|  |
| --- |
| SDC |
| PSD2 |
| Getting Started  Version 1.6 |

Contents

[2. Introduction 5](#_Toc69368008)

[2.1. 2 different ways to access SDC PSD2 5](#_Toc69368009)

[2.2. Overview on calling the PSD2 API 6](#_Toc69368010)

[3. TPP approval by FSA 8](#_Toc69368011)

[4. Register with SDC for access to PSD2 API 9](#_Toc69368012)

[5. How to get an Access (and refresh) Token 10](#_Toc69368013)

[5.1. Preconditions: 10](#_Toc69368014)

[5.2. Calling the Access Provider as a TPP 11](#_Toc69368015)

[5.2.1. Token Endpoints 11](#_Toc69368016)

[5.2.2. Token Swagger endpoint 11](#_Toc69368017)

[5.2.3. Input for ‘Authorization Grant’ call (getting Access Token) 11](#_Toc69368018)

[5.2.4. AccessProvider Login: 13](#_Toc69368019)

[5.2.4.1. SDC Biometric 13](#_Toc69368020)

[5.2.4.2. DK- NemID 2 factor 15](#_Toc69368021)

[5.2.4.3. DK- NemID OCES 17](#_Toc69368022)

[5.2.4.4. NO- BankID brikke 19](#_Toc69368023)

[5.2.4.5. NO- BankID Mobil 21](#_Toc69368024)

[5.2.4.6. SE- BankID File 22](#_Toc69368025)

[5.2.4.7. SE- BankID Mobil QR Code 23](#_Toc69368026)

[5.2.4.8. 1-SCA process 24](#_Toc69368027)

[5.2.5. References 26](#_Toc69368028)

[5.3. Exchanging Authorization Code or refreshToken to an access token 28](#_Toc69368029)

[5.3.1. Example for getting and changing AuthorizationCode with Token: 28](#_Toc69368030)

[5.4. Test Client App 37](#_Toc69368031)

[5.5. Calling the Access Provider as a Trusted TPP 41](#_Toc69368032)

[6. Calling the API Manager 44](#_Toc69368033)

[6.1. Preconditions for calling the API 44](#_Toc69368034)

[6.2. API Manager Endpoints 44](#_Toc69368035)

[6.3. Swagger 44](#_Toc69368036)

[6.4. Calling the APIs 45](#_Toc69368037)

[6.4.1. API Endpoint 45](#_Toc69368038)

[6.5. Calling the payment confirmation endpoint 45](#_Toc69368039)

[6.5.1. Calling the payment confirmation endpoint for a Multiple Approver payment 46](#_Toc69368040)

[6.6. AIS. Pagination 46](#_Toc69368041)

[7. Fallback solution 49](#_Toc69368042)

[8. FAQ 50](#_Toc69368043)

[8.1. General 50](#_Toc69368044)

[8.2. AIS 50](#_Toc69368045)

[8.3. PIS 50](#_Toc69368046)

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.8 | Aug 2019 | Chnx | Document created. |
| 1.0 | Sep 2019 | Chnx | Updated with go-live information |
| 1.1 | Oct 2019 | Chnx | Updated based on first use from TPPs |
| 1.2 | Feb 2020 | Ema | Updates with   * Examples on authentication flow * Authentication flow description * Updated postman with examples of input/output for all calls |
| 1.3 | Apr 2020 | Chnx | Link to get NemID keycard updated to NETS standard |
| 1.4 | Jun 2020 | Chnx | * Authorisation grant input table updated with new values * Biometry option added to login and signing |
| 1.5 | Aug 2020 | Chnx | * Login overview process flows updated |
| 1.6 | Apr 2021 | Chnx | * PIS. Multiple Approver (6.5.1) * PIS. 1-SCA payments (5.2.4.8) * AIS. Pagination of large requests (6.6) |

# Introduction

SDC offers a PSD2 solution based on the OAUTH 2.0 standard for authentication and the Berlin Group Standard for APIs.

This document will help you understand

* What is offered by SDC in relation to PSD2
* How to onboard to SDC services
* How to use the Authentication / Authorization
* How to use the APIs
* How to use the fallback solution if necessary

The SDC API allows similar access to payment accounts and payments as other SDC channels. This includes Domestic, Instant, Bulk, Cross border, Giro and Periodic payments (PIS) and account information requests (AIS). Both types can be done for both Private and Corporate accounts. The interface is the same for all of SDCs 120+ banks.

To get access and use the APIs you need to go through the following steps

1. Get approved by your local FSA (e.g. Finanstilsynet in DK) (section 3 of this document)
2. Register for access to SDC via <http://sdcinfo.dk/tpp/>   
   SDC will issue credentials to
   1. Authentication server (Access Token)
   2. API Manager (APIs)
3. Obtain a test user with SCA credentials (NemID, BankID NO or BankID SE) and send this to SDC for creation in test – Or use the test users specified by NETS on <https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx> these are already in SDC test environments as a standard.
4. Test with SDCs test environments (which are dynamic, so no stubs or static data)
5. Confirm test with SDC
6. Request PROD access

## 2 different ways to access SDC PSD2

SDC works with 2 different approaches to PSD2; Normal and Trusted TPP.

The Normal PSD2 and trusted TPP access use the same APIs but vary in how you obtain the access token for the PSU.

In the normal approach you call the authentication endpoint and either use the redirect towards the National SCA processes (NemID/BankID/MitID) or use the SDC Biometric option, which is a decoupled App similar to the PSU login to SDC’s eBank. Possible Login types (national schemes) are

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Strong Customer Authentication (SCA) | DK | FO | NO | SE | FI |
| NemID Bank 2 factor | X | X | - | - | - |
| NemID OCES | X | - | - | - | - |
| Norwegian BankID (brikke) | - | - | X | - | - |
| Norwegian BankID (Mobile) | - | - | X | - | - |
| Swedish BankID (file) | - | - | - | X | - |
| Swedish Mobile BankID (QRCode) | - | - | - | X | - |
| Biometry | X | X | X | - | - |
| Note: All tags must be URL encoded using %20 to replace whitespaces. | | | | | |

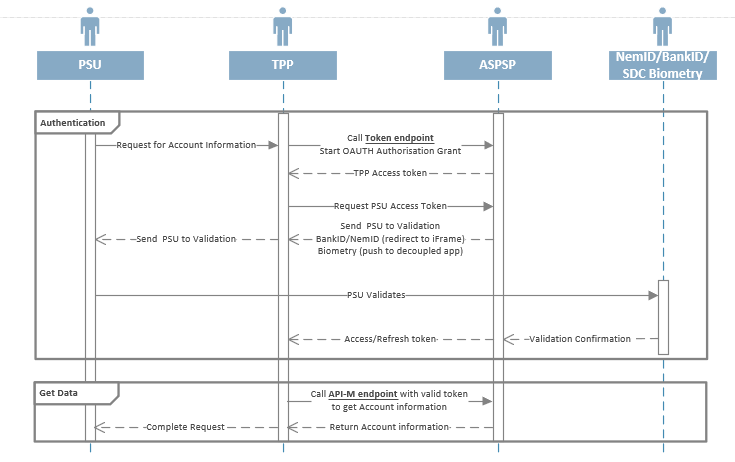
As a Trusted TPP, the TPP handles SCA internally and accept liability and reporting requirements for all transactions. This requires commercial agreement between the TPP and the SDC banks in question. A trusted TPP issues their own token and exchanges to an SDC token based on the trusted relationship. This is handled via the standard <https://tools.ietf.org/html/rfc7523>. Should you wish to pursue this option please contact SDC directly.

If you do not wish to use the PSD2 API you can access data at SDC through the fallback solution. This is described at the end of this document.

## Overview on calling the PSD2 API

When you have gone through above mentioned steps to set up access and have received credentials/test users you are ready to call the Token endpoint and the API. A normal process for calling the API follows these steps.

1. Get an Access token from <https://auth.sdc.dk/>
   1. Start the authentication process by an authorization grant request to token endpoint
   2. Access Request. Call the token endpoint to establish a connection
   3. Return Access token.
   4. Create request. Call the token endpoint with a specific user to get an Access Token
   5. Token endpoint will redirect to the national SCA selected to validate user
   6. Access (and refresh) token is issued.   
      With a valid refresh token (expires after 90 days) you can get a new access token without redirecting to national SCA.
2. With the Access token call the API and get information on accounts, transactions details or create a payment



If you have any questions, please feel free to use the support form available on <http://sdcinfo.dk/tpp/> for requests.

# TPP approval by FSA

This process is handled between the TPP and the relevant FSA.

When you apply for access at SDC we will check with the FSAs if you are either approved, in process to be approved or passported to one of the countries we cover (NO, DK, FO, FI, SE).

# Register with SDC for access to PSD2 API

To get started with the process go to <http://sdcinfo.dk/tpp/>. Fill out the form and then SDC will revert to you within 5 working days.

You will need to provide details on your company including a company email (not related to a person, ex. [psd2@company.dk](mailto:psd2@company.dk)) and details on a contact person.

In addition, you will also have to supply the public part of your certificate, either an X.509 certificate or an eIDAS QWAC certificate. This will be used to validate you when calling the API.

Last you will have to supply a static IP for testing as we whitelist your access point for entry into our test-environments.

Once your registration is completed you will receive two set of credentials to call the Access Provider (Token Endpoint) and API Manager respectively.

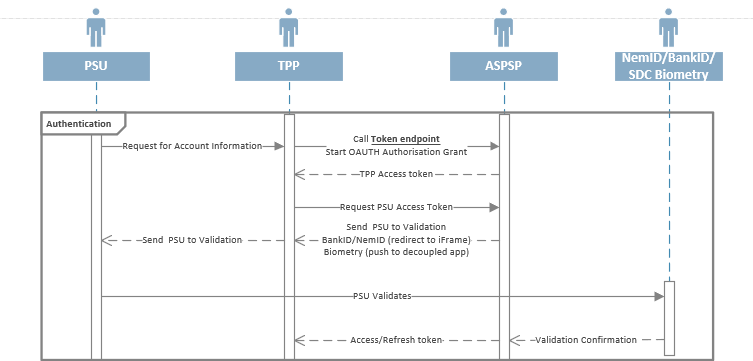
If you have any questions, please feel free to use the support form available on <http://sdcinfo.dk/tpp/> for requests.

# How to get an Access (and refresh) Token

In the following it is described how to get an access/refresh token from the SDC access provider. The access token is used in subsequent calls to the API.

Access tokens in TEST and PROD have a lifetime of 5 minutes.

The process looks as follows:



## Preconditions:

TPP is registered with SDC and have received credentials to call the SDC Access provider for one (or multiple) SDC bank(s). This is necessary as SDC represents 120+ banks and need to know where to route the request.

Note: If you are an Aggregator acting on behalf of other TPP’s then regulation stipulate that you should call with the eIDAS QWAC certificate of the TPP that receive the information from the service (PIS, CBPII/PIIS or AIS). In this case we expect that the Aggregators own credentials and the final recipient TPPs QWAC certificate is used.

SDC TEST servers are IP restricted. To test you must supply the IPs or IP Range in advance to SDC who will allow access through the firewall.

SDC requires use of the TLS 1.2 protocol when calling endpoints.

## Calling the Access Provider as a TPP

When calling the Access Provider (token endpoint) it follows a standard Oauth 2.0 flow. You should expect following steps.

1. Call login endpoint with State (GUID format) to get an Authorization Code to use in the next steps of the flow.
2. Call the token endpoint with ‘Authorization Grant’ and use the authorization code and state. Please see details of the token in below table 1.
   1. Go through BankID/NemID/Biometric dialogue with the User ID
   2. SDC will check on the redirect URL (if it matches the one given at registration) and return a URL-string with auth code in parameter ?code=authorization code and &state=GUID.
   3. SDC will return the Authorization code
3. Access and refresh token are returned

### Token Endpoints

You can find the access provider at following endpoints

TEST: <https://azure-auth-t2.test.sdc.dk>/Token   
PROD: https://auth.sdc.dk/Token

### Token Swagger endpoint

Swagger can be found at

TEST: <https://azure-auth-t2.test.sdc.dk/swagger/index.html>  
PROD: <https://auth.sdc.dk/swagger/index.html>  
(Link only works in Chrome)

### Input for ‘Authorization Grant’ call (getting Access Token)

The following HTTP GET starts the Access Provider Login flow:

Please set HTTP header "SDCGUID" to a UUID, which will be used by the SDC Logging Framework.

The name-value pairs in the message are described in Table 1 for Login.

|  |  |
| --- | --- |
| **Name** | **Example Value** |
| Endpoint | “<https://azure-auth-t2.test.sdc.dk/Account/Login?>”  Endpoint from either test of prod |
| response\_type | “code” |
| grant\_type | "authorization\_code" or "refresh\_token" |
| code | "[Authorization Code]" or "[refreshToken]" |
| Redirect URL | <http://redirect.company.dk>  The redirect you have provided when signing up. |
| ClientID | "19ea41c1-40d9-42e8-8f66-ee3e7db491bc"  Client reference for the bank that you are calling to obtain data. Issued by SDC to TPP in onboarding process |
| ClientSecret | "[client-Secret]"  Client secret obtained from SDC in onboarding process |
| Scope | “psd2.aisp”  The scopes that you have been approved for. Possible values are psd2.aisp psd2.pisp psd2.piisp psd2:aisp psd2:pisp psd2:piisp |
| State | State |
| Login type  (Optional) | “NemID%20Bank%201%20factor&”  Login type that you wish to use. Following values are accepted: NemID+Bank+2+factor  NemID+OCES Norwegian+Bankid+(brikke)  Norwegian+Bankid+(Mobile) Swedish+BankId+(file) Swedish+Mobile+BankID+(QRCode) Biometry  Note: If login type is empty or omitted, then all the possible Login types are shown in a list |
| Device id  (Optional) | " Windows10Mobile "  Note: If deviceID is empty or omitted, then the HTTP header user agent is used as deviceID |

Example:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=171be1e7-5960-4aa0-8dc5-81dc7a597e34

&state=state\_this

&login\_type=NemID+Bank+2+factor

&deviceID=Windows10Mobile

### AccessProvider Login:

From the Login, you will get ’Authorization Code’ in the querystring.

You have this possibility to exchange ‘Authorization Code’ to ‘AccessToken’ from Token service (<https://azure-auth-t2.test.sdc.dk/Token>).

You can use the below url, but you need to have ‘scope’, ‘clientId’ and ‘redirect url’:

[https://azure-auth-t2.test.sdc.dk/Account/Login?scope=[scope]&response\_type=code&redirect\_uri=[redirect\_Url]&client\_id=[client\_Id]&state=state\_this](https://azure-auth-t2.test.sdc.dk/Account/Login?scope=%5bscope%5d&response_type=code&redirect_uri=%5bredirect_Url%5d&client_id=%5bclient_Id%5d&state=state_this)&login\_type=[ login\_type]

### SDC Biometric

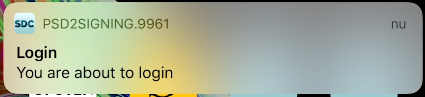
SDC offers a biometric option to log-in the PSU and for the PSU to sign payments. The feature is built into the iOS and Android mobilebank apps offered by SDC and is the same User Interface as the PSU uses to log into SDC mobilebank.

Preconditions:

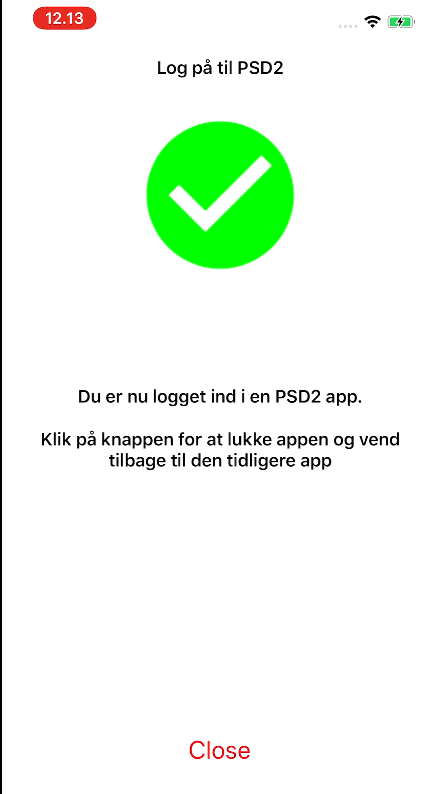
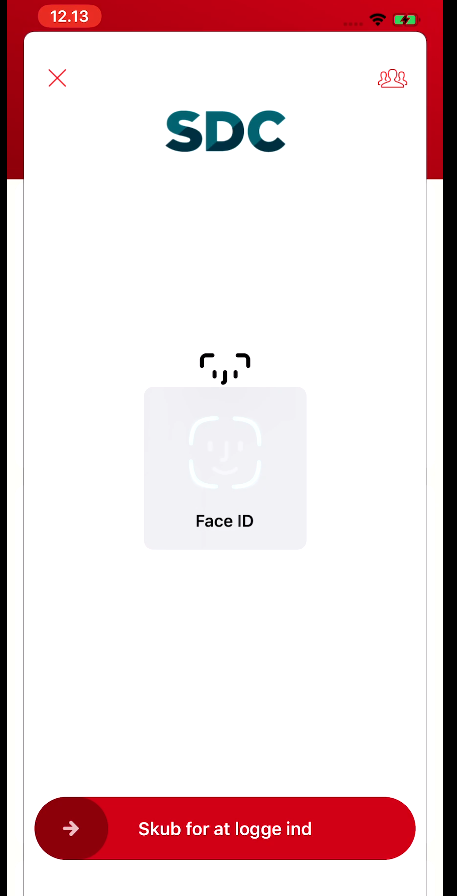
* PSU has downloaded SDC mobilebank App
* PSU has activated Biometric option in his Mobilebank (includes device binding)
* PSU has activated push messages on

Call the above URL with LoginType = Biometry. This will trigger a push message to the PSU. Once the PSU has confirmed the login an Access/Refresh Token will be issued

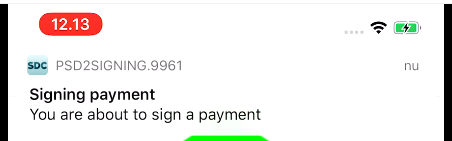
Push message will look similar to

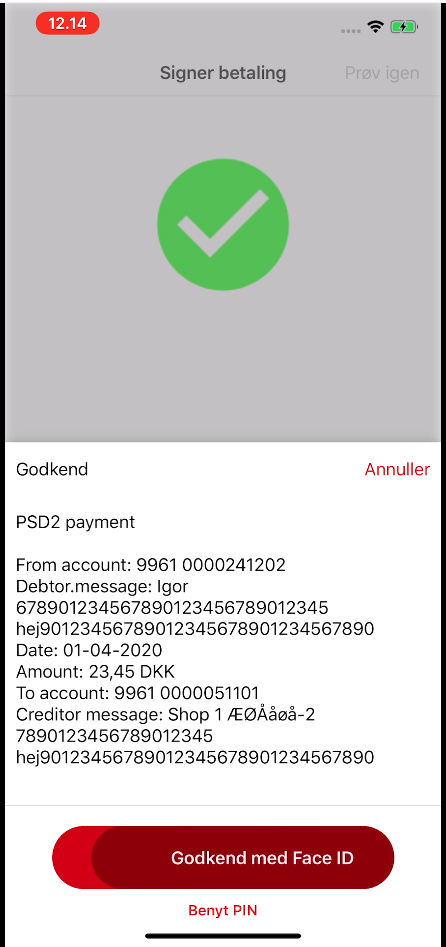
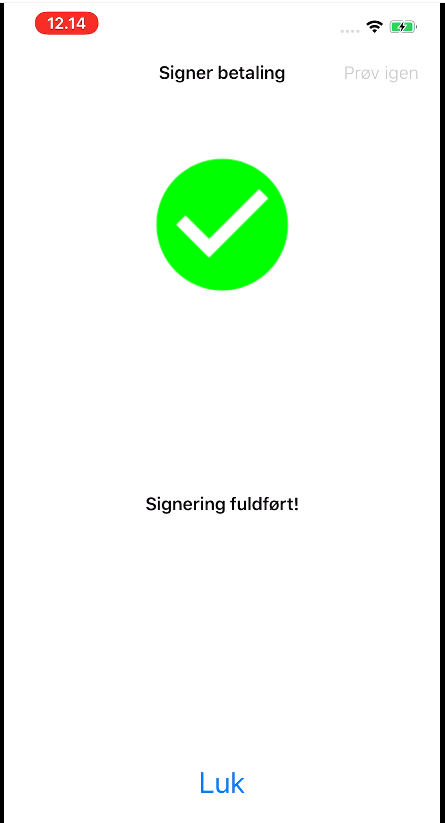


##### Login flow on iOS



##### Signing flow on iOS



### DK- NemID 2 factor

How to generate your own test user

(<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#nemid>):

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

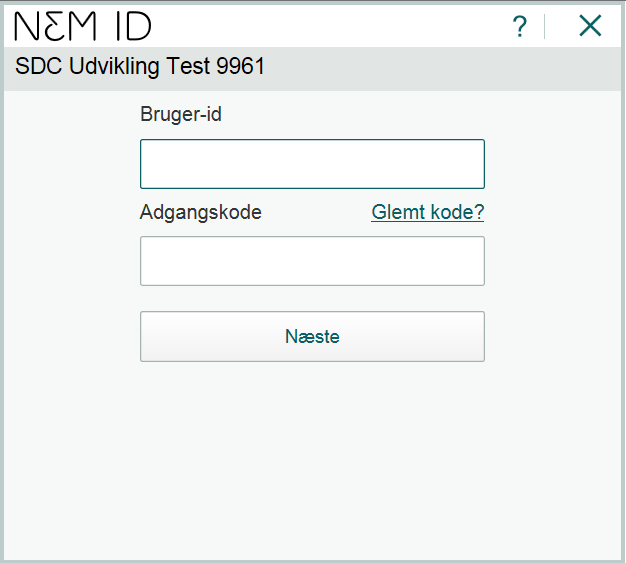
&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=171be1e7-5960-4aa0-8dc5-81dc7a597e34

&state=state\_this

&login\_type=NemID+Bank+2+factor



Test User:

Cpr: 1403532411

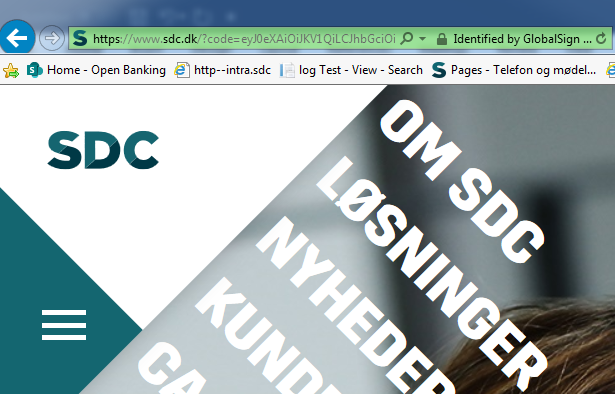
pass: asasas12

OTP: You can find the nemid(nøglekort) from the below url:

<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#nemid>



After login, you will get ‘Authorization Code’ in the querystring(the velue of ‘code’ in querystring).



### DK- NemID OCES

How to generate your own test user:

<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#moces>

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

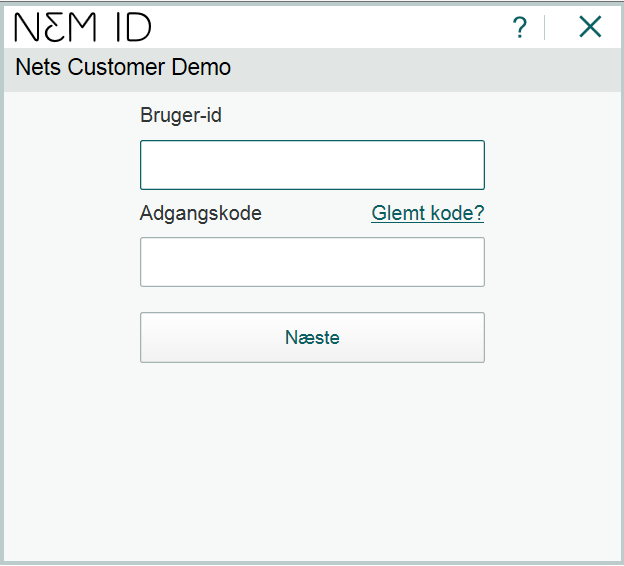
&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=171be1e7-5960-4aa0-8dc5-81dc7a597e34

&state=state\_this

&login\_type=NemID+OCES



Test User:

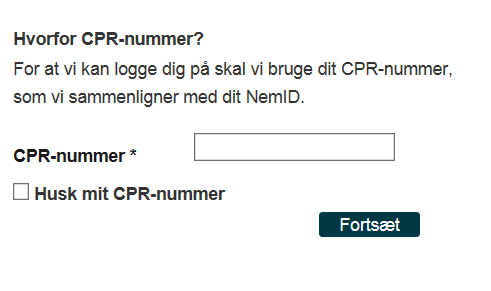
Cpr: 1403532411

pass: asasas12

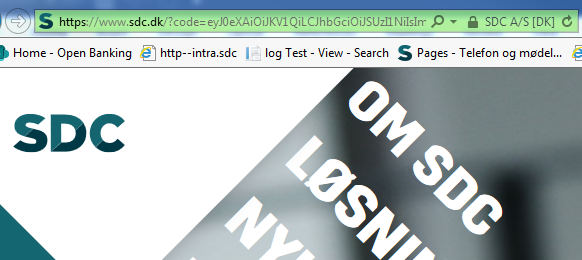
OTP: You can find the nemid(nøglekort) from the below url:

[<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#nemid>](https://10.31.123.75/simulator/listotpcards)





After enter your cpr, you will get ‘Authorization Code’ in the querystring(the velue of ‘code’ in querystring).



### NO- BankID brikke

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=583294b8-6228-4d4c-941d-f5e05dc44e79

&state=state\_this

&login\_type=Norwegian+Bankid+(brikke)

UserTest:

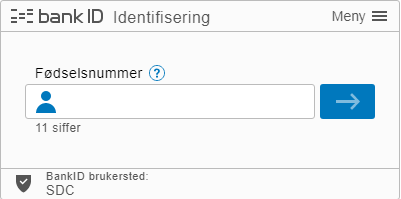
cpr=21016100061

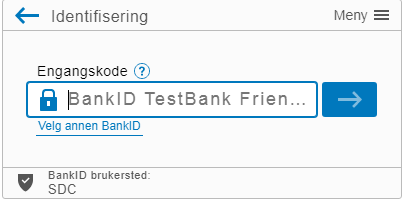
Dummy password: otp

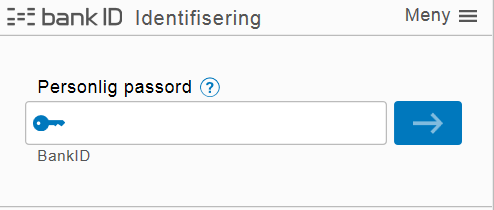
pw: qwer1234

More Test User:

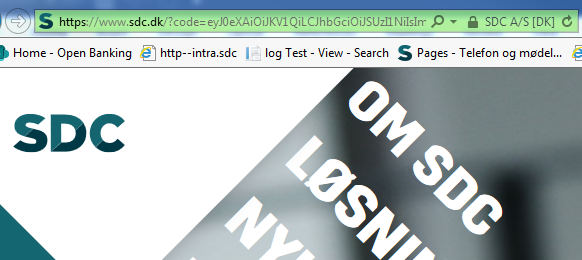
<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#bankidno>







After enter your cpr, you will get ‘Authorization Code’ in the querystring(the velue of ‘code’ in querystring).



### NO- BankID Mobil

To test BankID on mobile, you need a dedicated mobile phone with a SIM-card for test purposes.

More Info:

<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#bankidonmobile>

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

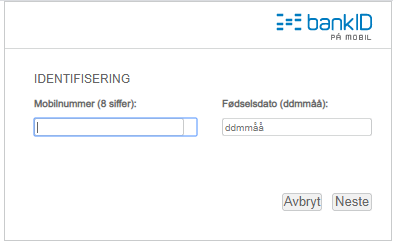
&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=583294b8-6228-4d4c-941d-f5e05dc44e79

&state=state\_this

&login\_type=Norwegian+Bankid+(Mobile)



### SE- BankID File

How to install and get a testUser:

<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#bankidse>:

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

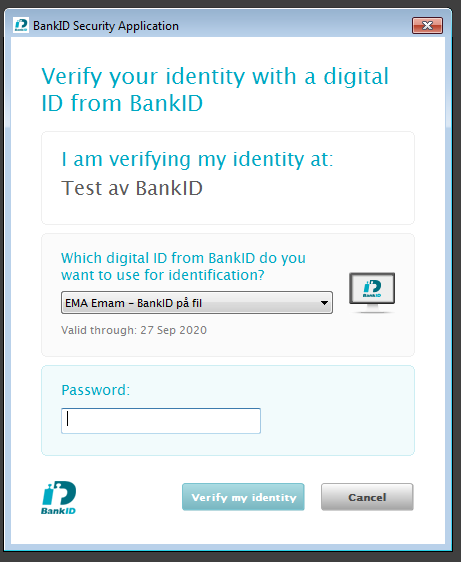
&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=855e1a9e-c930-4ed4-957c-662c05d3bf02

&state=state\_this

&login\_type=Swedish+BankId+(file)





### SE- BankID Mobil QR Code

You should install BankId app on you mobilephone.

After choosing this one, a window with QRCode wil be opend:

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=855e1a9e-c930-4ed4-957c-662c05d3bf02

&state=state\_this

&login\_type=Swedish+Mobile+BankID+(QRCode)



You should scan the code with BankId app on your mobile.

After scaning you will get ‘Authorization code’ in query string:



### 1-SCA process

The SDC eBank operations with a 2-SCA process; 1 for login and 1 for signature.

Based on the EBA opinion from 4th June 2020 we have however introduced an option for TPPs to handle a payment flow where the PSU only is asked for one SCA at signing.

NB! The 1-SCA process is only available with Biometric option, i.e. it cannot be used with National identification schemes (BankID / NemID).

In order to activate the 1-SCA process SDC has enabled that any TPP can call and get a token through the token exchange protocol. This token can be gained without SCA approval from the PSU. With the token the payment is created through the same API as ‘normal’ payments are created.

When calling the endpoint make sure to set the "signtype=biometric".

Please see following example for the gathering the access token

Endpoint: <https://auth.sdc.dk/Token>

POST Example:

{

"grant\_type": "urn:ietf:params:oauth:grant-type:jwt-bearer",

"code": "[base 64 encoded JWT]",

"redirect\_uri": "",

"client\_id": "[SDC delivered client\_id (guid)]",

"scope": "[SDC defined scope like NEOS APE psd2.aisp]"

}

Please set HTTP header "SDCGUID" to a UUID, which will be used by the SDC Logging Framework.

The claims inside the signed JWT must contain the following:

{

"iss": "[Aggreed on issuer]",

"sub": "[Identification of caller]",

"aud": "[Agreed on audience]",

"exp": [JWT Standard claim - must not be expired],

"nbf": [JWT Standard claim],

"iat": [JWT Standard claim],

"jti": "[Unigue GUID for each request - used to prevent token replay]",

"urn:sdc.dk:chnl”: "[Agreed on channel]"

"urn:sdc.dk:orgid": "[JHREG for calling bank]"

}

The signed JWT can contain one or more of these additional claims

{

"acr": [Logon strength 0-4, will default to 2 if missing],

"amr": [Logon method],

"sid": [Session id (guid)],

"urn:sdc.dk:ssn": "[Social Security Number for end user]"

"urn:sdc.dk:orn": "[Organization Registration Number related to end user]"

}

Once payment is created then the Payment ID is approved/confirmed in the same way as normal payments in the Biometric signing process.

### References

#### URL to Nets (test users):

<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#nemid>

#### NemdID ErrorCodes:

You can check the errorCode from NemID Here:

<https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Documents/NemID%20Error%20Codes.pdf>

#### ClientId for tests

You can make url by below clientIds for test:

Url Link for test:

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=[clientId]

&state=state\_this

&login\_type=[LoginType]

Note: For Below clientIds:

scope=psd2.aisp

redirect\_uri=https%3A%2F%2Fwww.sdc.dk

you can use the below list for Login\_Type:

1. NemID+Bank+2+factor
2. NemID+OCES
3. Norwegian+Bankid+(brikke)
4. Norwegian+Bankid+(Mobile)
5. Swedish+BankId+(file)
6. Swedish+Mobile+BankID+(QRCode)
7. Biometry

clientIds for DK (regNr: 9961 ):

* c85044f1-3c18-4b8a-a086-507284bda22d
* 5db19cfc-c7c7-4a64-ad56-57d15ea2ae2b
* 9fd21a16-1d55-45a9-b4dd-2acb8d0421a4
* a4bbfd33-86a3-47ff-9763-b9504c6d0e07
* 91c5c650-917e-47f8-b75a-4e64a8721b55

clientIds for NO (regNr: 4730 ):

* e0173e5a-99e0-4464-93c5-5b2c59c0a8a0
* 002765d1-b7c9-413b-ae8c-31135f4bd759
* 0d76851e-99d5-4382-a692-065804318201
* 09c4a509-fea1-4a93-b28e-7f930dd17f6f
* bac68188-63ba-484d-b74a-8bee1648f6b8

clientIds for SE (regNr: 9927 ):

* 97cb38bd-628a-4e8a-9e6e-cc15ecd2c274
* aa7609ac-1416-4917-be08-7e498233b055
* fbcf5595-df0d-4c47-8d39-ad127f9886e2
* e41330ed-53aa-4554-84cd-86cff4bef388
* 59b3acbc-905e-41af-8d03-77a0fc5dd46a

### **Exchanging Authorization Code or refreshToken to an access** token

After Login, you get an Authorization code in querystring,

you can use the authorizationCode to exchange with Token:

* swagger: <https://azure-auth-t2.test.sdc.dk/swagger>
* service: <https://azure-auth-t2.test.sdc.dk/Token>

Note: Please open swagger by Chrome

- for AuthorizationCode:

{

"grant\_type": "authorization\_code",

"code": "[AuthorizationCode]",

"redirect\_uri": "[redirect-uri]",

"client\_id": "[client-Id]",

"client\_secret": "[client-Secret]",

"scope": "psd2.pisp

}

-for refreshToken:

{

"grant\_type": "refresh\_token",

"code": "[refreshToken]",

"redirect\_uri": "[redirect-uri]",

"client\_id": "[client-Id]",

"client\_secret": "[client-Secret]",

"scope": "psd2.aisp"

}

### Example for getting and changing AuthorizationCode with Token:

##### Note

To make a test, you need to have below info:

* ClientID
* RedirectUrl
* Scope
* ClientSecret

##### Login

<https://azure-auth-t2.test.sdc.dk/Account/login>?

scope=psd2.aisp

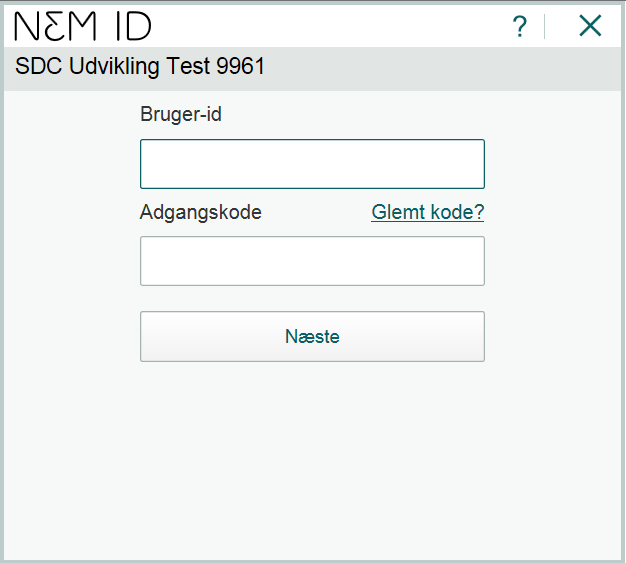
&response\_type=code

&redirect\_uri=https%3A%2F%2Fwww.sdc.dk

&client\_id=a4bbfd33-86a3-47ff-9763-b9504c6d0e07

&state=state\_this

&login\_type=NemID+Bank+2+factor





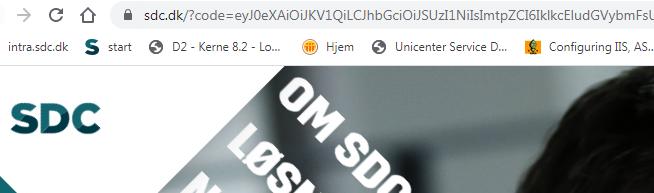
Test User:

Cpr: 1403532411

pass: asasas12

OTP: You can find the nemid(nøglekort) from the below url:

[<https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx#nemid>](https://10.31.123.75/simulator/listotpcards)



url:

##### Get Authorization code from querystring (Value of ‘code’):

https://www.sdc.dk/?code=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcEludGVybmFsU2lnbmluZ0tleS8yZDlkYWNmNDM0OTU0OTk3YmZlOWE0MDE0MTc0NGM4NSJ9..IsWerxQXxb1XfOSF6XaH6a5spsWZ7THYYyqgdipOSlPhMeVLUEMwOuoeV0v7hAPLDS-Kt4-EPMkFGXaN6Qd4Ga5Co\_0wk7LqfnZefMMm4h1omRs0uZWHeQ2YctQhr2YVSsJ0XOWyesA2zLyF3zuL6W2xAcEFDSyrhY2aRQDkv7ozYPdwO1MtB1YE9W0v0CuF6QbIh44K8SwWD8DTIM50hf52yrOagFzy6fa1\_1NQ9dbGRMnW4y1ANXfRVpsepqtFl1JDJkXvwBasHP208Y0tBaqkdZB1LOZFr22jvFF2v-Vqw8Asjcz3k0oiBJn9CLtEIu7kjAX1Dt3twjWdKRiZGA&state=state\_this

##### Now Call the Swagger:

<https://azure-auth-t2.test.sdc.dk/swagger>

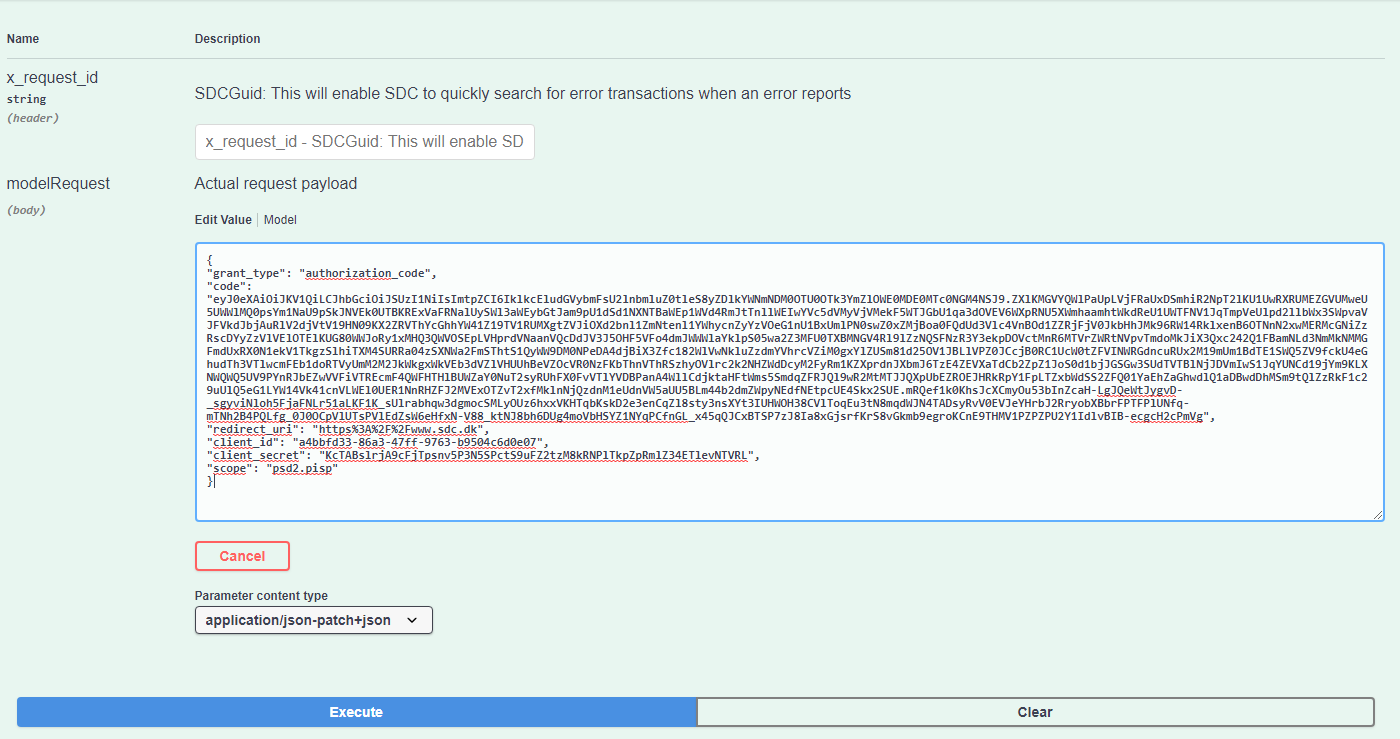
Go to TokenService and Choose ‘ /Token’

* Note: Everytime after login, a new authorizationCode would be generated.
* Client Secret is unique for ClientID, and it will create the same time with ClientID:

Example for ClientID and ClientSecret:

"client\_id": "a4bbfd33-86a3-47ff-9763-b9504c6d0e07",

"client\_secret": "KcTABslrjA9cFjTpsnv5P3N5SPctS9uFZ2tzM8kRNPlTkpZpRmlZ34ETlevNTVRL",



* Here is one example based on the above clientid and authorziationCode:

{

"grant\_type": "authorization\_code",

"code": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcEludGVybmFsU2lnbmluZ0tleS8yZDlkYWNmNDM0OTU0OTk3YmZlOWE0MDE0MTc0NGM4NSJ9..IsWerxQXxb1XfOSF6XaH6a5spsWZ7THYYyqgdipOSlPhMeVLUEMwOuoeV0v7hAPLDS-Kt4-EPMkFGXaN6Qd4Ga5Co\_0wk7LqfnZefMMm4h1omRs0uZWHeQ2YctQhr2YVSsJ0XOWyesA2zLyF3zuL6W2xAcEFDSyrhY2aRQDkv7ozYPdwO1MtB1YE9W0v0CuF6QbIh44K8SwWD8DTIM50hf52yrOagFzy6fa1\_1NQ9dbGRMnW4y1ANXfRVpsepqtFl1JDJkXvwBasHP208Y0tBaqkdZB1LOZFr22jvFF2v-Vqw8Asjcz3k0oiBJn9CLtEIu7kjAX1Dt3twjWdKRiZGA",

"redirect\_uri": "https%3A%2F%2Fwww.sdc.dk",

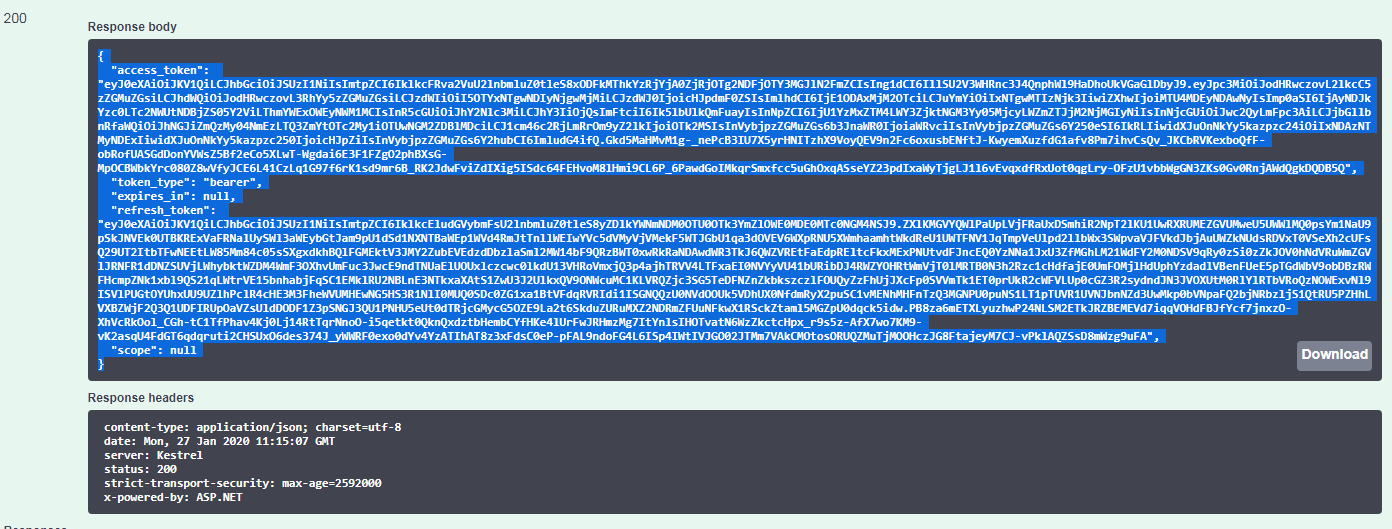
"client\_id": "a4bbfd33-86a3-47ff-9763-b9504c6d0e07",

"client\_secret": "KcTABslrjA9cFjTpsnv5P3N5SPctS9uFZ2tzM8kRNPlTkpZpRmlZ34ETlevNTVRL",

"scope": "psd2.pisp"

}

##### Execute the service and get the Token:



The answer:

{

"access\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcFRva2VuU2lnbmluZ0tleS8xODFkMThkYzRjYjA0ZjRjOTg2NDFjOTY3MGJlN2FmZCIsIng1dCI6IllSU2V3WHRnc3J4QnphWl9HaDhoUkVGaGlDbyJ9..Gkd5MaHMvM1g-\_nePcB3IU7X5yrHNITzhX9VoyQEV9n2Fc6oxusbENftJ-KwyemXuzfdG1afv8Pm7ihvCsQv\_JKCbRVKexboQfF-obRofUASGdDonYVWsZ5Bf2eCo5XLwT-Wgdai6E3F1FZgO2phBXsG-MpOCBWbkYrc080Z8wVfyJCE6L41CzLq1G97f6rK1sd9mr6B\_RK2JdwFviZdIXig5ISdc64FEHvoM8lHmi9CL6P\_6PawdGoIMkqrSmxfcc5uGhOxqASseYZ23pdIxaWyTjgLJ1l6vEvqxdfRxUot0qgLry-OFzU1vbbWgGN3ZKs0Gv0RnjAWdQgkDQDB5Q",

"token\_type": "bearer",

"expires\_in": null,

"refresh\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcEludGVybmFsU2lnbmluZ0tleS8yZDlkYWNmNDM0OTU0OTk3YmZlOWE0MDE0MTc0NGM4NSJ9..PB8za6mETXLyuzhwP24NLSM2ETkJRZBEMEVd7iqqVOHdFBJfYcf7jnxzO-XhVcRkOol\_CGh-tC1TfPhav4Kj0Lj14RtTqrNnoO-i5qetkt0QknQxdztbHembCYfHKe4lUrFwJRHmzMg7ItYnlsIHOTvatN6WzZkctcHpx\_r9s5z-AfX7wo7KM9-vK2asqU4FdGT6qdqruti2CHSUxO6des374J\_yWWRF0exo0dYv4YzATIhAT8z3xFdsC0eP-pFAL9ndoFG4L6ISp4IWtIVJGO02JTMm7VAkCMOtosORUQZMuTjMOOHczJG8FtajeyM7CJ-vPklAQZSsD8mWzg9uFA",

"scope": null

}

##### for Regenerating a new token:

you need to use refresh token from above answer of the service:

{

"grant\_type": "refresh\_token",

"code": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcEludGVybmFsU2lnbmluZ0tleS8yZDlkYWNmNDM0OTU0OTk3YmZlOWE0MDE0MTc0NGM4NSJ9..PB8za6mETXLyuzhwP24NLSM2ETkJRZBEMEVd7iqqVOHdFBJfYcf7jnxzO-XhVcRkOol\_CGh-tC1TfPhav4Kj0Lj14RtTqrNnoO-i5qetkt0QknQxdztbHembCYfHKe4lUrFwJRHmzMg7ItYnlsIHOTvatN6WzZkctcHpx\_r9s5z-AfX7wo7KM9-vK2asqU4FdGT6qdqruti2CHSUxO6des374J\_yWWRF0exo0dYv4YzATIhAT8z3xFdsC0eP-pFAL9ndoFG4L6ISp4IWtIVJGO02JTMm7VAkCMOtosORUQZMuTjMOOHczJG8FtajeyM7CJ-vPklAQZSsD8mWzg9uFA",

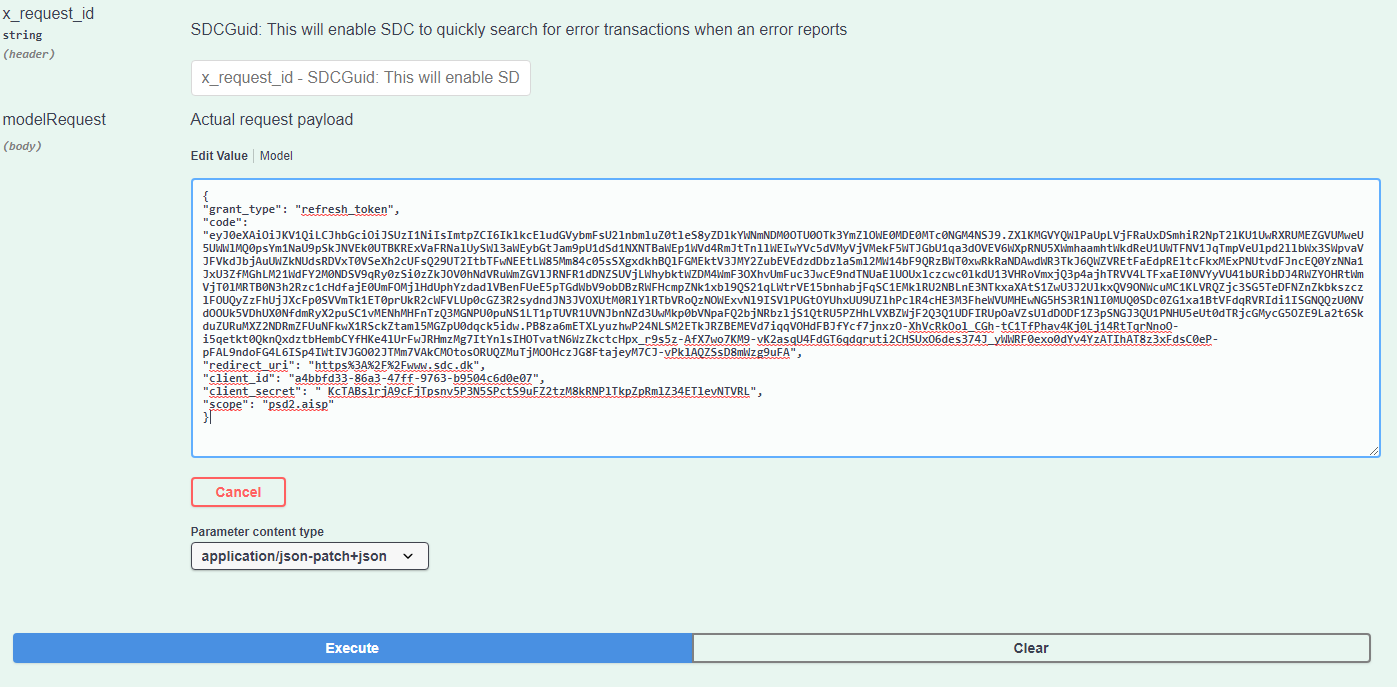
"redirect\_uri": "https%3A%2F%2Fwww.sdc.dk",

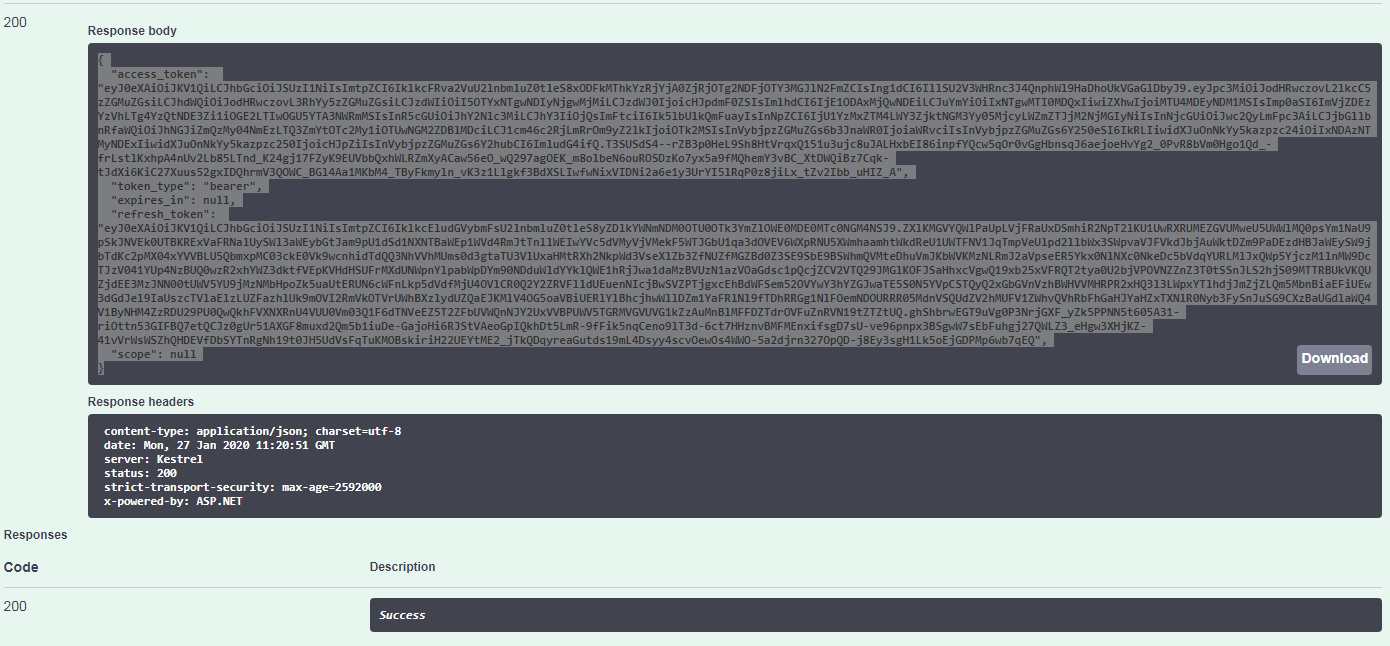
"client\_id": "a4bbfd33-86a3-47ff-9763-b9504c6d0e07",

"client\_secret": " KcTABslrjA9cFjTpsnv5P3N5SPctS9uFZ2tzM8kRNPlTkpZpRmlZ34ETlevNTVRL",

"scope": "psd2.aisp"

}





{

"access\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcFRva2VuU2lnbmluZ0tleS8xODFkMThkYzRjYjA0ZjRjOTg2NDFjOTY3MGJlN2FmZCIsIng1dCI6IllSU2V3WHRnc3J4QnphWl9HaDhoUkVGaGlDbyJ9..T3SUSdS4--rZB3p0HeL9Sh8HtVrqxQ151u3ujc8uJALHxbEI86inpfYQcw5qOr0vGgHbnsqJ6aejoeHvYg2\_0PvR8bVm0Hgo1Qd\_-frLstlKxhpA4nUv2Lb85LTnd\_K24gj17FZyK9EUVbbQxhWLRZmXyACaw56eO\_wQ297agOEK\_m8olbeN6ouROSDzKo7yx5a9fMQhemY3vBC\_XtDWQiBz7Cqk-tJdXi6KiC27Xuus52gxIDQhrmV3QOWC\_BGl4Aa1MKbM4\_TByFkmyln\_vK3z1Llgkf3BdXSLIwfwNixVIDNi2a6e1y3UrYI5lRqP0z8jiLx\_tZv2Ibb\_uHIZ\_A",

"token\_type": "bearer",

"expires\_in": null,

"refresh\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IklkcEludGVybmFsU2lnbmluZ0tleS8yZDlkYWNmNDM0OTU0OTk3YmZlOWE0MDE0MTc0NGM4NSJ9..ghShbrwEGT9uVg0P3NrjGXF\_yZk5PPNN5t605A31-riOttn53GIFBQ7etQCJz0gUr51AXGF8muxd2Qm5b1iuDe-GajoHi6RJStVAeoGpIQkhDt5LmR-9fFik5nqCeno9lT3d-6ct7HHznvBMFMEnxifsgD7sU-ve96pnpx3BSgwW7sEbFuhgj27QWLZ3\_eHgw3XHjKZ-41vVrWsWSZhQHDEVfDbSYTnRgNh19t0JH5UdVsFqTuKMOBskiriH22UEYtME2\_jTkQDqyreaGutds19mL4Dsyy4scvOewOs4WWO-5a2djrn327OpQD-j8Ey3sgH1Lk5oEjGDPMp6wb7qEQ",

"scope": null

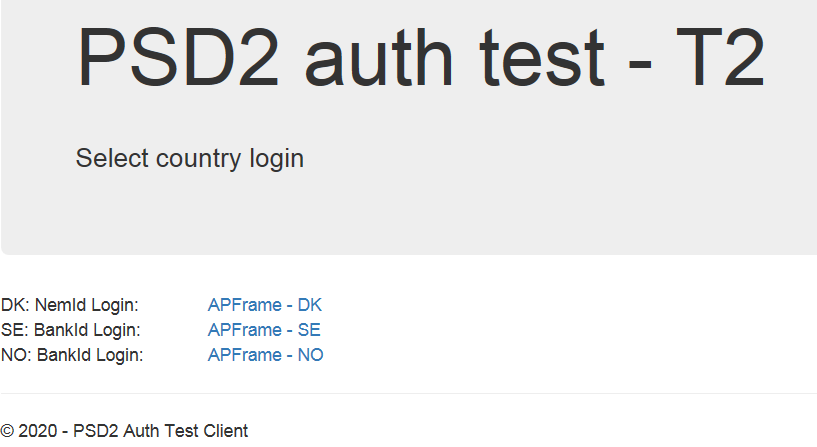
}

## Test Client App

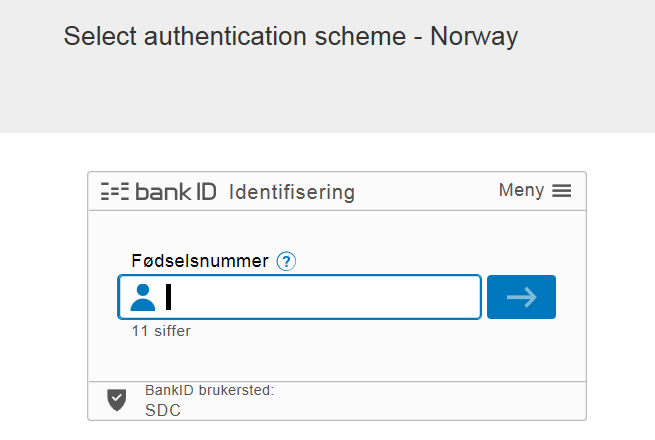
You can Also use this app to test Psd2 Login and GetToken:

<https://app-tek-c1-01:444/PSD2Test-T2/Home/Login>

You can Choose the Country & login Type:





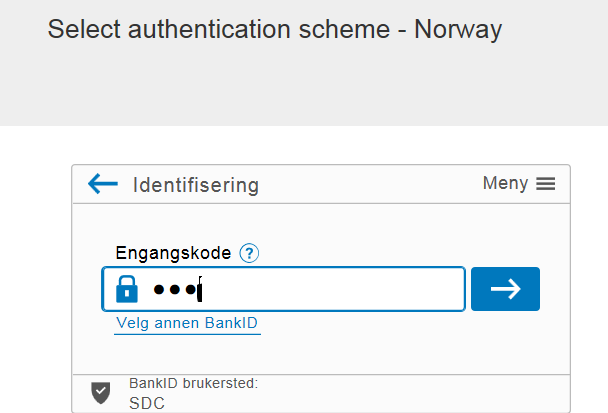


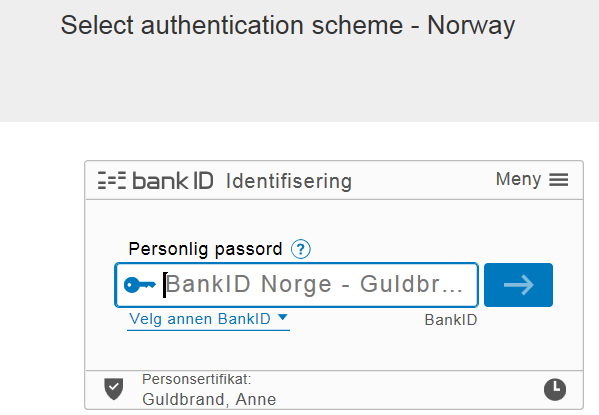
UserTest:

cpr=21016100061

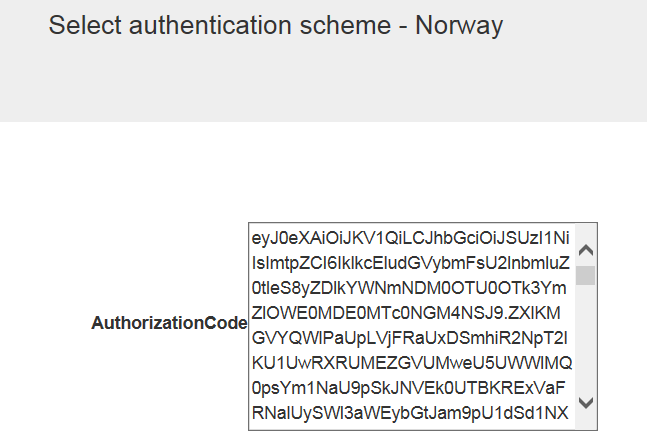
Dummy password: otp

pw: qwer1234

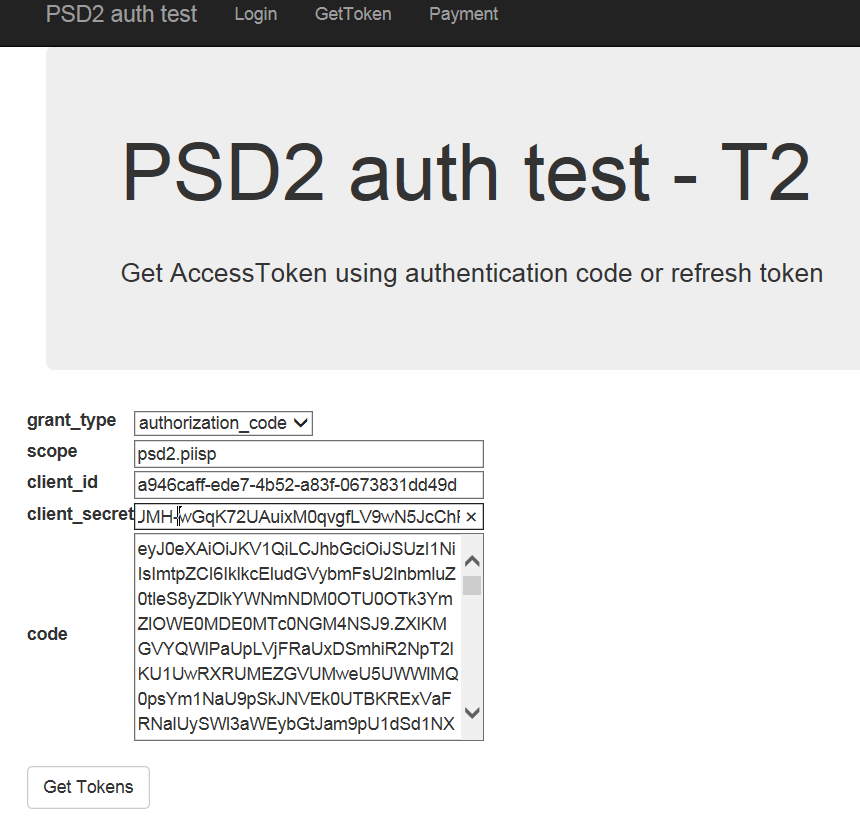


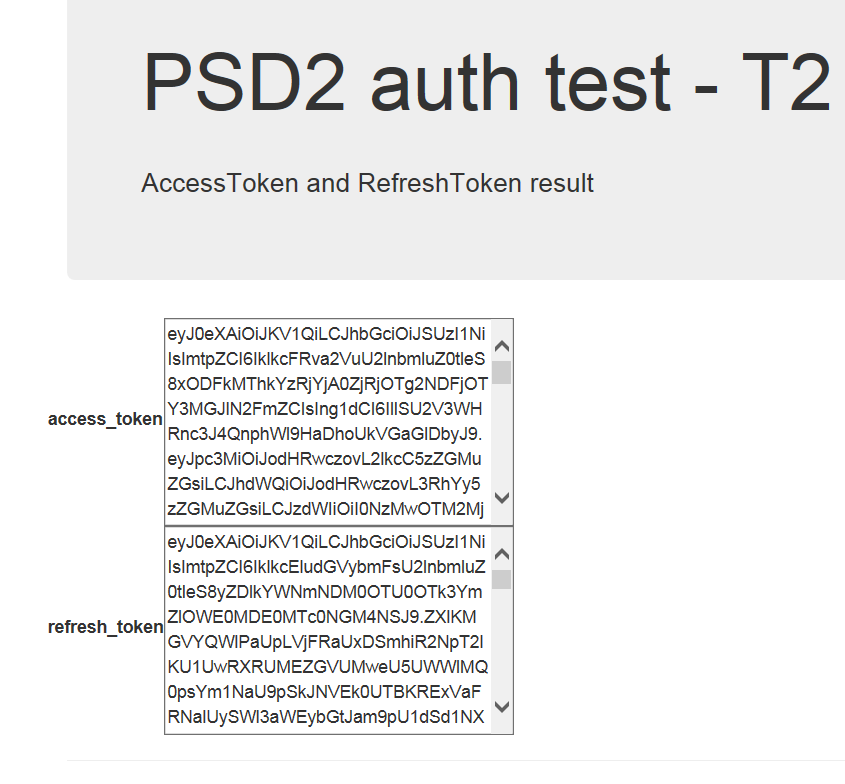


Here you have AuthorizationCode:



After That from menu, choose ‘GetToken’





Here you have AccessToken and RereshToken.

## Calling the Access Provider as a Trusted TPP

Structure of JWT token to SDC Access Provider

The field "code" is a base 64 encoded JWT. The JWT consists of header, a payload and a signature.

Please set HTTP header "SDCGUID" to a UUID, which will be used by the SDC Logging Framework.

JWT Header

The header part of the JWT must have this JSON-format:

{

"alg": "[the signing algorithm used, e.g. H256]",

"typ": "JWT"

}

JWT Payload

The payload part of the JWT must have this JSON-format:

{

"iss": "TPP ABC"

"sub": "DK-DKFSA-12345678”

"aud": "[see table]"

"exp": [JWT Standard claim - must not be expired],

"nbf": [JWT Standard claim],

"iat": [JWT Standard claim],

"jti": "[Unigue GUID for each request]",

"urn:sdc.dk:cnty": "DK",

"urn:sdc.dk:orgid": "[Specific Bank registration number – JHREG, see table]",

"urn:sdc.dk:chnl": "PMA",

"urn:sdc.dk:ssn": "[Social Security Number for end user, see table]"

}

In the following Table 2 is described the claims in payload part of the JWT  
See also <https://tools.ietf.org/html/rfc7519>

|  |  |
| --- | --- |
| **Claim Name** | **Claim Value** |
| "iss" | The "iss" (**iss**uer) claim identifies the principal that issued the JWT.  The issuer value to be used by the TPP is: “**TPP ABC**” |
| "sub" | The "jti" (**J**W**T** **I**D) claim provides a unique identifier for the JWT. Must be a unique ID (GUID) for each request  E.g. “E455B669-1FAF-4C68-B574-4C6E3B9F413A” |
| "aud" | The "aud" (**aud**ience) claim identifies the recipients that the JWT is intended for. In test environments use the value "**urn: SDC-Test**" In prod. environments use the value "**urn: SDC**" |
| “acr” | The security level of the claim. Shold always be ‘4’. |
| "exp" | The "exp" (**exp**iration time) claim identifies the expiration time on or after which the JWT MUST NOT be accepted for processing. |
| "nbf" | The "nbf" (**n**ot **b**e**f**ore) claim identifies the time before which the JWT MUST NOT be accepted for processing. |
| "iat" | Optional. The **I**ssued **at** token is the time the token was issued. |
| "urn:sdc.dk:cnty" | The **c**ou**nt**r**y** code must be "**DK”, “SE”, “NO” “FI”or “FO”** |
| "urn:sdc.dk:orgid" | The **org**anization **id** must be the registration number – JHREG of the financial institute in context (four digits).  In test environment 8.2: FED Test bank (9961) - 601x: ”**9961**”  In prod. environment: BankNordik: ”**6503**” Lån og Spar Bank: ”**0400**” |
| "urn:sdc.dk:orgidt" | This must always be ‘idor’ |
| "urn:sdc.dk:chnl" | "[Bank **ch**an**n**e**l** - intn]" Use the value “**TTPP**” |
| "urn:sdc.dk:ssn" | "[**s**ocial **s**ecurity **n**umber for end user]" |

Table 2: Payload part of JWT, claim names and claim values

JWT Signature

The signature is made by using the TPP client certificate.

“code” = base64enc(header) + “.” payload + “.” + signature)

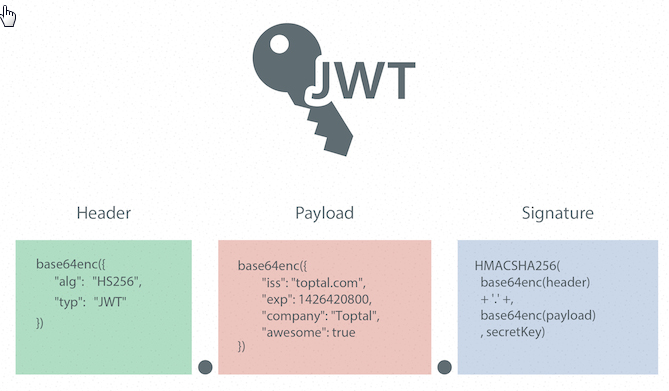


Figure 1: Jason Web Token structure

# Calling the API Manager

With the access token in hand (generated for the first time or through the refresh token) you are now ready to call the API Manager.

SDC follows the Berlin Group standard for PSD2 APIs. General information on this initiative and general Swagger files can be found here, <https://www.berlin-group.org/psd2-access-to-bank-accounts>

## Preconditions for calling the API

Certificates  
The TPP must preregister a certificate to be used for validating requests to the SDC API. There must be one certificate for test environment and one certificate for production environment.

The TPP keeps the certificate private key and sends certificate public key to SDC.

The test certificates can be self-signed X.509 certificates or eIDAS QWAC certificates.

PROD certificates must be eIDAS QWAC certificates.

Other

As a TPP you have received the subscription keys for the API manager – Or you are able to retrieve keys from the developer portal, that you have access to.

As a TPP once you log into the API Manager (APIM or API-M) then you can have been granted access to several products. As a minimum you should have the current PSD2 BG product.

For SDC PSD2 APIs you should use the product called PSD2 BG.

SDC requires use of the TLS 1.2 protocol when calling endpoints.

## API Manager Endpoints

The API Manager can be reached at the following endpoints

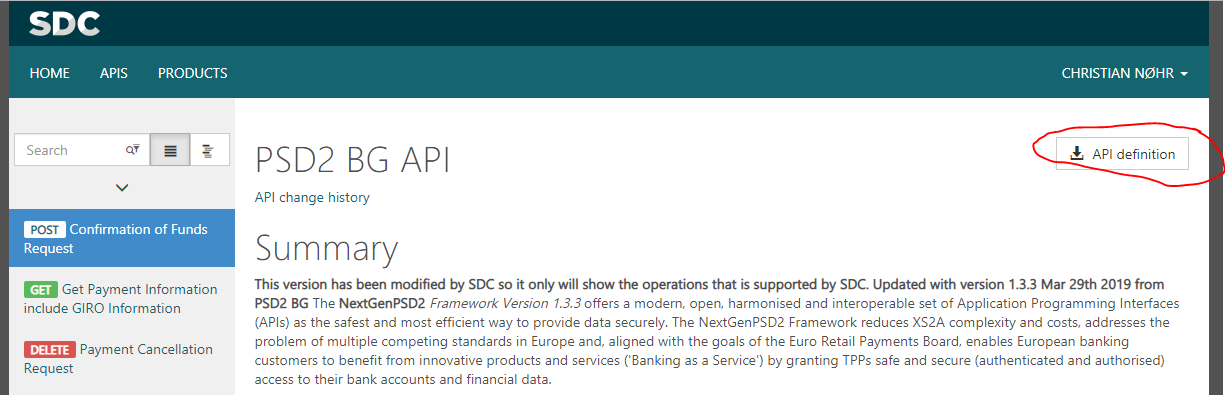
TEST: <https://api-portal.test.sdc.dk/>

PROD: <https://api-portal.sdc.dk/>

In the API Manager you can find the actual endpoint to be called from the TPP. It should look like following example for the Account look up function (in TEST): https://api-proxy.test.sdc.dk/api/psd2/v1/accounts/.

## Swagger

Swagger is available when you log into the API Manager and access the product



## Calling the APIs

Once you are ready to call the APIs this can be done following the swagger.

## API Endpoint

TEST: https://api-proxy.test.sdc.dk/

PROD: https://api-proxy.sdc.dk/

Please see the following postman collections for examples on how to call the different operations in the BG Standard. Updated Postman collections and test results (with text based examples, if you don’t use postman) can be found on <https://sdcinfo.dk/tpp/> under the tab ‘Documentation’.

NB!!!!  
Note that because of the authentication requirements then the endpoints for signing and approving a payment are different (same as authentication) than the other parts of the API. This endpoint is called regardless of which type of payment (domestic, giro, cross border) is created. See below how it works.

### Calling the payment confirmation endpoint

The Payment confirmation endpoint is slightly different from calling the other APIs.

Process is as follows

1. Call the Create Payment function of your choice (Giro, Domestic, etc.) in the API-Manager
2. When the payment is created part of the return value is a paymentID
3. Call our signing provider to approve and execute this payment at (TEST:   
   [https://azure-auth-t2.test.sdc.dk/](https://azure-auth-t2.test.sdc.dk/swagger/index.html?urls.primaryName=AccessProvider%20V1) PROD: https://auth.sdc.dk) with following parameters /Account/Payments/{paymentId}/{paymentType}   
   NB.: this is a different endpoint than is used for creating the payment.
4. The payment type parameter controls if you run a one-step (most) or two-step approval (internet trades where approval is done now, but money transferred later).
   1. You can either call the ‘Approve’ and then call ‘Confirm’ later when ready to ship goods (mainly internet shopping payments) or
   2. Call ‘Pay’ service that executes ‘Approve’ and ‘Confirm’ in one go (typically for account-to-account transfers).

### Calling the payment confirmation endpoint for a Multiple Approver payment

SDC supports multiple approver payments, where the payment requires multiple approvals to be issued. The flow for this process follows the ordinary flow (see 6.5) with one exception.

When the Approver 1 approves the payment it would normally change status from ‘RCVD’ to ‘ACSP’. Instead you as TPP will see the following return message and the status will change from ‘RCVD’ to ‘ACWP’. When the payment is in state ‘ACWP’ it is awaiting the next approver.

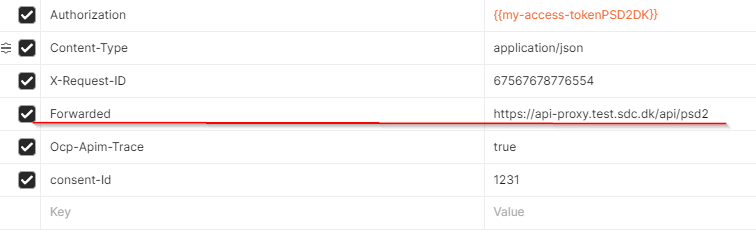
As in SDC eBank then Approver 2 have to access the payment through his own channels and confirm the payment. This can be done either via the PSD2 interface (by calling Approve with the right PSU) or can be done through the normal eBank.

### AIS. Pagination

SDC paginates large AIS requests to ensure performance for all TPPs. This means that if the period you request in the API (if you do not request a date range it will automatically be set to next banking day and going back 30 days. The limit is set at 50 transactions per request.

The solution follows the pagination standard used in the Berlin Group standard APIs.

As a standard you as a TPP have to set the following header value in your request (with PROD link when calling PROD).



You will then get following response and have to follow the link to get the next 50 transactions.

"transactionId": "996123252014879@YFAC3437@2021-02-02@2021-02-02-11.09.11.150309",

"entryReference": "2021-02-02-11.09.11.150309",

"creditorAccount": {

"iban": "",

"currency": "DKK"

},

"debtorName": "Nancy Hansen",

"debtorAccount": {

"bban": "99610000036048",

"currency": "DKK"

},

"transactionAmount": {

"currency": "DKK",

"amount": "-236.89"

},

"bookingDate": "2021-02-02",

"valueDate": "2021-02-02",

"remittanceInformationUnstructuredArray": [

"FN:02022021 000010"

],

"\_links": {

"transactionDetails": {

"href": "https://api-proxy.test.sdc.dk/api/psd2/v2/accounts/996123252014879/transactions/996123252014879@YFAC3437@2021-02-02@2021-02-02-11.09.11.150309"

}

}

}

],

"pending": [],

"information": [],

"\_links": {

"account": {

"href": "https://api-proxy.test.sdc.dk/api/psd2/v2/accounts/996123252014879"

},

"next": {

"href": "https://api-proxy.test.sdc.dk/api/psd2/v2/accounts/996123252014879/transactions?dateFrom=2010-03-29T00:00:00&dateTo=2021-03-09T00:00:00&bookingStatus=Booked&withBalance=true&pstgHostTs=2021-02-02-11.09.11.137681&pstgEfctvDate=2021-02-02&cdnIdfr=0&postValueDate=2021-02-02&ecratText=&aabpiAmt=357737873.05&aabpiMaxTms=2021-04-08-07.16.38.066852"

}

}

},

"balances": [

{

"balanceType": "closingBooked",

"balanceAmount": {

"currency": "DKK",

"amount": "643908078.37"

# Fallback solution

If you as a TPP do not want to use the PSD2 API or the solution for some reason is not available, then SDC offers a fallback solution to TPP’s according to regulations in PSD2.

The fallback solution is based on scraping of the existing mobilebank APIs. These can be found at following endpoint <https://prod.smartdk.sdc.dk>. When you call you should add the reg nr. of the bank involved and the credentials of the PSU you wish to aquire data from.

As a TPP you can identify yourself in the fallback solution by either by a User Agent or a Header Identification

* Using a user agent with an ID   
  Example. TPP X (+https://www.TPPx.com/; xyz@tppx.com)
* Identifying yourself in the header with a certificate  
  Example. Fill out XTPPIdentityHeaderPresent with your public certificate  
  "eyJhbGciOiJSUzI1NiIsImtpZCI6IjUyZWI2MjFhN2I1YTkyNGRmM2E0NWUyZWQ0YzI4MWUwZDg5OGMwMzQiLCJ0eXAiOiJKV1Qif…”

If you have questions about this don’t hesitate to contact us at psd2support@sdc.dk.

# FAQ

## General

* Do I need to provide a test user with a valid SCA (NemID, BankID)?  
  Yes, otherwise you will get stuck on the redirect step. You can obtain test BankID or NemID through following process <https://www.nets.eu/developer/e-ident/eids/Pages/testusers.aspx>
* Can I use a production SCA in test?  
  No, it is not allowed to use PROD data in TEST. SCA portals will also be directed to the corresponding test environments at the SCA provider.
* I’m in the process of being approved as a TPP. Can I still get access?  
  Different rules apply for this in the different countries. SDC will evaluate each request on a case-by-case basis.

## AIS

* How many transactions are returned by the ‘Read transaction list of an account’ function?  
  As a standard then then all transactions in your date range are returned. If you wish more transactions, you can get this by changing the date range of your request. If you do not provide a date range, 30 days is automatically added as date range.
* How far back can I get transactions?  
  You can retrieve the same amount of transactions as is visible in other SDC channels. This is generally 5 – 10 years.

## PIS

* Can I do all types of transactions?  
  Yes, the SDC API supports all the same transaction types as the current eBank channels. This includes Debit/Credit transfers, Cross border, Instant, Periodic and Giro payments.
* Process for creating a payment
  1. Create payment via the correct payment create URL (ex. <https://api-proxy.test.sdc.dk/api/psd2/v1/payments/GiroNO-sepa-credit-transfers/GIRONO>)
  2. Check status is "“RCVD” using <https://api-proxy.test.sdc.dk/api/psd2/v1/payments/sepa-credit-transfers/>{{paymentID}}/status
  3. Approve payment using https://auth.sdc.dk) with following parameters /Account/Payments/{paymentId}/{paymentType}
  4. Check b is “ACSP” using <https://api-proxy.test.sdc.dk/api/psd2/v1/payments/sepa-credit-transfers/>{{paymentID}}/status

If you have questions not covered by this guide, please use the form on <http://sdcinfo.dk/tpp/>

and submit your questions to SDC.