Fyorin Open Banking API Integration

This document provides a guide to TPPs on how to integrate using APIs and perform actions on behalf PSUs.

- Test server: `dev.fyorin.com`
- Live server: `live.fyorin.com`

Login

POST `<SERVER>/api/psd2/login`

```
{
  "email": "<email>",
  "certAndTimestamp": {
    "certificate": "<PEM encoded certificate base-64 without newlines>",
    "encryptedTimestamp": "<UTC timestamp encoded as 8 bytes bin-endian
encrypted with private key>"
  }
}
```

The `encryptedTimestamp` parameter is the time in milliseconds since January 1, 1970 00:00:00 UTC (within 1 minute) encoded as 8 bytes big-endian (e.g. time `0x1213141516171819` would be encoded as bytes `{0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19}`), encrypted using the RSA private key and BASE64 encoded.

Response will have the following structure:

```
{
  "auth": {
    "token": "<Bearer token to be used in Authorization header for
subsequent calls>"
  }
}
```

Request Access

To perform actions on behalf of a PSU, first a TPP needs to request access by creating a request for access and redirecting the PSU to the consent url, which will show the request information the PSU and allow them to accept or reject the access.

Remove non-applicable roles. Specify `overrideId` if you want to override a prior access request (e.g. first request was for AISP, second one overrides to request both AISP and PISP)

POST `<SERVER>/api/business/shares/requests/_/create`

```
{
  "featureRole": {
    "ACCOUNTS": "AISP",
    "PAYMENTS": "PISP"
  },
  "overrideId": "<OPTIONAL prior access request to override>"
}
```

Response header `generated-id` holds the generated request id.

The consent URL where to redirect PSU is `<SERVER>/business/request/consent/{request_id}`

Calling APIs

Once access is granted, the TPP can impersonate the PSU by calling login followed by

POST `<SERVER>/api/auth/current/impersonate`

```
{
  "shareRequestId": "<generate-id from the prior call>"
}
```

Response will have the same structure as login:

```
{
  "auth": {
    "token": "<Bearer token to be used in Authorization header for
subsequent calls >"
  }
}
```

This token represents this specific PSU impersonated by the TPP. The token from the previous call is still valid and can be used to impersonate other PSUs in the same manner.

In case the access request has not been accepted, the result is HTTP error 409 with the following structure:

```
{
  "errorCode": "UNSUCCESSFUL"
}
```

The following APIs may be called:

- AISP
  - POST `/accounts/get` returns the list of accounts (+ filter)
  - POST `/accounts/{id}/get` returns the balances of a specific account
  - POST `/accounts/{id}/references` return the bank account number of the account
  - POST `/accounts/{id}/statement` returns the statement of the account
- PISP
  - POST `/payments/_/create` creates a payment

See `<SERVER>/openapi/` for details.

Testing

A postman script is provided, for help with integration. To use it, set the following environment variables:

- `tpp_email`
- `tpp_certificate`
- `tpp_private_key`
- `psu_email`
- `psu_password`

The scripts use some test endpoints which are not available on the live system, to help verify your implementation or for bypassing manual processes. The test system does not provide any guarantees of uptime or data retention and data is periodically purged.

To test, you

The `tpp_private_key` variable is used by a test endpoint to encrypt the timestamp (you can use this to test correct implementation). **DO NOT** use with your real private key! Instead, generate a test certificate & private key as described above.

Should you encounter any issues, kindly send an email to `psd2-support@fyorin.com`