payments/sepa-credit-transfers/{payment_id}/status	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	200 OK
Payment delete	DELETE	/v1/payments/sepa-credit-transfers/{payment_id}	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	204 No Content
Payment (pain format)	POST	/v1/payments/pain.001-sepa-credit-transfers	- Authorization: "Bearer <access_token>" - Content-Type: "application/xml" - PSU-IP-Address: <PSU-IP-Address> - X-Request-ID: <UUID>	201 Created
Payment details (pain format)	GET	/v1/payments/pain.001-sepa-credit-transfers/{payment_id}	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	200 OK
Payment status (pain format)	GET	/v1/payments/pain.001-sepa-credit-transfers/{payment_id}/status	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	200 OK
Payment delete (pain format)	DELETE	/v1/payments/pain.001-sepa-credit-transfers/{payment_id}	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	204 No Content
Periodic payment	POST	/v1/periodic-payments/sepa-credit-transfers	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	201 Created
Periodic payment details	GET	/v1/periodic-payments/sepa-credit-transfers/{payment_id}	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	200 OK

Periodic payment status	GET	/v1/periodic-payments/sepa-credit-transfers/{payment_id}/status	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	200 OK
Periodic payment delete	DELETE	/v1/periodic-payments/sepa-credit-transfers/{payment_id}	- Authorization: "Bearer <access_token>" - X-Request-ID: <UUID>	204 No Content
Funds confirmation	POST	/v1/funds-confirmations	- Authorization: "Bearer <access_token>" - Content-Type: "application/json" - X-Request-ID: <UUID>	200 OK

1 – Expected HTTP Response Code in case of successful request.