

Consent authorization using Postman

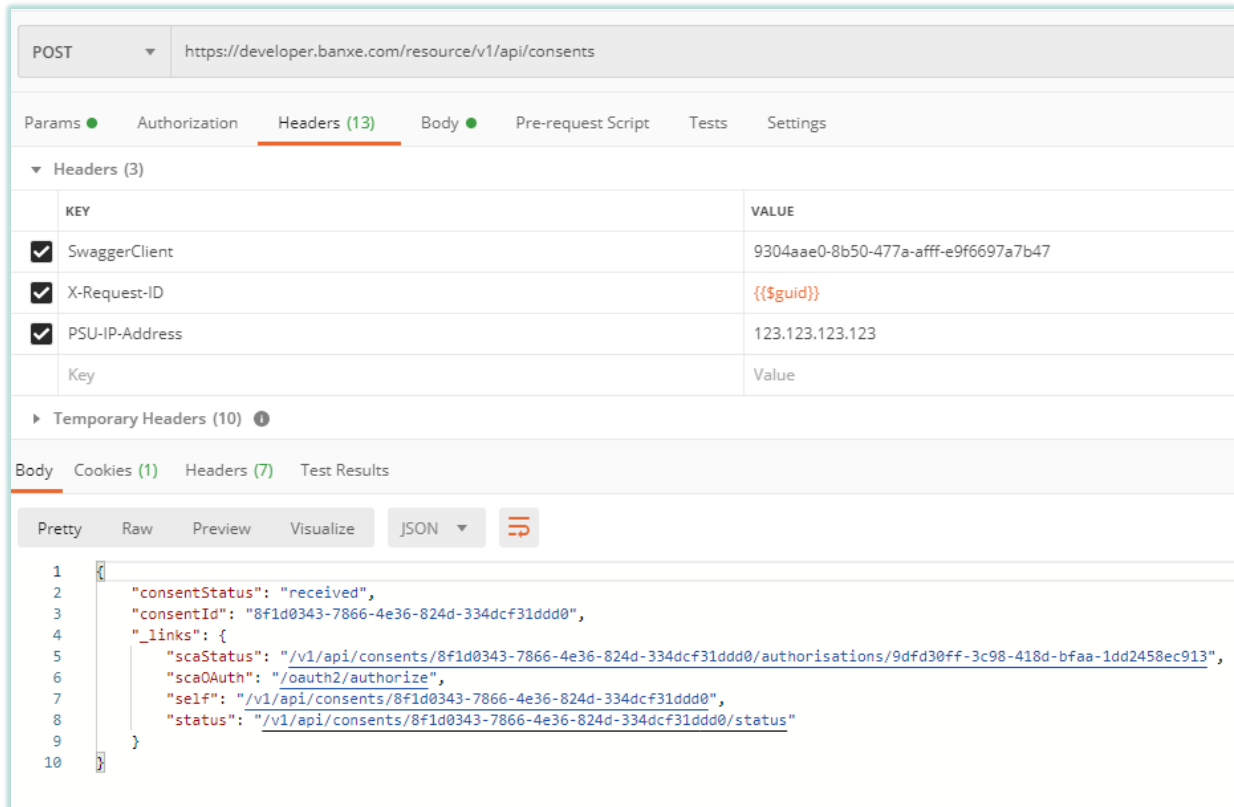
1. Execute "POST /v1/api/consents" endpoint. You will get a consent object in JSON format as a response.

Params:

- SwaggerClient = Application Key;
- X-Request-ID = ID of the request, unique to the call, as determined by the initiating party;
- PSU-IP-Address = The forwarded IP Address header field consists of the corresponding http request IP Address field between PSU and TPP.

Body:

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



The screenshot shows a Postman interface for a POST request to `https://developer.banxe.com/resource/v1/api/consents`. The Headers tab is selected, displaying three headers: `SwaggerClient` (value: `9304aae0-8b50-477a-afff-e9f6697a7b47`), `X-Request-ID` (value: `{{ $guid }}`), and `PSU-IP-Address` (value: `123.123.123.123`). Below the headers, the Body tab is active, showing a JSON response in Pretty view:

```
1  {
2    "consentStatus": "received",
3    "consentId": "8f1d0343-7866-4e36-824d-334dcf31ddd0",
4    "_links": {
5      "scaStatus": "/v1/api/consents/8f1d0343-7866-4e36-824d-334dcf31ddd0/authorisations/9dfd30ff-3c98-418d-bfaa-1dd2458ec913",
6      "scaOAuth": "/oauth2/authorize",
7      "self": "/v1/api/consents/8f1d0343-7866-4e36-824d-334dcf31ddd0",
8      "status": "/v1/api/consents/8f1d0343-7866-4e36-824d-334dcf31ddd0/status"
9    }
10 }
```

2. As a next step, created consent need to be authorized.

To start authorization process, PSU must be redirected to URL formatted according to OAUTH2 specification:

https://<URL>/auth/oauth2/authorize?response_type=code&client_id=<TPP_KEY>&redirect_uri=<REDIRECT_URI>&scope=AISP%3A<CONSENT_ID>&state=<RANDOM_NUMBER>

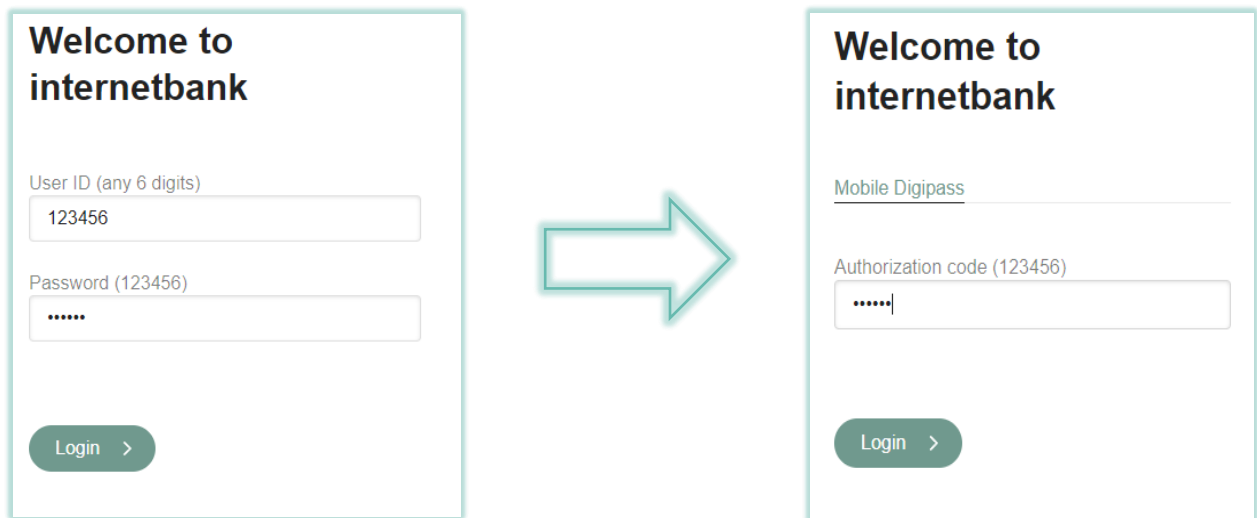
Params:

- response_type = code;
- <TPP_KEY> = Application key;
- <REDIRECT_URI> = Application redirect URL;
- <CONSENT_ID> = Consent id received in previous step;
- <RANDOM_NUMBER> = random number (request ID).

Example:

https://developer.banxe.com/auth/oauth2/authorize?response_type=code&client_id=9304aae0-8b50-477a-afff-e9f6697a7b47&redirect_uri=http://test1748.lv&scope=AISP%3A8f1d0343-7866-4e36-824d-334dcf31ddd0&state=587454

3. Go through Internetbank authorization process:



4. 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)

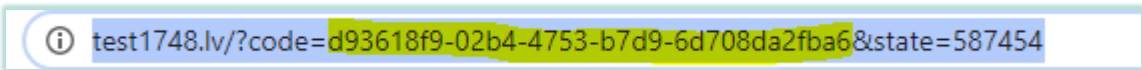
123456

Authorization tool No. *

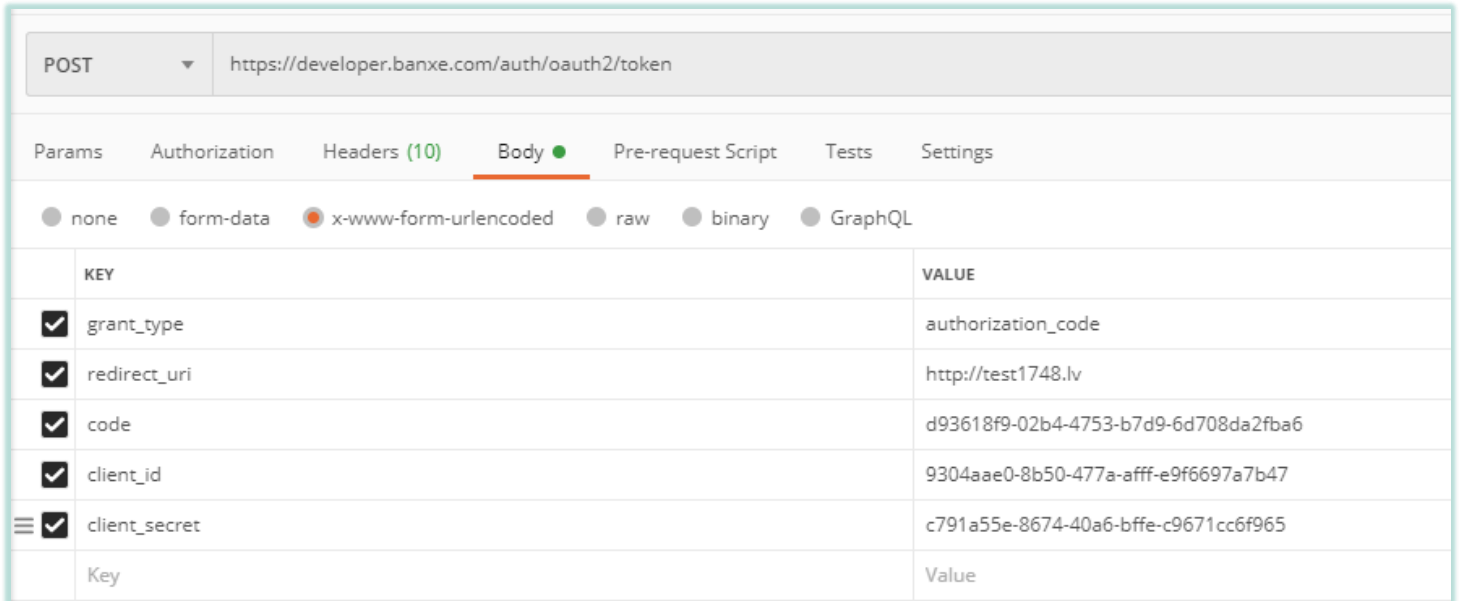
FDM1802654

Send >

5. After account selection you will be redirected to the redirect URL with code. This code will be used to get token.



6. To get token must be called “POST <https://developer.banxe.com/auth/oauth2/token>” endpoint.

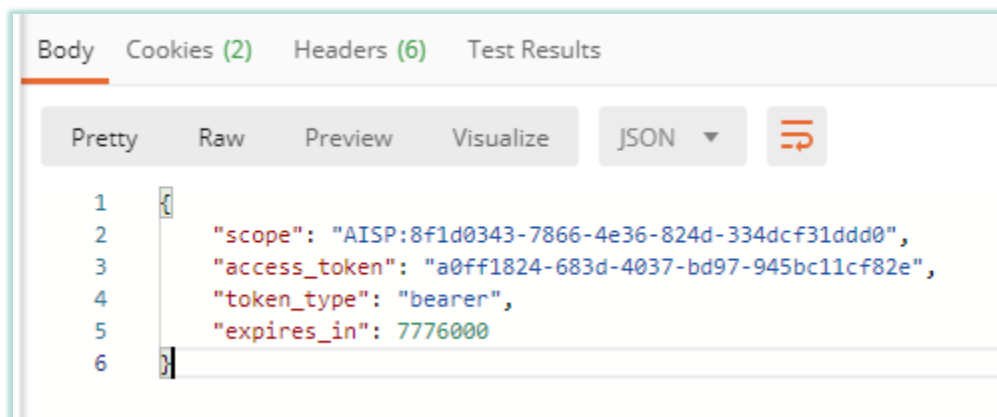


KEY	VALUE
<input checked="" type="checkbox"/> grant_type	authorization_code
<input checked="" type="checkbox"/> redirect_uri	http://test1748.lv
<input checked="" type="checkbox"/> code	d93618f9-02b4-4753-b7d9-6d708da2fba6
<input checked="" type="checkbox"/> client_id	9304aae0-8b50-477a-afff-e9f6697a7b47
<input checked="" type="checkbox"/> client_secret	c791a55e-8674-40a6-bffe-c9671cc6f965
Key	Value

Params:

- grant_type = authorization_code;
- redirect_uri = Application redirect URL;
- code = code received in previous step;
- client_id = Application key;
- client_secret = Application secret.

Response:



```
1 {
2   "scope": "AISP:8f1d0343-7866-4e36-824d-334dcf31ddd0",
3   "access_token": "a0ff1824-683d-4037-bd97-945bc11cf82e",
4   "token_type": "bearer",
5   "expires_in": 7776000
6 }
```

7. Now we can use received token to get information about accounts, balances etc.

Example of getting information about accounts:

Authorization type OAuth 2.0.

The screenshot shows the 'Authorization' tab of an API client. The 'TYPE' is set to 'OAuth 2.0'. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about...'. The 'Access Token' field contains the value 'a0ff1824-683d-4037-bd97-945bc11cf82e' and has a 'Get New Access Token' button. The 'Add authorization data to' dropdown is set to 'Request Headers'. A 'Preview Request' button is visible at the bottom left.

Params:

- Access Token = token received in the previous step.

The screenshot shows the 'Headers' tab of an API client for a GET request to 'https://developer.banxe.com/resource/v1/api/accounts?withBalance=true'. The 'Headers (3)' section is expanded, showing a table with the following data:

KEY	VALUE
<input checked="" type="checkbox"/> X-Request-ID	{{ \$guid }}
<input checked="" type="checkbox"/> Consent-ID	8f1d0343-7866-4e36-824d-334dcf31ddd0
<input checked="" type="checkbox"/> SwaggerClient	9304aae0-8b50-477a-afff-e9f6697a7b47
Key	Value

Below the table, there is a section for 'Temporary Headers (9)' with an information icon.

Params:

- SwaggerClient = Application Key;
- X-Request-ID = ID of the request, unique to the call, as determined by the initiating party;
- Consent-ID = Consent ID received in 1. step.

Response:

```
Body Cookies (1) Headers (5) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "accounts": [
3     {
4       "resourceId": "LT863420000000010003",
5       "iban": "LT863420000000010003",
6       "bban": null,
7       "msisdn": null,
8       "currency": "XXX",
9       "name": null,
10      "product": "1000",
11      "cashAccountType": null,
12      "status": "enabled",
13      "bic": "CONFLV22XXX",
14      "linkedAccounts": null,
15      "usage": null,
16      "details": null,
17      "balances": [
18        {
19          "balanceAmount": {
20            "currency": "EUR",
21            "amount": "2856950.55"
22          },
23          "balanceType": "cleared",
24          "creditLimitIncluded": false,
25          "lastChangeDateTime": "2020-06-02T13:04:57",
26          "referenceDate": "2020-07-30"
```