

## Quick Access Guide

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

#### Index:

List of URLs and endpoints.....	2
2 Technical baseline.....	4
2.1 Certificate in Santander Sandbox.....	4
2.2 SCA Approaches .....	4
2.2.1 DECOUPLED .....	5
2.2.2 EMBEDDED .....	5
2.2.3 SCA Status .....	6
2.2.4 Validity of the authorisation resource .....	6
3 Functional details .....	7
3.1 Consents .....	7
3.1.1 Global consents .....	8
3.1.2 Dedicated consents with IBAN.....	8
3.1.3 Online / Offline consents and recurring indicator.....	8
3.1.4 Frequency per day and combined services.....	8
3.2 Accounts.....	8
3.2.1 Balances.....	9
3.2.2 Transactions .....	9
3.2.3 Transactions balances .....	9
3.2.4 Transactions cursor .....	10

3.2.5 Additional information .....	10
3.3 Payments.....	10
3.3.1 Payment services .....	10
3.3.2 Payment service JSON.....	10
3.3.3 Payment service XML.....	11
3.3.4 Payment status codes.....	11
3.3.5 SEPA Instant Payments.....	12
3.4 Periodic Payments (Standing Order).....	12
3.4.1 Frequency Codes .....	13
3.4.2 Execution rule and day of execution .....	13
3.4.3 Suspension of Periodic Payments.....	14
3.4.4 Get all Periodic Payments.....	14
3.4.5 Update or delete Periodic Payments.....	14
3.5 Overview of endpoints and details .....	15

## List of URLs and endpoints

Production URL:

<https://api2-cc.santander.de>

Sandbox URL:

<https://api2-sandbox-cc.santander.de>

Endpoint:	Method:	Resource:
PUT Authorizations	PUT	/ {payment-service} /v1/ {payment-product} / {paymentId} /authorisations/ {authorisationId} or /consents/v1/ {consentId} /authorisations/ {authorisationId}
Authorization status	GET	/ {payment-service} /v1/ {payment-product} / {paymentId} /authorisations/ {authorisationId} or /consents/v1/ {consentId} /authorisations/ {authorisationId}
Consent	POST	/consents/v1

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

Funds confirmation	POST	/funds-confirmations/v1
--------------------	------	-------------------------

## 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 must 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-0008**

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 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.



**Please note that Santander Germany does no longer support the SCA method REDIRECT. Adding `TPP-Redirect-Preferred = true` will therefore result in an error.**

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	true	not given or valid or invalid	ERROR	
valid	not given or false	not given or invalid <sup>2)</sup>	ERROR	
not given or invalid	not given or false	valid	ERROR	
valid	not given or false	valid	EMBEDDED or DECOUPLED	<i>Explicit Embedded or Decoupled</i>

\* in case of an allPsd2 or availableAccount consent request, no debtor account validation is performed

<sup>2)</sup> when "not given", then consent "allPsd2" or "availableAccount" must be created

### 1.2.1 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".

### 1.2.2 EMBEDDED

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

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

or

```
PUT .../consents/v1/{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

### 1.2.3 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.

### 1.2.4 Validity of the authorisation resource

Every authorisation resource is valid for maximum 300 seconds! Depending on SCA approach, the moment of the creation of the authorisation resource is different. In case of EMBEDDED or DECOUPLED the authorisation resource is created in the same moment as the POST request is confirmed with "201 Created". The PSU must confirm the authorisation demand inside the SantanderSign APP, or the PUT password and PUT mTAN requests must be done successfully within the given period, otherwise the `scaStatus` will be updated automatically to `failed`.

When the authorization request is approved or rejected successfully, the payments, periodic-payments or consents status is updated immediately.



Also, when the SCA process for payments finished successfully with status “finalised”. Always request the payments status to ensure that the payment was not rejected due to final validations.

## 2 Functional details

The Santander Open Banking API is fully compliant with the obligations according to the PSD2 regulation of the European Commission and additionally the Regulatory Technical Standards and all published opinions from the European Banking Authority. The technical baseline for the Santander Open Banking API is the Berlin Group specification if 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 it 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 examples for `/consents` request bodies in raw JSON:

```
{
  "access": {
    "allPsd2": "allAccounts"
  },
  "recurringIndicator": false,
  "validUntil": "2024-11-25",
  "frequencyPerDay": 1,
  "combinedServiceIndicator": false
}

{
  "access": {
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": true,
  "validUntil": "2024-11-25",
  "frequencyPerDay": 1,
  "combinedServiceIndicator": false
}
```

### 2.1.1 Global consents

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

Additionally, the global consent type availableAccounts can be used:  
"availableAccounts": "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 requests per day is 4. Combined services are not supported.  
"combinedServiceIndicator": false

## 2.2 Accounts



### 2.2.1 Balances

The balances endpoint `/accounts/.../balances` enables to receive information about the balance details included in the following overview.

All balances are relating “booked” transactions (for details of transactions included see: 2.2.2 Transactions).

balanceType	Description
interimAvailable	Realtime balance of the account. <i>(In the Santander Online Banking, this balance is informed in the financial overview and in the transactions overview for the related account)</i>
expected	Realtime balance of the account: - reduced by authorized payments from Girocard (POS) and VISA DEBIT (POS + Online transactions)
authorised	Realtime balance of the account: + including the agreed overdraft facility - reduced by authorized payments from Girocard (POS) and VISA DEBIT (POS + Online transactions) <i>(In the Santander Online Banking, this balance is shown in the transactions initiation screen below the field “Betrag” labelled as “Maximalbetrag”).</i>

### 2.2.2 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`.

Only booked transactions are available! Not booked transactions are never informed in the transactions service, which for example includes known but not processed debits from SEPA Direct Debits or authorized payments form Giro- or VISA DEBIT card. These authorized payments explicitly reduce the “expected” and “authorised” balance (to be requested in the balances endpoint), but they are not informed with the booked transactions!

SEPA Instant Payments immediately credit or debit the account and these transactions are immediately included in the booked transactions.

### 2.2.3 Transactions balances

The balances section inside the transactions response informs about following details:

balanceType	Description
openingBooked	Book balance of the account at the beginning of the account reporting period.

closingBooked	<p>In case of a given “dateTo” parameter and the dateTo value is <u>not</u> the actual day the balance of the account at the end of the account reporting period is reflected.</p> <p>Otherwise the closingBooked balance is identically to the interimAvailable amount of the balances request and informs the realtime balance of the account, also including all intraday booked transactions.</p>
---------------	---

## 2.2.4 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.

## 2.2.5 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 global consent `availableAccounts` must be created to obtain the list of IBANs. The customer then must choose his preferred account before approving the transaction.

Since 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. The endpoint <code>"PUT payments/.../authorisations/..."</code> ( <code>updatePsuAuthentication</code> ) was called successfully. 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. ( <code>"scaStatus": "finalised"</code> )
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 /payments/v1/sepa-credit-transfers` or `/payments/v1/pain.001-sepa-credit-transfers` is automatically send as SEPA Instant Payment, if possible.



The possible endpoints `POST /payments/v1/instant-sepa-credit-transfers` or `/payments/v1/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. Because 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 considered 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 /periodic-payments/v1/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: <sup>1</sup>
PUT Authorizations	PUT	/ {payment-service} /v1/ {payment-product} / {paymentId} /authorisations/ {authorisationId} or /consents/v1/ {consentId} /authorisations/ {authorisationId}	Content-Type: "application/json" PSU-ID: <PSU-ID> X-Request-ID: <UUID>	200 OK
Authorization status	GET	/ {payment-service} /v1/ {payment-product} / {paymentId} /authorisations/ {authorisationId} or /consents/v1/ {consentId} /authorisations/ {authorisationId}	X-Request-ID: <UUID>	200 OK
Consent	POST	/consents/v1	Content-Type: "application/json" X-Request-ID: <UUID>	201 Created
Consent details	GET	/consents/v1/ {consentId}	X-Request-ID: <UUID>	200 OK
Consent status	GET	/consents/v1/ {consentId} /status	X-Request-ID: <UUID>	200 OK
Consent delete	DELETE	/consents/v1/ {consentId}	X-Request-ID: <UUID>	204 No Content
Accounts	GET	/accounts/v1	- Consent-ID: <consentId> - X-Request-ID: <UUID>	200 OK
Account details	GET	/accounts/v1/ {resourceId}	- Consent-ID: <consentId> - X-Request-ID: <UUID>	200 OK
Account balances	GET	/accounts/v1/ {resourceId} /balances	- Consent-ID: <consentId> - X-Request-ID: <UUID>	200 OK

Account transactions	GET	/accounts/v1/{resourceId}/transactions	- Consent-ID: <consentId> - Accept: "application/json" <b>or</b> - Accept: "application/xml" - X-Request-ID: <UUID>	200 OK
Payment	POST	/payments/v1/sepa-credit-transfers	- Content-Type: "application/json" - PSU-IP-Address: <PSU-IP-Address> - X-Request-ID: <UUID>	201 Created
Payment details	GET	/payments/v1/sepa-credit-transfers/{payment_id}	- X-Request-ID: <UUID>	200 OK
Payment status	GET	/payments/v1/sepa-credit-transfers/{payment_id}/status	- X-Request-ID: <UUID>	200 OK
Payment delete	DELETE	/payments/v1/sepa-credit-transfers/{payment_id}	- X-Request-ID: <UUID>	204 No Content
Payment (pain format)	POST	/payments/v1/pain.001-sepa-credit-transfers	- Content-Type: "application/xml" - PSU-IP-Address: <PSU-IP-Address> - X-Request-ID: <UUID>	201 Created
Payment details (pain format)	GET	/payments/v1/pain.001-sepa-credit-transfers/{payment_id}	- X-Request-ID: <UUID>	200 OK
Payment status (pain format)	GET	/payments/v1/pain.001-sepa-credit-transfers/{payment_id}/status	- X-Request-ID: <UUID>	200 OK
Payment delete (pain format)	DELETE	/payments/v1/pain.001-sepa-credit-transfers/{payment_id}	- X-Request-ID: <UUID>	204 No Content
Periodic payment	POST	/periodic-payments/v1/sepa-credit-transfers	- X-Request-ID: <UUID>	201 Created
Periodic payment details	GET	/periodic-payments/v1/sepa-credit-transfers/{payment_id}	- X-Request-ID: <UUID>	200 OK



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

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