User integration using OAuth

User-level integrations are make using OAuth grants. A user of a 3th-party system is bound to a Bold user and requests permission to access anything the bold user can access. To implement a user-level integration, the following is needed:

- a client id `CLIENT_ID`
- a client secret `CLIENT_SECRET`
- a redirect uri `REDIRECT_URI`

The client id and client secret are provided by Bold on request. The redirect uri is part of the 3th party system and is to be provided to Bold.

Step 1: Call Bold authorization URL

The 3th party client application should call the Bold authorization URL. This will make sure the Bold user is asked whether the 3th party is allowed to access the Bold system.

`https://auth.boldsmartlock.com/?client_id=CLIENT_ID&response_type=code&state=STATEDATA&redirect_uri=REDIRECT_URI`

The `state` parameter can contain any data and will be passed back to the 3th party system later. The 3th party system typically uses this data to validate the granted permission and to relate it to a specific request.

Calling the authorization URL will either open a login screen in which the user should login with his Bold credentials, or it will simply open the Bold app when it is installed using 'deep linking'.

Step 2: Implement the redirect URI

When the user has granted permission to the 3th party, the Bold platform calls the redirect URI. This URI should be implemented by the 3th party application server, or through a 'deep link' of the 3th party app.

The called URI will have two additional query parameters:

- `state=STATEDATA`: containing the same state data that was passed to the authorization URL, and
- `code=CODE`: containing the authorization code needed for logging in.

In case the user has denied the permission, the redirect URI is still called, except with an error instead of the authorization code.

- `state=STATEDATA`: containing the same state data that was passed to the authorization URL, and
- `error=access_denied`: indicating that the permission was denied.

Step 3: Login the user

With the authorization code `CODE` that was provided as a parameter to the redirect uri call, the 3th party system can login the user with the Bold system. One can either use the API or the SDK.

Step 3a: Login using the API

The Bold OAuth API provides the following API to login using an authorization code:

```
POST https://api.boldsmartlock.com/v2/oauth/token
```

The following form fields should be provided, as of the OAuth v2 spec:

```
grant_type     authorization_code
code           CODE
redirect_uri   REDIRECT_URI
client_id      CLIENT_ID
client_secret  CLIENT_SECRET
```

The returned data looks like this example:

```
{
  "access_token": "03c64166-2c09-456d-ad7e-c1f3a6969b0c",
  "refresh_token": "38a32770-a799-49a2-b631-096b7ad8ece0",
  "token_type": "Bearer",
  "expires_in": 86400
}
```

The `access_token` is used to make API calls. USING THE HEADER Authorization 'Bearer $access_token' XXXXX The `expires_in` value tells how many seconds the token is valid. To prevent expiration, tokens should be periodically refreshed. This is done using the same API, but with the following form parameters:

```
grant_type     refresh_token
refresh_token  REFRESH_TOKEN
client_id      CLIENT_ID
client_secret  CLIENT_SECRET
```

After a refresh, both the former `access_token` and `refresh_token` are no longer valid.

Step 3b: Login using the SDK

When the `BoldDeviceSDK` is used, it suffices to call `loginWithAuthorizationCode` as shown below. When succeeded, an authorization token is returned that can be used to call the Bold API. The SDK will automatically refresh access tokens when needed.

```
BoldDeviceSDK.loginWithAuthorizationCode(CLIENT_ID, CLIENT_SECRET,
REDIRECT_URI, CODE)
```

Organization integration using Access Tokens

With an organization integration, 3th party systems are given permissions to everything that is managed by a Bold organization. The 3th party should request a Bold organization and an admin account on that organization prior to integrating. With the integration, the 3th party can create and manage users, groups, shares, etc. Additionally, the 3th-party system can request access authorization for a particular user. This is needed for a 3th party end-user app accessing the Bold API.

Step 1: Create an access token

A 3th-party system accesses the Bold API using an **Access Token**. Access token are managed by organization admins and should not be confused with an `access_token` used by OAuth. An access token has an id and secret which are used to access the API. Note that the secret is provided **only once** when creating an access token.

Step 1a: Create an access token in the portal

Login to the portal as an organization user and create an access token on the 'integrations' page. ***Note that this page has not yet been implemented so please refer to step 1b***.

Step 1b: Create an access token using the API

An access token can also be created through the Bold API directly. To create an access token, one must have a valid OAuth access. See the Access Token API.

```
POST /v1/accounts/ACCOUNT_ID/access-tokens
```

The response contains the access token's id and secret:

```
{
"accessTokenId" : String, UUID 4 formatted (e.g. "a426e157-8a5c-456a-8865-
bec6394867ab"),
"accessTokenSecret" : String,
"clientInstanceId" : Number, record primary key,
"accountId" : Number, record primary key,
"organizationId" : Number, record primary key,
"description" : String,
"dateExpiration" : Date/time with a timezone (ISO 8601, e.g. "2022-06-
23T13:53:02+02:00")
}
```

Step 2: Access the API

With the access token's id and secret one can access the API using basic authentication:

`Authorization 'Basic BASE64_OF_accessTokenId:accessTokenSecret'`

Contrary to regular OAuth access, this token remains valid indefinitely and does not need to be refreshed periodically. It is therefore the integrator's responsibility to keep the access token information secret and to request a new one (step 1) when deemed necessary.

Step 3: Create users and other items

The 3th party system can create users in the Bold platform using User Management API:

```
POST /v1/users
{
  "phone": string
  "email": string
  "managed": true
}
```

Notice that the users should be marked `managed`. This flag indicates that management is under control of a 3th-party system and enables, for example, that the 3th-party system can request access authorization for that user. The returned information contains the id of the user that can be used by the 3th party system for bookkeeping:

```
{
  "userId": number
}
```

Step 4: Request access for users

In order to let a user of a 3th party system or app access our API, for example using the Bold Device SDK, the 3th party platform requests an authorization code from the Bold platform and sends this to the user app. The user app requests an OAuth access token using this authorization code, and refresh this token daily, as prescribed by OAuth. The Bold Device SDK will do this for you.

```
POST /v2/integrations/CLIENT_ID/authorization
Header Authorization 'Basic BASE64_OF_accessTokenId:accessTokenSecret'
{
  {
    "clientId": CLIENT_ID,
    "responseType": "code",
    "scope": "platform",
    "allowed": true
  }
}
```

The returned information contains the authorization code needed to authenticate the user:

```
{
"code": CODE,
"clientId": CLIENT_ID,
"expiration": "2023-04-06T07:10:39.694749Z"
}
```

This code can be used as described in step 3 of the 'User integration' documentation.