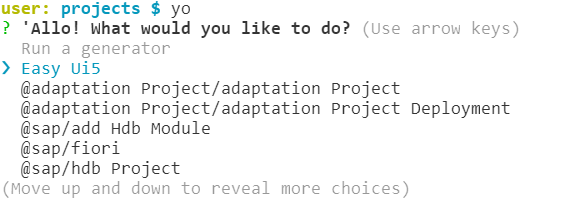
First of all. connect SAP BAS to your CF subaccount

1. Creating YEOMAN project

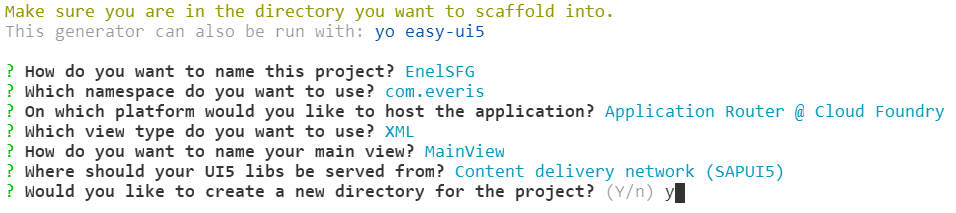
Open Terminal and enter on your path. Ex: cd project

To create a YOEMAN project type “yo” and select “Easy Ui5”. If you don’t have this option you need to install this. To do this, type “npm install -g yo generator-easy-ui5”.

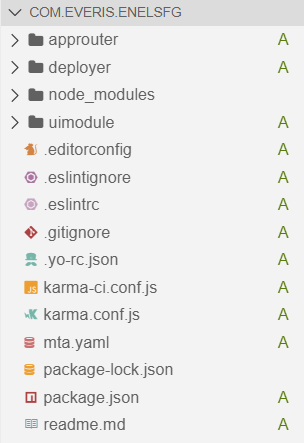
Now you should have this result:



And then follow this instruction:



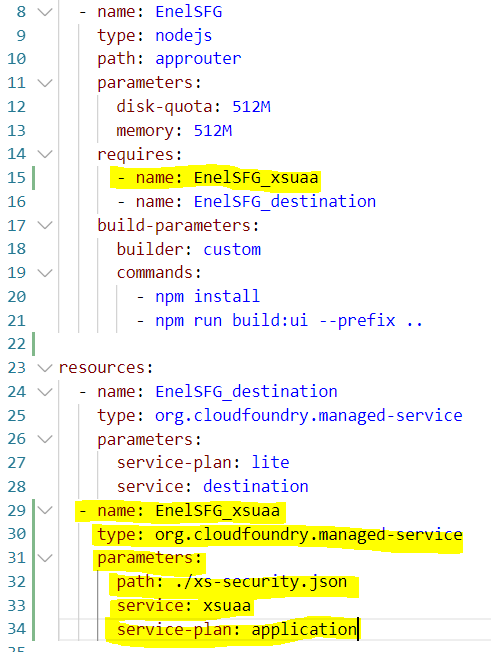
At this point you should have a new folder generated form YOEMAN:



1. Add authentication and authorization

At this point you need to modify the mta.yaml and xs-app.json files to enable authorization and authentication. And then create the xs-security.json file to define roles.

MTA.YAML (add the highlighted lines)



XS-APP.JSON (inside approuter folder): the scope parameter (role) will be defined inside the xs-security.json

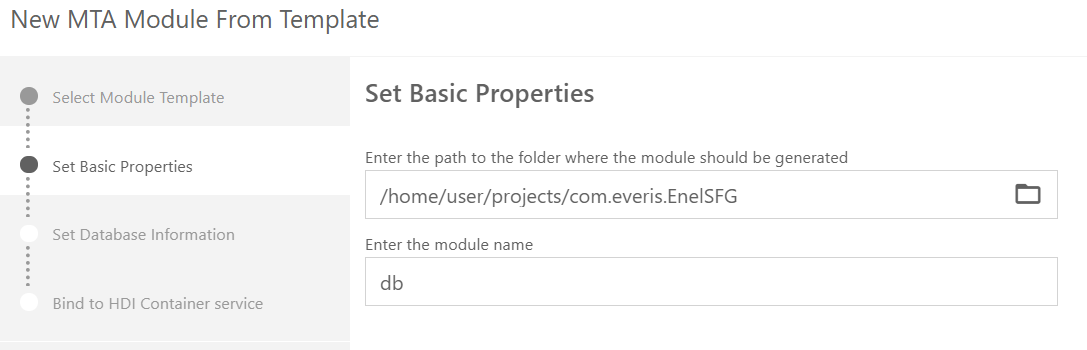


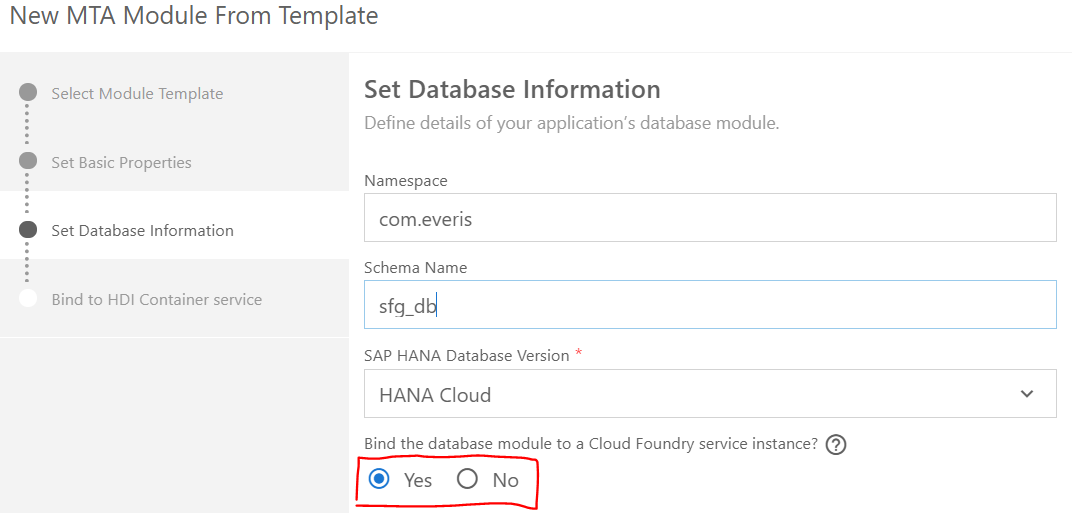
XS-SECURITY.JSON: create file with this properties



1. Create DB

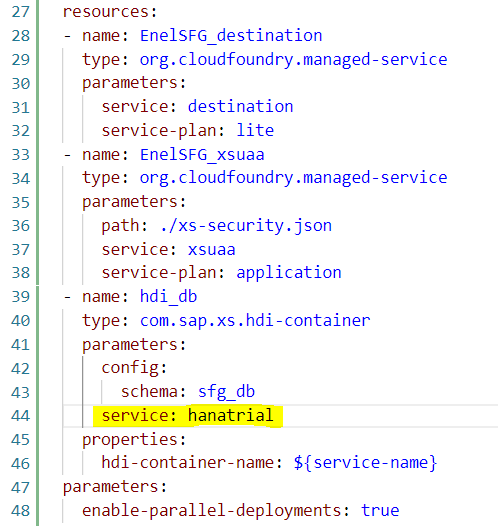
Inside your folder project you can find the mta.yaml file. Left click on this file and choose “create MTA module from template” and then “SAP HANA db module”.





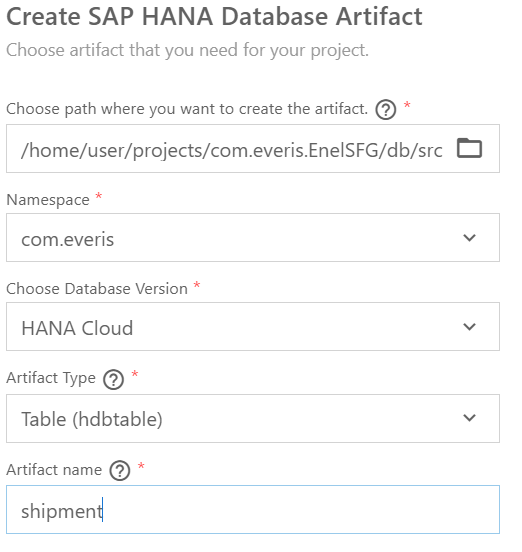
(if you are in trial account select “no”)

If you are in trial account go to MTA.YAML and insert the following row:

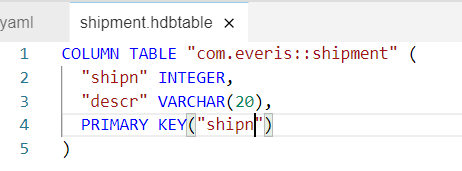


Now you can create all yours table, procedures, ecc…

Click on “SRC” folder inside the “DB” folder. Pres CTRLN+SHIFT+P and write hana --> select “Create sap hana database artifact”.



After this you will find a new file called “shipment.hdtable” and inside this you can define your columns.



1. Create XSJS service

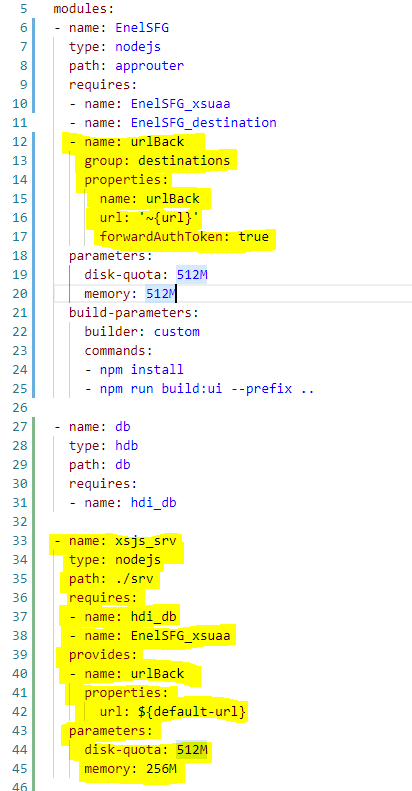
Inside you project create these files and put all your logic:

* srv/lib/srv/filexsjs.xsjs
* srv/lib/srv/filexsjslib.xsjslib
* srv/lib/srv/filexsodata.xsodata
* srv/package.json
* srv/server.js

File examples:



Now you have to define the XSJS service inside the MTA.YAML



1. Call XSJS from UI5 App using authorization

Inside the ui5 app controller you can put the following code:



The “ulrBack” is the “destination” to call XSJS. “urlBack” was defined in MTA.YAML

Now, we need to create a role to access to the XSJS. So, we need to modify the XS-SECURITY.JSON file:



Now in the XS-APP.JSON file inside the approuter we can assign roles:



In this way, with the “$XSAPPNAME.roleApp” role you can access to the UI5 app, and with "$XSAPPNAME.roleXsjs" you can access all the route provides from the “urlBack” destination (XSJS service).

It is very important to define the property “source” of the urlback service in this way. The root “^/urlBack/” define the destination to call from ui5 app. So, for every service is mandatory define the root to call it from ui5 app.

1. Implementing User Information API

Insert “user-api” root destination inside the XS-APP.JSON of approuter



Then, create SERVER.JS file and modify PACKAGE.JSON file inside the approuter folder:



(Change the highlighted row)

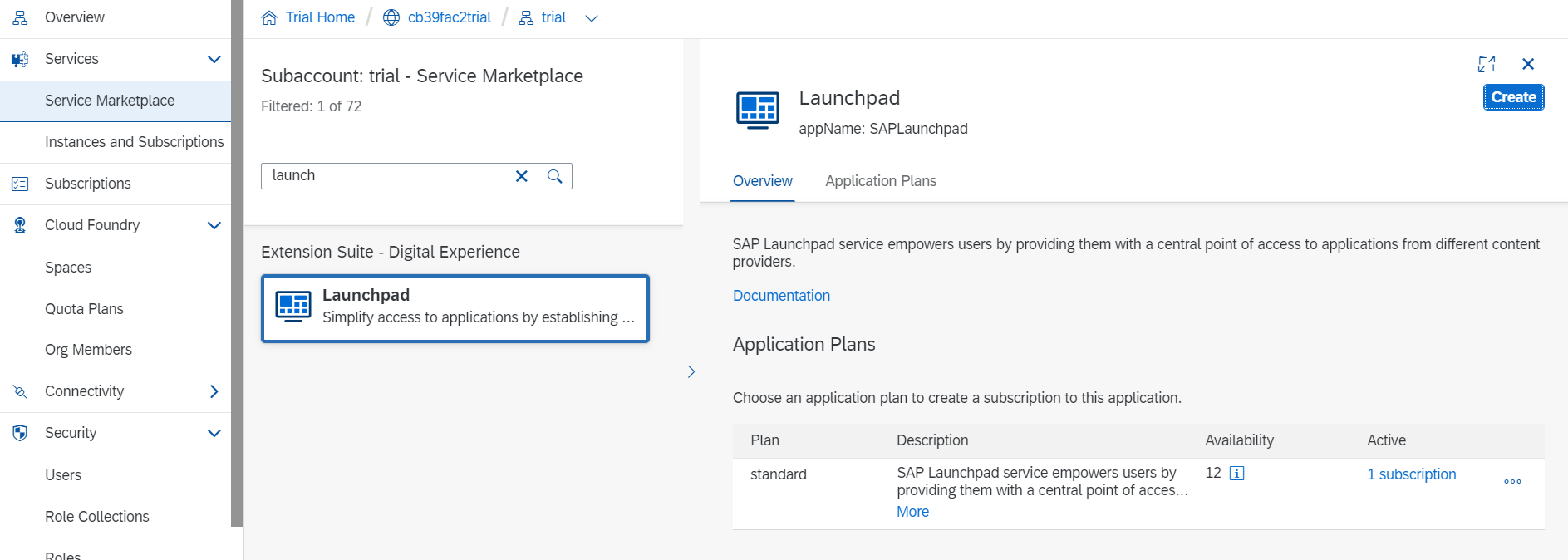


Inside the controller:



1. Implementing theme

Create and instantiate a **Launchpad** subscription and instantiate **UI Theme Design** service



Now you need to assign to yourself the right roles:

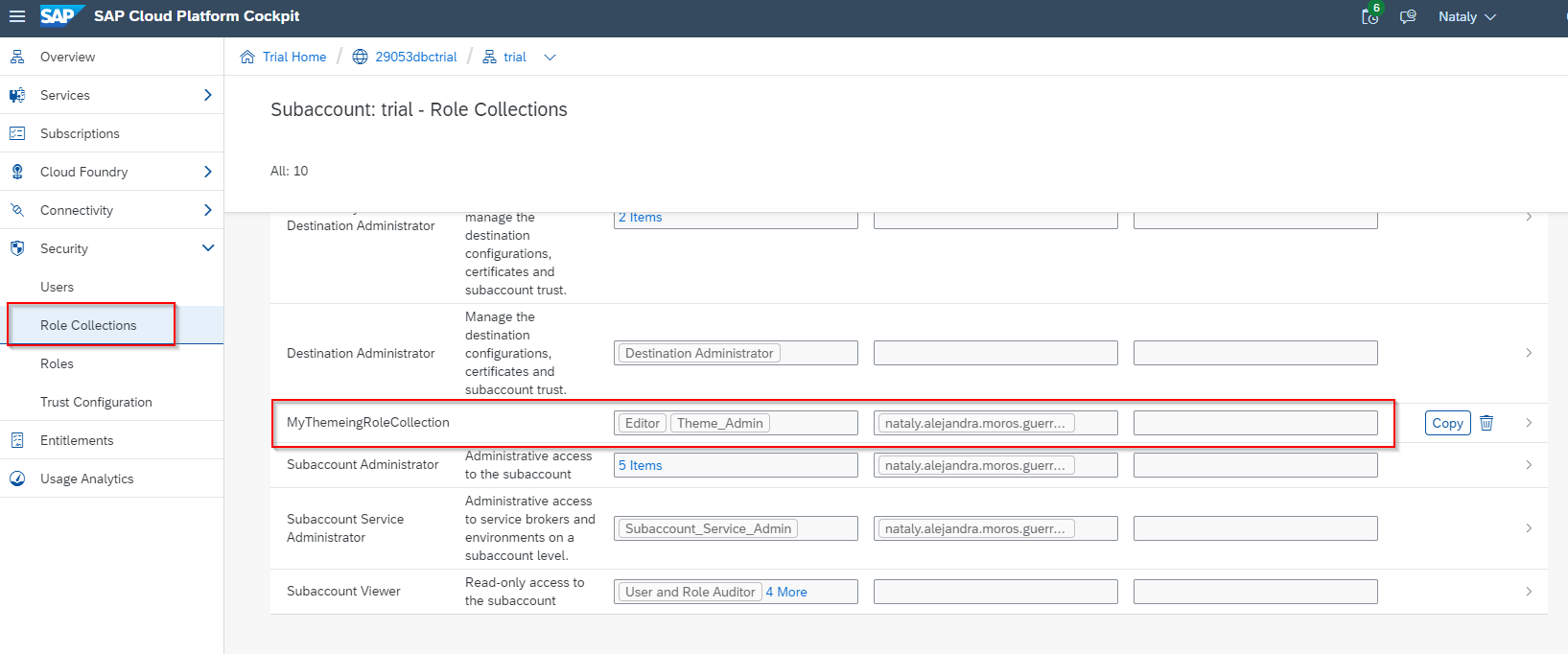
To enable key users to create and manage themes, create a role collection and assign the following roles:

1. In the Cloud Platform cockpit of your subaccount, go to **Security** >> **Role Collections**.
2. Click **New Role Collection**.
3. Specify a name such as **MyThemeingRoleCollection**, and click **Save**.
4. Click the new role collection, and go to the **Roles** screen.
5. Click **Add Role** and add the following roles
   1. For creating roles in the UI Theme Designer, add the following role:

|  |  |  |
| --- | --- | --- |
| **Property** | **Value** | **Description** |
| Application Identifier | sap-theming!<id> | The ID of the application that provides the role template. For example: sap-theming!b2675 |
| Role Template | Editor | Manages the themes of Portal sites. |
| Role | Editor | Manages the themes of Portal sites. |

* 1. For managing themes in the launchpad, add the following role

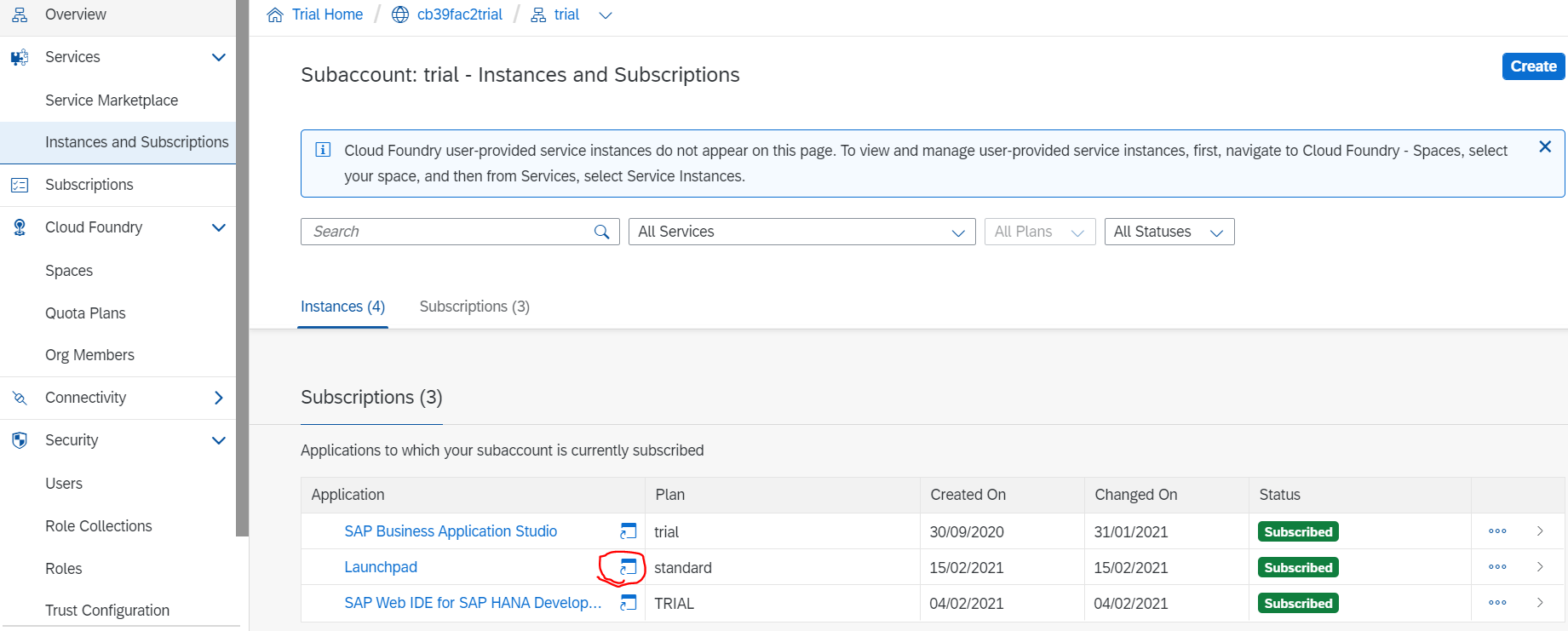
|  |  |  |
| --- | --- | --- |
| **Property** | **Value** | **Description** |
| Application Identifier | portal-cf-service!<id> | The ID of the application that provides the role template. For example: portal-cf-service!1366 |
| Role Template | Theme\_Admin | Manages the themes of Portal sites. |
| Role | Theme\_Admin | Manages the themes of Portal sites |



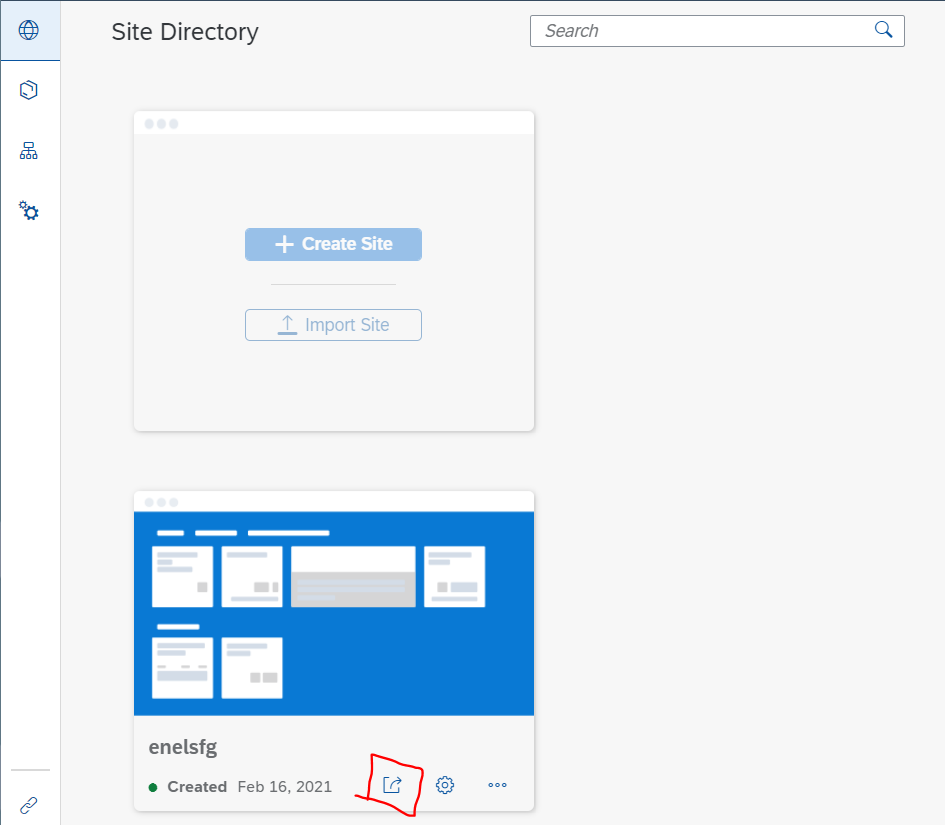
At this point you can assign your custom roles to yourself from **Security Trust Configuration menu.**

You need to assign also **Launchpad\_External\_User and Launchpad\_Admin** roles

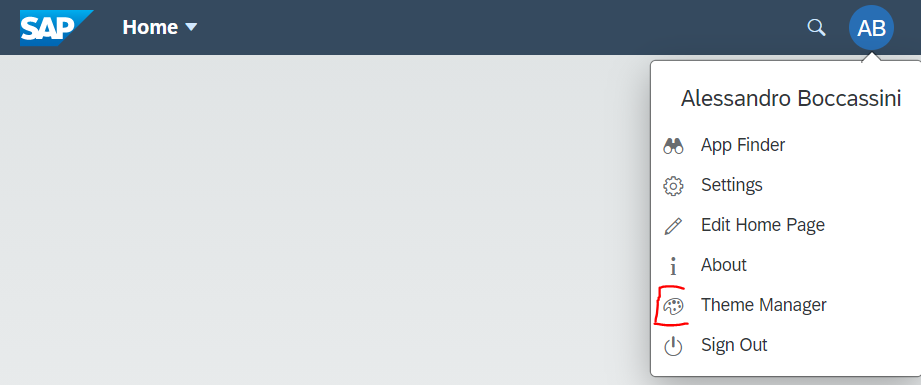
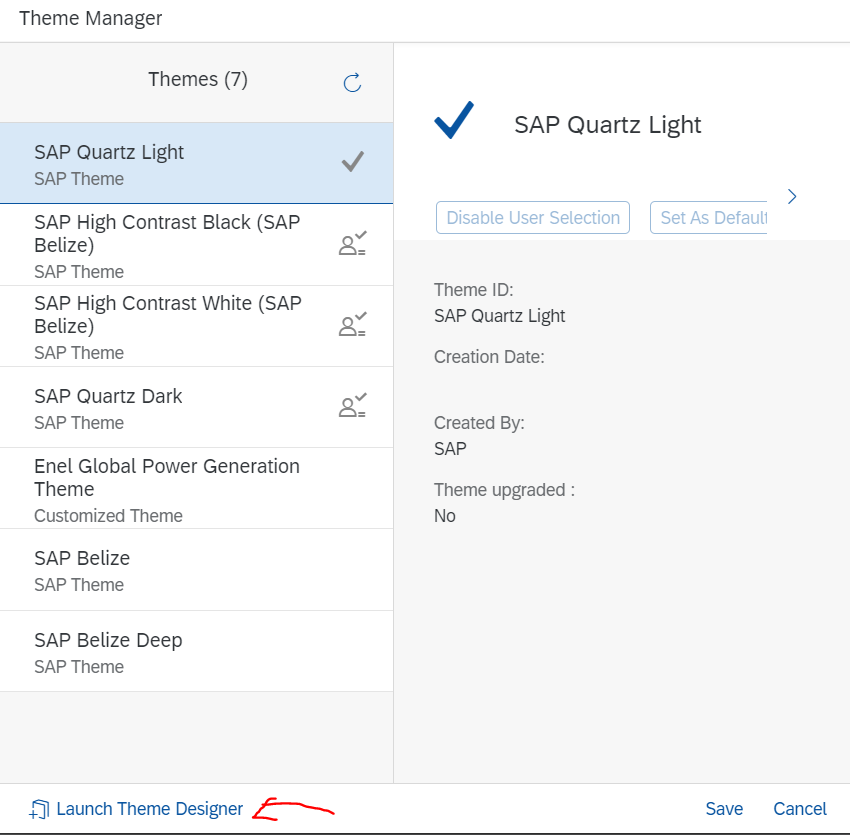
Now you can open the Launchpad:



In the “site directory” menu create a new site.



Enter to this new one and open the theme manager and launch theme designer

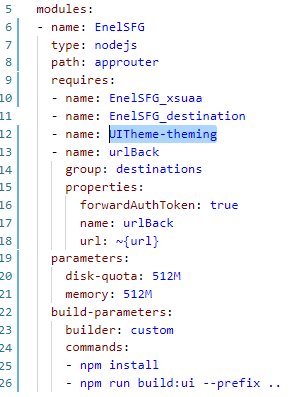
 

At thi point you are able to import a existent theme pack or create new one.

At this point go back to our UI5 app.

In MTA.YAML file you need to assigne the **UI Theme Design** service to the app router:

(use the same used to create the service!)

Now we need to modify the XS-APP.JSON in approuter:



And then set our custom theme in ui5 app (INDEX.HTML in uimodule) (enel\_gpg is the name of created theme pack):



DUPLICATE PROJECT:

Name of duplicated project: com.everis.TestCopy

Search “EnelSFG” and substituite with “TestCopy” in these following path:

* In “uimodule” folder: manifest, index, component, view, controller, ui5.yaml;
* In “approuter” folder: package.json;
* In the main folder of the project: mta.yaml, package.json, xs-security.json (change the xsappname)

Delete folder “DB” and delete definition of db service in MTA.YAML (you don’t need to duplicate db!)

Change the name of the XSUAA service in MTA.YAML. Example: from EnelSFG\_xsuaa to EnelSFG\_xsuaa\_2 and change the assignment.

Change roles in XS\_SECURITY.JSON and XS-APP.JSON in approuter folder

Change the name of the XSJS service in MTA.YAML. Example: from xsjs\_srv to xsjs\_srv2.

At this, point in the main folder of the project, you need to type in terminal “npm install” to create node modules.

Now you can build and deploy!

Local copy of the duplicated project:



Local copy of the original project:

