Lab 12: Customize the OpenIDM Self Service User Interface

In this lab you customize the OpenIDM self-service user interface by rebranding pages, applying a new theme, and creating a custom template.

Objectives

- Customize branding and locale
- Create and deploy a custom template

Requirements

- OpenIDM installed and running.
- A text editor suitable for editing json and HTML files.

References

The following references in the OpenIDM Integrator's Guide were used in the development of this lab and might be helpful for future reference:

- Chapter 4.4, Customizing a UI Template.
- Chapter 4.8, Customizing the UI.
- Chapter 4.9, Changing the UI Theme.

Exercise 1: Prepare Your System for Customization

In this exercise, you change the branding for the Self Service UI.

Objectives

In this exercise you will:

- Prepare your environment for customization
- Create the extension directory
- Explore Bootstrap themes (optional)

Detailed Tasks

Task 1: Install a new version of OpenIDM for the Self Service customization exercises

1. Shut down any running instance of OpenIDM by entering the shutdown command in the Felix OSGI Console.

2. Create a directory named /opt/selfservice then change to that directory:

```
$ cd /opt
$ mkdir selfservice
$ cd selfservice
```

3. Unzip the openidm-4.0.0.zip file and then start OpenIDM:

```
$ unzip /opt/forgerock/software/openidm4.0.0.zip
$ cd openidm
$ ./startup.sh
```

4. Monitor startup progress in the Felix OSGI Console.

Task 2: Prepare the Self Service extension directory

Customizations are stored and modified in the extension directory for your Openidm ui/admin and ui/selfservice directories. Naming is important because the presence of an extension directory in ui/admin or ui/selfservice tells OpenIDM where to look for customizations.

1. In a new terminal window, copy the ui/selfservice/default directory to a new directory named extension.

```
$ cd /opt/openidm/ui
```



```
$ cp -r selfservice/default/. selfservice/extension
```

2. Browse the contents of the selfservice/extension directory and become familiar with what's there. These are the default resources that are now available for your customization.

3. Copy the resource zip file you will use for your customizations the to a working directory and unzip it. Explore the contents there.

```
$ mkdir /opt/selfservice
$ cd /opt/selfservice
$ unzip /opt/forgerock/labfiles/CustomizeUI.zip
```

This creates a customizeUI directory in your /opt folder that contains logo and theme files.

Task 3: Use Google Developer Tools to monitor customization (optional)

The Google Developer tools listed here are optional but may prove helpful in customizing OpenIDM.

1. This task uses Google Developer Tools so you will need the Chrome browser installed in your system. It is already installed on the classroom learning environment. To start:

```
$ cd /opt/forgerock/chrome
$ ./chrome &
```

2. Open the OpenIDM Self Service UI and select More Tools/Developer tools and monitor interaction with OpenIDM:

   a. Use the Network tab to monitor the interaction between the client requests and server responses

   b. Disable the cache to avoid caching issues while testing

   c. Use the "Save as HAR with Content" option to diagnose the interaction

   d. Use options to change to the network speed and device size to observe how the user interface reacts

Exercise 2: Customize the UI Theme

You can customize the UI theme by changing branding, applying a new theme for login, and applying a new theme to a OpennIDM project.

Objectives

In this exercise you will:

- Change the logos
- Change the UI language
- Change account labels

Detailed Tasks

Task 1: Change the logos

1. Copy the theme logo your files to your selfservice/extension/images directory and note their file names. Do not overwrite the existing files.

```
$ cd /opt/selfservice/customizeUI
$ cp *.png *.jpg
../openidm/ui/selfservice/extension/images
```

2. List the files in the selfservice/extension/images directory to verify the copy:

```
$ ls ../openidm/ui/selfservice/extension/images
```

3. Open /opt/openidm/conf/ui-themeconfig.json:

```
$ cd ../openidm/conf/
$ gedit ui-themeconfig.json
```

4. Change the logo value:

```
"settings" : {
  "logo" : {
    "title" : "RockTheme",
     "alt" : "RockTheme"
```

5. Make a similar change to specify the loginLogo:

```
"loginLogo" : {
  "src" : "images/logoFR.png",
"title" : "RockTheme",
"alt" : "RockTheme",
```



```
"height" : "99px",
"width" : "92px"
```

6. Finally, change the footer:

```
"mailto" : "info@rocktheme.com"
```

7. To test, open the Self Service UI. You should see the new login logo and footer. Once you log in, you should see the new banner. As you make changes, you can simply hit the browser's reload button to see the latest change.

Task 2: Change the UI language

In this task you change two properties in the UI locale to French (fr): "Remember my username" and "Copyright". For the sake of time you will change only those two properties but if you have time feel free to translate other properties.

1. Copy the the translation.json file in your selfservice/extension/locales/en directory to a directory named fr:

```
$ cd
/opt/selfservice/openidm/ui/selfservice/extension/lo
cales
$ cp -R en fr
```

2. Open the translation.json file in your extension/locales/fr directory with a text editor such as gedit. You should still be logged inot the locals directory from step 1:

```
$ cd fr
$ gedit translation.json
```

3. Search for the LoginTemplate property and use Google Translate to translate all the strings to French. For example:

```
"loginRemember" : "Remember my username",
```

4. Change the value to: "Mémoriser mon nom d'utilisateur":

```
"loginRemember" : "Mémoriser mon nom d'utilisateur",
```

5. When done editing the LoginTemplate properties, earch for **copyright**:

```
"copyright" : "Copyright 2010-15 ForgeRock AS.",
```

6. Change the value to:

```
"copyright" : "Droits d'auteur 2010-16 RockTheme
AS.",
```

7. Change the lang property in your projects conf/ui-configuration.json to:

```
"lang" : "fr",
```

8. When done making your changes and saving the files, log into OpenIDM Self Service. You will see the new messages.

Exercise 3: Customizing user self-service screens

In this exercise you change the UI theme for a new instance of OpenIDM.

Objectives

- In this exercise you customize the following Self Service screens:
  - User Registration
  - Password Reset
  - Forgotten Username

Detailed Tasks

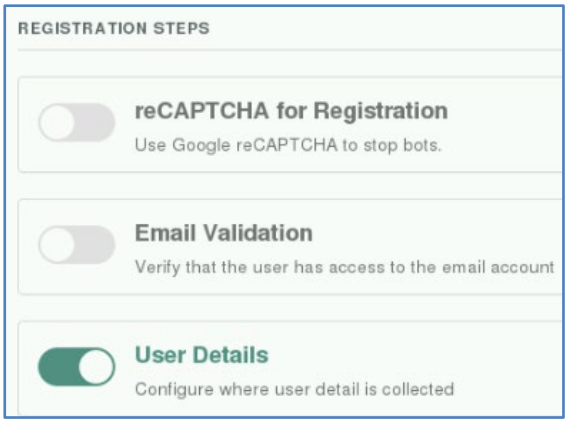
Task 1: Customize the Registration Screen Template

Customers see a default registration when they try to register. In this task you add an Employee ID field to that form.

1. If you are working on a new installation of OpenID, you need to enable User Registration.

   a. As the admin user, go to Configure and select the Enable button for User Registration.

   b. By default enabling user registration triggers a registration email. You need to turn that off by deactivating Email Validation. The screen should look like this:



   c. Log out and log back into Self Service. You should see the self-registration link at the bottom of the screen.

2. Open the userDetails-initial.html:

```
$ cd /
/opt/selfservice/openidm/ui/selfservice/extension/te
mplates/user/process/registration
```

3. With a text editor open userDetails-initial.html.

4. In this example, add an EmployeeID field:

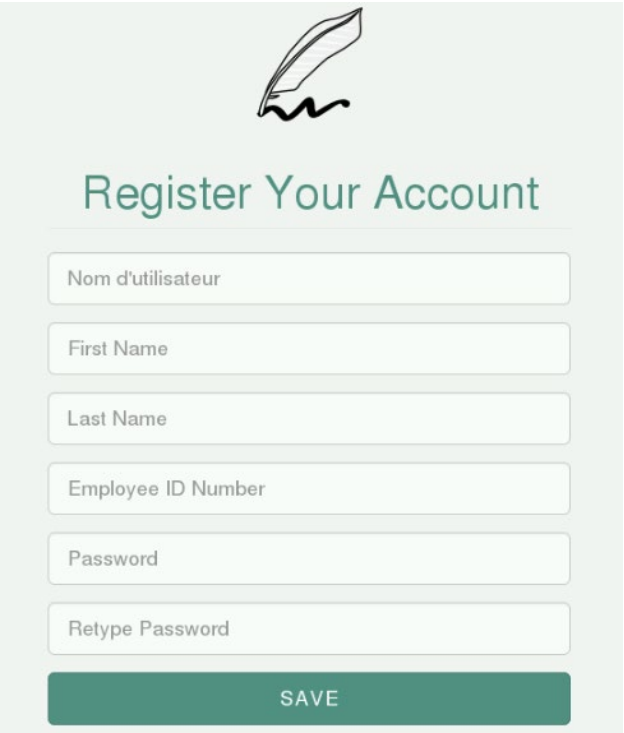
   a. Create a new form-group stanza for that number. For this procedure, the stanza appears after the stanza for Last Name (or surname) sn:

```
<div class="form-group">
    <label class="sr-only" for="input-
employeeNum">{{t 'common.user.employeeNum'}}</label>
    <input type="text" placeholder="{{t
'common.user.employeeNum'}}" id="input-employeeNum"
name="user.employeeNum" class="form-control input-
lg" />
</div>
```

5. Edit the translation.json file in your extension/locales/fr directory. This is a large file and you need to locate the correct location for the Employee ID Number label. Search for givenName – there should be only one instance. Then add the following after sn:

```
...
"givenName" : "First Name",
"familyName" : "Family Name",
"sn" : "Last Name",
"employeeNum" : "Employee ID Number",
"basicInfo" : "Basic Info",
...
```

6. Once the userDetails-initial.html and translation.json files have been edited and saved, login into Self Service and click the self-registration link at the bottom of the page. You will see the Employee ID field.



Task 2: Customize the Password Reset and Forgotten Username forms

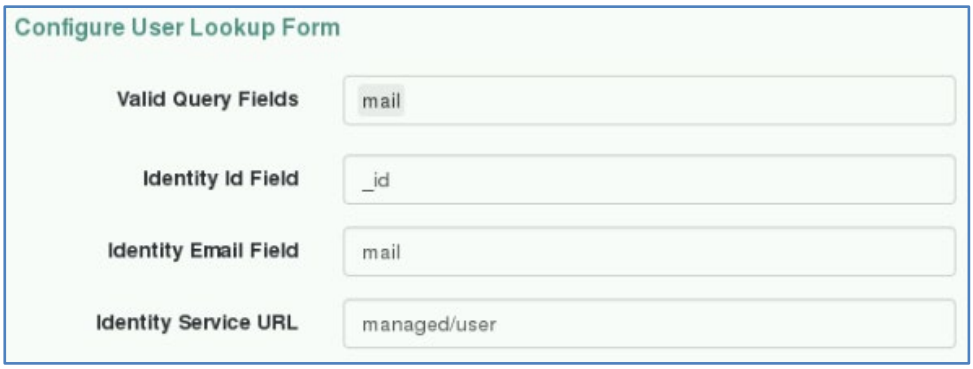
In this task you modify the Valid Query Fields that are used for password reset and forgotten username in Self Service.

1. Enable Password Reset and Forgotten Username as the Openidm Administrator in the Configuration menu.

2. As the OpenIDM Admin user, log into the Administration UI and select Configuration/Forgotten Username.

3. Disable the Email username button but leave Display Username enabled.

4. Select User Query Form and remove givenName and sn from the Valid Query Fields field. Just leave mail.



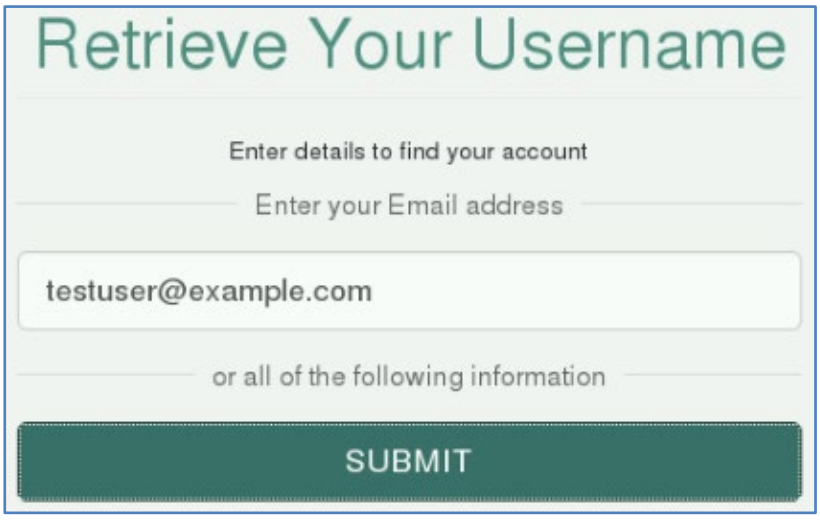
5. Open the ui/selfservice/extension/user/process/username/userQuery-initial.html.

6. Comment out the givenName and sn form groups:

```
<!--div class="filter-group filter-and">
  <div class="form-group">
    <label class="sr-only" for="input-givenName">{{t
'common.user.givenName'}}</label>
    <input type="text" placeholder="{{t
'common.user.givenName'}}" id="input-givenName"
name="givenName" class="form-control filter-value
input-lg" />
  </div>

  <div class="form-group">
    <label class="sr-only" for="input-sn">{{t
'common.user.sn'}}</label>
    <input type="text" placeholder="{{t
'common.user.sn'}}" id="input-sn" name="sn"
class="form-control filter-value input-lg" />
  </div>
</div-->
```

7. In your browser go to the Self Service login page. Click Forgot Username and enter testuser@example.com (the user you created earlier)



The username is returned:

8. Make a similar change for the Password Reset function, using the file ui/selfservice/extension/user/process/reset/userQuery-initial.html.