Scenario

Use OAUTH with a custom authentication in an enterprise context.

We have to build a mobile application for an enterprise with hundreds of employees. We are not in a public context. We have just to check the identity of employees to be sure they get access to their scope only.

We assume that the *client* is a trusted application

The code described in the post is available on Github.

Contenu Masquer

Reminders

ORDS provides support for OAUTH through three clients types:

- Client Credential
- Authorization Code
- Implicit Grant

The two last are shaped for identifying a physical user, not the first one.

In our scenario, we want authenticate a physical user. Therefore, we want use either *Authorization Code* or *Implicit Grant* in order to identify the user.

In these cases, ORDS provides a generic authentication module and it seems that this

module can’t be customized. The only user directories supported are:

- File system
- APEX User directory
- Tomcat or Weblogic user directory

What about a scenario where we want completely custom the authentication process, by relying on a Ldap ou DB directory ?

This is the use case which inspires this post.

Overall approach

We decide to build a « new » kind of oauth flow, an hybrid between *Client credentials* and *Implocit Grant*.

We assume that the authentication is not based on the standard ORDS/OAURH authentication but has been build by a completely different method, whatever the method.

Initial setup

At ORDS level, we create a new *client* with the « *Client Credentia*l » type.

We create a REST module `eu.gpmfactory.custoauth` which handles the authentication process and the generation of a valid token.

The cinematic is as following:

If a token is not set yet, the client redirect to a REST endpoint (`login`) which send an authentication form. The client must send the Client ID in parameter, as for a implicit flow scenario.

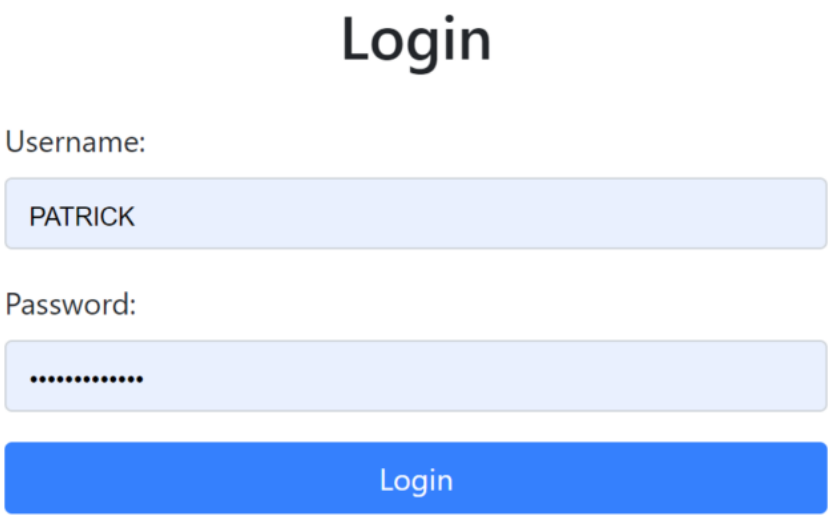
The user fills the credentials and submit the form to `check` endpoint.

The REST APi receive the credentials as long with the Client ID.
The REST API calls the authentication module which checks password.

If ok, the module calls the URL that generate a valid token for a « client Credential » client type then sends back the token to the client.

In this scenario, there is no code conversion into a token whici is delivered directly, instead (as *Implicit flow*)
We assume that the client is a trusted application.

Following is the custom login page.



In our case, Login page is provided by a GET endpoint (login) which generates all the HTML markup in order to display the login page.

Credentials are sent to a POST endpoint (auth) along with the client_id.
A function in a package (OAUTH_PKG) checks if the credentials ar corrects, checks the validity of the client_id then asks for a token as for a « client credentials » flow.
The client secret is read directly form the ORDS dictionary.

It’s possible to add a redirect url to the client definition in order to mimic an « implicit grant » flow or the the token can be sent directly in a payload to the app

Drawbacks

Not possible to use a refresh token.