Documentation for creditor account to account payment

API version - 1.0.0

Creditor introduction

The creditor A2A API is used for the Account Checkout service. The Account Checkout service is a flexible Open Banking payment solution that enables you (the creditor) to request a payment initiation from a consumers bank account. The Account checkout service consists of an API for payment initiation requests and payment initiation statuses, plus a debtor web app (interface) for debtor bank/account selection and payment initiation authorization.

The following payment types are supported:

1. Day-to-day payments
2. Instant payment

The following steps are involved in completing a payment:

1. Creditor POST a payment request at the A2A API
2. Creditor redirects the consumer to the Account checkout interface (can be embedded on creditors site)
3. The consumer selects bank and account and authorizes the payment
4. Creditor GET the payment status to see payment initiation resolution
5. Creditor reconcile the payment when payment is received

Creditor A2A API v1.0.0

Authentication

Before you can use *Creditor A2A API*, you must go through an on-boarding process. You will receive two keys, one set for ***Sandbox*** (Nets' external test environment) and another set for ***Production*** access, which are the pairs of unique identifiers called *Client ID* and *Client Secret*. The Client Secret should not be shared with anyone. The two keys will be needed to authenticate your application to the respective environments. The key for sandbox environment is used for a test transaction, and that of production environment for a live transaction.

You need access tokens to invoke Creditor A2A API's resources. Access tokens are passed in the HTTP header when invoking the API. The authorization server provides a Token Endpoint that you can use to generate or renew your access token. The response of the Token Endpoint is a JSON message. You extract the token for the JSON and pass it with an HTTP Authorization header to access the API.

OAuth 2.0

OAuth 2.0 is the industry-standard protocol for authorization. Read more at The OAuth 2.0 Authorization Framework, https://tools.ietf.org/html/rfc6749

Creditor A2A API currently supports OAuth 2.0 specification with confidential client type. A Confidential client is capable of maintaining the confidentiality of its credentials provided by an authorization server.

OAuth 2.0 defines four roles:

1. Resource owner: An entity capable of granting access to a protected resource.
2. Resource server: The server hosting protected resources, capable of accepting and responding to protected resource requests using access tokens.
3. Client: An application making protected resource requests on behalf of the resource owner and with its authorization.
4. Authorization server: The server issuing access tokens to the client after successfully authenticating the resource owner obtaining authorization.

At a very high-level, it is possible to break the full OAuth flow into two parts.

1. Get a token from the authorization server.
2. Use the token to access the resource server.

OAuth 2.0 defines a concept called "authorization grant" which is a credential representing the resource owner's authorization (to access its protected resources) used be the client to obtain an access token. Creditor A2A API supports Client Credentials grant type.

Token endpoint

- Url for external test environment: https://auth.pisp-test1.nets.eu/pisp-auth/oauth/token
- Url for production environment: https://auth.pisp.nets.eu/pisp-auth/oauth/token

With the Client Credentials grant type, the client can request an access token using only its credentials when the client is requesting access to the protected resource under its control. Once the client makes this request to the authorization server, it will return back an access token to the protected resource. The access token returned back to the client for Mandate API is Bearer type.

Generate access token using Client Credential grant type

1. Obtain a valid client_id and client_secret.
2. Combine the pair in the format client_id:client_secret and encode the combined string using base64. See Encode to Base64 format, https://www.base64encode.org/
3. Use the below sample cURL command to obtain the access token.

* *The access token received will have a time to live of one hour.*

```
$ curl -k -d "grant_type=client_credentials"
        -H "Authorization: Basic <Base64 encoded client_id:client_secret>"
        -H 'Content-Type: application/x-www-form-urlencoded'
         https://auth.pisp-test1.nets.eu/pisp-auth/oauth/token
```

You will receive a response similar to the following:

```
{
  "token_type": "Bearer",
  "expires_in": 2061,
  "access_token": "ca19a540f544777860e44e75f605d927"
}
```

Resource API - Creditor A2A API

Resource endpoints

- Base url for external test environment: https://api.pisp-test1.nets.eu/pisp/api/v1/
- Base url for production environment: https://api.pisp.nets.eu/pisp/api/v1/

Headers and payload

- Request data must be in JSON format. The Content-Type header value must be application/json.
- The Accept header value must be application/json.
- Response will be in JSON format.

Error message structure

Creditor A2A API uses conventional HTTP response codes to indicate the success or failure of an API request. In general: codes in the 2XX range indicate success. Codes in the 4xx range indicates data errors on the client side or client access errors, e.g. initial validation on payment request failed. Codes in the 5XX range indicate an error with Creditor A2A API's servers or access to banks APIs.

Create new payment request - POST /payments/request

Request

Submit a payment request to the account to account checkout solution. This request will generate a checkout URL which the creditor need to route its customer in order for them to complete the payment initiation process, including user giving its consent for fetching users available bank accounts, account balance and initiating the payment towards the correct bank. Once the process is finalized with the account to account checkout solution and the customer of the creditor, the Creditor A2A solution will callback the creditor to the provided callback URL in the payment request.

There are two options for payment type:

- DOMESTIC_TRANSFER - the creditor also need to specify *execution_date* >= todays date. Payment will be processed on the date mentioned in the request.
- INSTANT_DOMESTIC_TRANSFER - will be a Danish "Straksoverførsel", and the payment will be processed according to that payment type in the banks, and the creditor will not configure an *execution_date*.

There are two ways of presenting banks SCA pages by specifying the *field usage_type:*

- standalone - standalone will redirect the customer to banks own page within the same browser tab.
- embedded - embedded will keep the customer on the same page for the checkout flow, and will then open banks SCA page in new tab in the browser **.

  *** if the customer close that newly open tab with SCA prior to finalizing the SCA, the checkout process will stop, and the customer have to start over in order to finalize the payment process.*

POST /payments/request

```
"callback_url": "https://www.nets.eu",
"cancel_payment_callback_url": "https://www.nets.eu/cancel",
"payment_information": {
    "type": "DOMESTIC_TRANSFER",
        "execution_date": "2020-12-19",
    "credit_transfer_transaction": {
        "creditor_account": {
            "type": "BBAN",
            "value": "34234234234",
            "country": "DK",
            "currency": "DKK"
        },
        "instructed_amount": {
            "amount": "26.00",
            "currency": "DKK"
        },
        "remittance_information": {
            "reference": "71000000536789594",
            "unstructured": "some free text"
        },
        "debtor": {
            "contact_details": {
                "email_address": "debtorname@domain.com",
                "mobile_number": "+4512345678"
            },
            "name": "Peder Danske",
            "address_line": "Klausdalsbrovej 601",
            "post_code": "DK-2750",
            "town_name": "Ballerup",
            "country": "Denmark",
            "id": {
                "other": "some_reference"
            }
        }
    },
    "supplementary_data": {
        "additional_info": "additional info",
        "purpose": "type",
        "debtor_country": "DK",
        "language": "DK",
        "custom_ui_configurations": {
            "usage_type": "standalone"
        }
    }
}
```

Response

The *pisp_payment_request_id* is the unique reference for the payment request. This id will be used when checking the status of an actual payment. The *pisp_checkout_url* is the checkout URL where the creditor need to route its customer in order for them to complete the payment initiation process.

201 Created

```
 "pisp_payment_request_id": "9ea5690e-ccd0-41d6-a694-785a8465a279",
 "status_reason": {
     "status": "AuthorizationRequired",
     "reason": "Payment request created, Debtor to complete payment.",
     "origin": "PISP"
 },
 "pisp_checkout_url": "https://web.pisp-test1.nets.eu/pisp-backend/#/?id=9ea5690e-ccd0-41d6-a694-
785a8465a279"
```

Get the payment status - GET /payments/request/{pisp_payment_request_id}

In order for the creditor to verify a status of a payment, the get payment status endpoint will be used. The *pisp_payment_request_id* received in the response of the payment request endpoint is the id needed to verify a payment. As it is a get call, the body is empty, and standard headers to be used.

Response

200 OK


```
"data": {
    "status_reason": {
        "status": "AuthorizationRequired",
        "reason": "Payment request created, PSU to complete payment.",
        "origin": "PISP"
    }
}
```