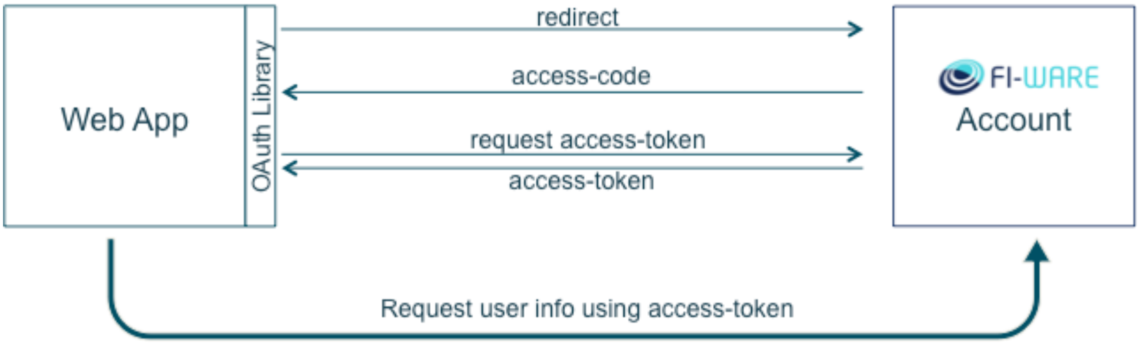
FI-WARE OAuth2- IDM authentication tutorial

1. Oauth2 implementation

First step is the implementation of OAuth2 [http://tools.ietf.org/html/rfc6749] protocol in your web app. The message flow between your web application and IDM account server should be:



In order to implement this flow you can use an OAuth2 library. Here [http://oauth.net/2/] you will find implementations for PHP, Cocoa, iOS, Java, Ruby, Javascript, Python …

We have developed a Node.js example [https://github.com/ging/oauth2-example-client] that you can download and test following these instructions:

1. Software requirements:

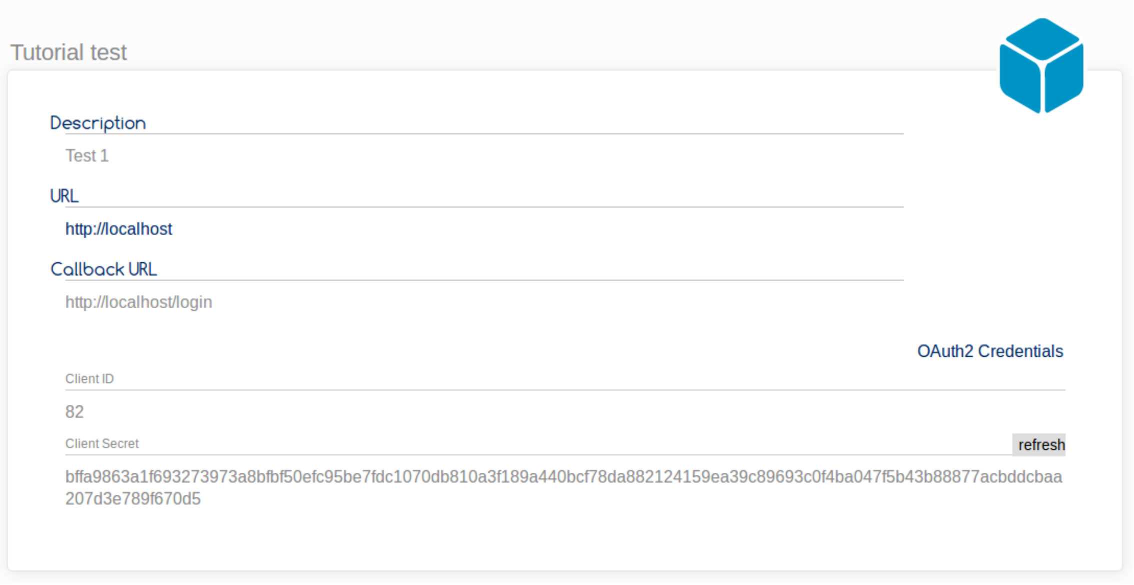
   nodejs
   npm

2. Install the dependencies:

```
npm install
```

3. Register your application in http://account.lab.fi-ware.eu

4. Configure OAuth2 credentials (ClientID and Client Secret) in config.js file. You will find them in your IDM account:



Tutorial test

Description

Test 1

URL

http://localhost

Callback URL

http://localhost/login

OAuth2 Credentials

Client ID

82

Client Secret

bffa9863a1f693273973a8bfbf50efc95be7fdc1070db810a3f189a440bcf78da882124159ea39c89693c0f4ba047f5b43b88877acbddcbaa207d3e789f670d5

5. Start example server

```
sudo node server
```

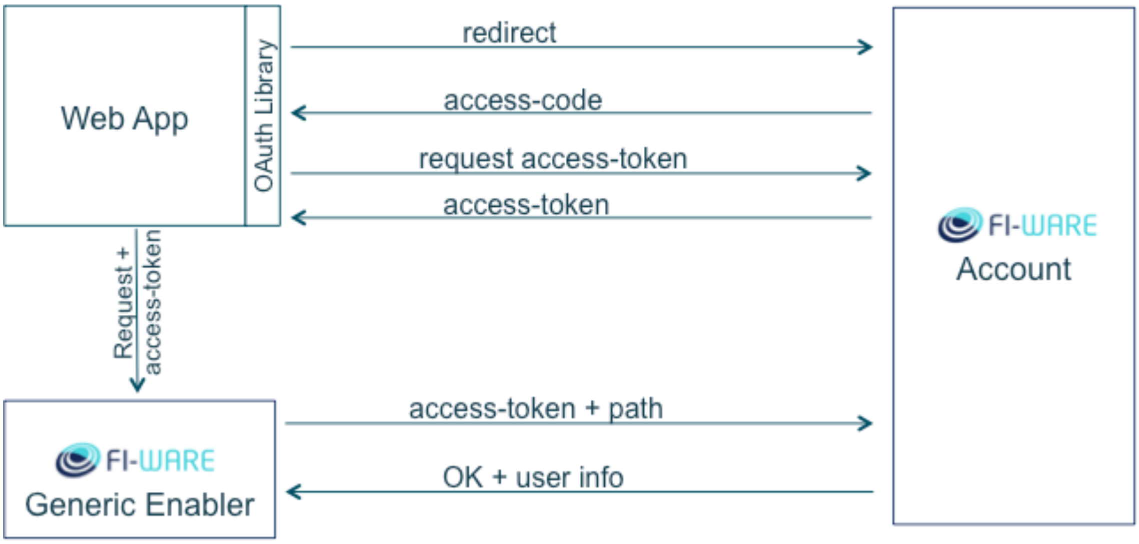
6. Connect to http://localhost to try the example

* Connect to http://localhost/logout to delete session cookies once you have logout in IDM portal

2. Sending requests to a FI-WARE Generic Enabler

If your web app is going to use a FI-WARE GE service you have to authenticate the requests.

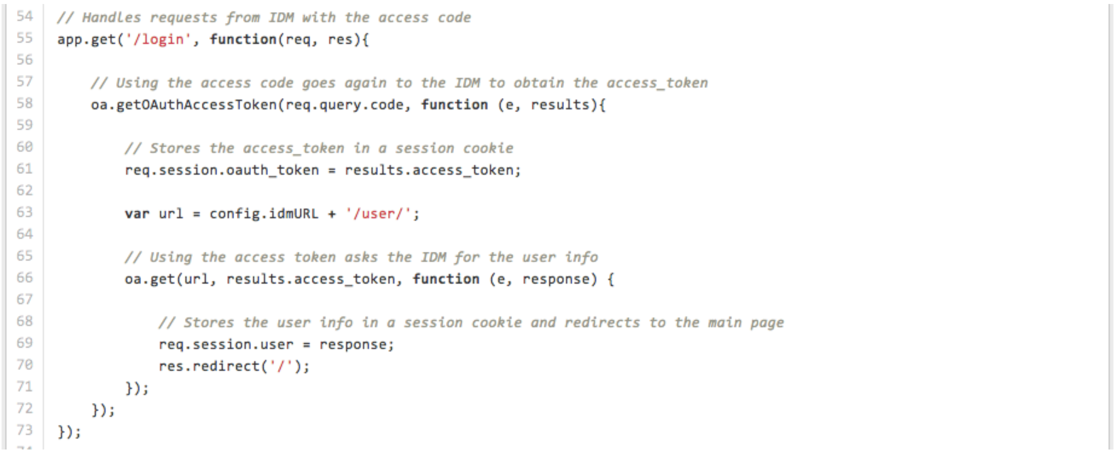
Once you have obtained an Oauth2 access-token you have to include it in your client requests in order to authenticate them. This way the GE will check your token with the IDM and decide if your requests are valid. The architecture of this flow is:



So for include the access-token in your requests you have to do:

```
GET https://GE_URL HTTP/1.1
Host: GE_hostname
X-Auth-Token: access_token
```

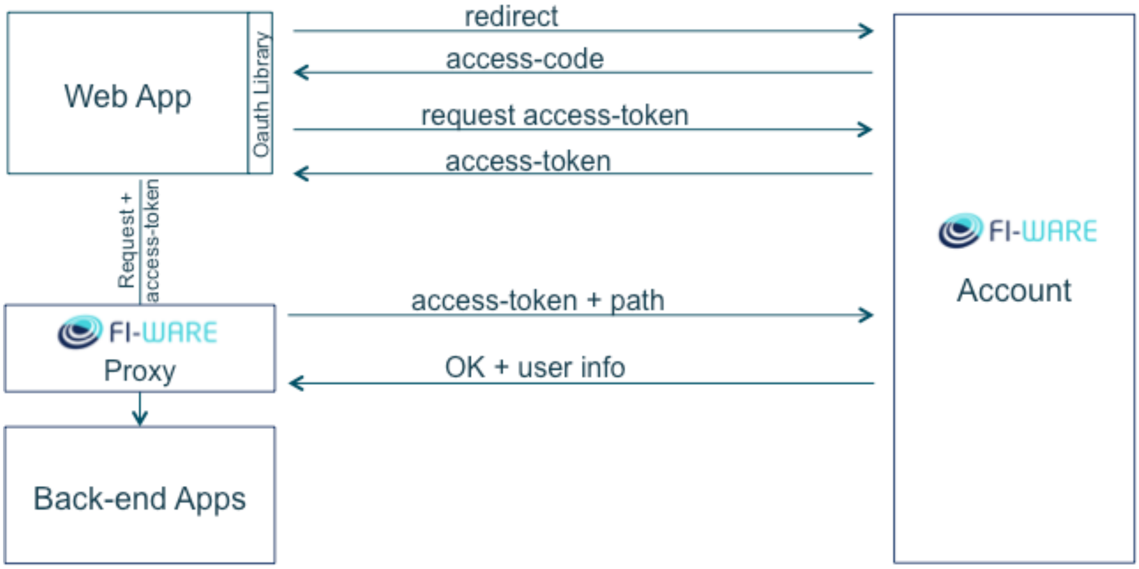
For instance, using our OAuth2 example explained previously, you can get the access_token in this code fragment (line 61):



```
54  // Handles requests from IDM with the access code
55  app.get('/login', function(req, res){
56
57      // Using the access code goes again to the IDM to obtain the access_token
58      oa.getOAuthAccessToken(req.query.code, function (e, results){
59
60          // Stores the access_token in a session cookie
61          req.session.oauth_token = results.access_token;
62
63          var url = config.idmURL + '/user/';
64
65          // Using the access token asks the IDM for the user info
66          oa.get(url, results.access_token, function (e, response) {
67
68              // Stores the user info in a session cookie and redirects to the main page
69              req.session.user = response;
70              res.redirect('/');
71          });
72      });
73  });
```

3. Authenticating requests to your Backend Service

You can also develop backend applications using FI-WARE Account authentication. To do so you should use our FI-WARE PEP Proxy that allows you to validate the requests before sending them to your server:



You can download the proxy here [https://github.com/ging/fi-ware-pep-proxy]. The instructions to install and use it are:

Installation

1. Software requirements:

    - nodejs
    - npm

Note: Both can be installed from (http://nodejs.org/download/)

2. Clone Proxy repository:

```
git clone https://github.com/ging/fi-ware-pep-proxy.git
```

3. Install the dependencies:

```
cd fi-ware-pep-proxy/
npm install
```

4. Configure app host in config.js file.

```
config.app_host = 'www.google.es'; //Hostname to forward authenticated requests
config.app_port = '80';            //Port where the HTTP server is running
```

5. Start proxy server

```
sudo node server
```

How to use

Requests to proxy should be made with a special HTTP Header like in the previous example: X-Auth-Token. This header contains the OAuth access token obtained from FI-WARE IDM GE.

```
GET / HTTP/1.1
Host: proxy_host
X-Auth-Token:z2zXk...ANOXvZrmvxvSg
```

To test the proxy you can generate this request running the following command:

```
curl --header "X-Auth-Token:z2zXk...ANOXvZrmvxvSg" http://proxy_host
```

Once authenticated, the forwarded request will include additional HTTP headers with user info:

```
X-Nick-Name: nickname of user in IDM
X-Display-Name: display name in IDM
```