

# Consent authorization using Sandbox

## 1. Execute "POST /v1/api/consents" endpoint.

### a) Press "Try it out" button.

**POST** /v1/api/consents Creates an account information consent resource at the ASPSP

Creates an account information consent resource at the ASPSP regarding access to accounts specified in this request.

Parameters Try it out

Name	Description
<b>X-Request-ID</b> * required string (header)	ID of the request, unique to the call, as determined by the initiating party.
<b>PSU-IP-Address</b> * required string (header)	The forwarded IP Address header field consists of the corresponding http request IP Address field between PSU and TPP.
TPP-Redirect-Preferred boolean (header)	TPP-Redirect-Preferred
TPP-Redirect-URI string (header)	TPP-Redirect-URI

### b) Fill in mandatory fields.

**PSU-IP-Address** \* required The forwarded IP Address header field consists of the corresponding http request IP Address field between PSU and TPP.  
string  
(header)

**X-Request-ID** \* required ID of the request, unique to the call, as determined by the initiating party.  
string  
(header)

c) If needed modify body and then press “Execute button”.

Request body for a consents establishing request.

Edit Value | Schema

```
{
  "access": {
    "accounts": [],
    "balances": [],
    "transactions": []
  },
  "recurringIndicator": "true",
  "validUntil": "",
  "frequencyPerDay": "",
  "combinedServiceIndicator": "false"
}
```

Cancel

Execute Clear

d) You will get consent object in JSON format as a response.

Code Details

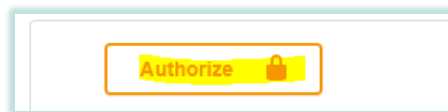
201 Response body

```
{
  "consentStatus": "received",
  "consentId": "7794b38d-0bc4-4ceb-a4c6-23394e06e642",
  "_links": {
    "scaStatus": "/v1/api/consents/7794b38d-0bc4-4ceb-a4c6-23394e06e642/authorisations/e6473b93-1ffa-4a5b-ac63-1857499ad2e3",
    "scaOAuth": "/oauth2/authorize",
    "self": "/v1/api/consents/7794b38d-0bc4-4ceb-a4c6-23394e06e642",
    "status": "/v1/api/consents/7794b38d-0bc4-4ceb-a4c6-23394e06e642/status"
  }
}
```

Download

2. As a next step created consent have to be authorized.

To start authorization process, press “Authorize” button.



3. In opened window you have to select “AISP” scope (“PISP” for payment) and input “consentId” received on 1st step.

Press “Authorize” button:

**Available authorizations** x

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.  
API requires the following scopes. Select which ones you want to grant to Swagger UI.

**ConferoAuth (OAuth2, authorizationCode)**

Authorization URL: /auth/oauth2/authorize  
Token URL: /auth/oauth2/token  
Flow: authorizationCode

client\_id:

client\_secret:

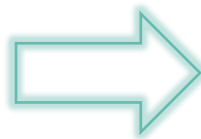
Scopes:  
 :

4. Go through Internetbank authorization process:

**Welcome to internetbank**

User ID (any 6 digits)

Password (123456)



**Welcome to internetbank**

Mobile Digipass

Authorization code (123456)

5. Select client accounts, enter authorization code and press "Send".

Select All

**Account information:**

- LT163420000000010002
- LT433420000000010001
- LT863420000000010003
- LT643420000000020002
- LT913420000000020001
- LT423420000000030001

**Balances:**

- LT163420000000010002
- LT433420000000010001
- LT863420000000010003
- LT643420000000020002
- LT913420000000020001
- LT423420000000030001

**Transactions:**

- LT163420000000010002
- LT433420000000010001
- LT863420000000010003
- LT643420000000020002
- LT913420000000020001
- LT423420000000030001

---

Authorization code (123456)

Authorization tool No. \*

6. Now Sandbox is authorized and can execute relevant endpoints:

### Available authorizations ✕

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.  
API requires the following scopes. Select which ones you want to grant to Swagger UI.

#### ConferoAuth (OAuth2, authorizationCode)

Authorized

Authorization URL: `/auth/oauth2/authorize`

Token URL: `/auth/oauth2/token`

Flow: `authorizationCode`

client\_id: `*****`

client\_secret: `*****`

[Logout](#) [Close](#)

7. As an example, we can execute “GET /v1/api/accounts” endpoint.

**GET** `/v1/api/accounts` Reads a list of bank accounts, with balances where required. 🔒

Read all identifiers of the accounts, to which an account access has been granted to through the /consents endpoint by the PSU. In addition, relevant information about the accounts and hyperlinks to corresponding account information resources are provided if a related consent has been already granted.

Parameters Cancel

Name	Description
<b>X-Request-ID</b> * required	ID of the request, unique to the call, as determined by the initiating party.
string (header)	<input type="text" value="7794b38d-0bcb-4ceb-a4c6-23394e06e642"/>
<b>Authorization</b> * required	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the related consent authorisation.
string (header)	<input type="text" value="7794b38d-0bcb-4ceb-a4c6-23394e06e642"/>
<b>Consent-ID</b> * required	This then contains the consentId of the related AIS consent, which was performed prior to this payment initiation.
string (header)	<input type="text" value="7794b38d-0bcb-4ceb-a4c6-23394e06e642"/>

Execute

Clear

## Responses

### Curl

```
curl -X GET "http://localhost:8080/resource/v1/api/accounts" -H "accept: application/json" -H "X-Request-ID: e93a71a6-7ebd-43fe-85ae-f590245c5f06" -H "Consent-ID: e93a71a6-7ebd-43fe-85ae-f590245c5f06" -H "Authorization: Bearer c26f120d-9aef-478d-9d28-4c8e21423632" -H "Swagger: True" -H "SwaggerClient: ffa75b04-670e-477a-95e6-47b93974ed57"
```

### Request URL

```
http://localhost:8080/resource/v1/api/accounts
```

### Server response

Code Details

200

### Response body

```
{
  "accounts": [
    {
      "resourceId": "LT86342000000010003",
      "iban": "LT86342000000010003",
      "bban": null,
      "msisdn": null,
      "currency": "XXX",
      "name": null,
      "product": "1000",
      "cashAccountType": null,
      "status": "enabled",
      "bic": "CONFV22XXX",
      "linkedAccounts": null,
      "usage": null,
      "details": null,
      "balances": null,
      "_links": {
        "balances": "/v1/api/accounts/LT86342000000010003/balances",
        "transactions": "/v1/api/accounts/LT86342000000010003/transactions?dateFrom=2020-01-01&dateTo=2020-06-01&bookingStatus=booked&withBalance"
      }
    }
  ]
}
```

Download