

CUSTOM BACKOFFICE & LOCALIZATION

Exercise

custom backoffice extensions

- A custom backoffice extension can contribute to the whole and unique Backoffice Webapp with new modules (areas), widgets and / or UI configuration
- Create a new backoffice extension:

- Create the extension with ant

`ant extgen`

- Select the `ybackoffice` template
 - State *name* and *default package* for the new extension, can leave the other values to their defaults (*sample widget* and *style sheet*)
 - Add the new extension to `config/localextensions.xml`
 - Perform complete build of the platform

`ant all`

- Import the extension project into the IDE

custom backoffice extensions

- The following files are relevant within a custom backoffice extension, located under the **resources** folder:
 - [mybackoffice/resources/mybackoffice-backoffice-config.xml](#): Configuration of components defined in the mybackoffice extension. It can also be loaded at runtime by doing a reset in the Application Orchestrator
 - [mybackoffice/resources/mybackoffice-backoffice-widgets.xml](#): Default configuration of components defined in the mybackoffice extension
 - [mybackoffice/resources/mybackoffice-backoffice-spring.xml](#): Contribution to the Widget Application Context. This is an additional Spring context with auto reload possibilities during runtime. It's shared among all the backoffice extensions
- The following files are relevant within a custom backoffice extension, located under the **backoffice** folder:
 - [mybackoffice/backoffice/resources/widgets/mywidget/definition.xml](#): Declaration of a specific widget. The component definition, inputs and outputs, and all meta data related to it.
 - [mybackoffice/backoffice/resources/widgets/mywidget/view.xml](#) (optional): Default view for a widget.
 - [mybackoffice/backoffice/resources/widgets/mywidget/cockpit-config.xml](#) (optional): Configuration of a specific widget. Think of it as the widget's default configuration.
 - [mybackoffice/backoffice/src](#): Source folder for Java classes acting as Controllers for this the widgets of this extension.

Localizing Backoffice Perspective's Name on Hover

- The names of Backoffice cockpits (or perspectives), that are displayed on mouse hover, can be localized. This short tutorial describes how to achieve that.
- You are able to define localization labels for the name of your Backoffice cockpit (perspective).

➤ **PROCEDURE**

- First, define the name of your perspective that you would like to have displayed on mouse hover in the training-backoffice-widgets.xml file, like this:
- trainingbackoffice-backoffice-widgets.xml

Localizing Backoffice Perspective's Name on Hover ...

```
<widgets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="http://www.hybris.com/schema/cockpitng/widgets.xsd">

    <widget-extension widgetId="backofficeMainSlot">

        <widget id="demoextension-perspective"
        widgetDefinitionId="com.hybris.demoextension.backoffice.widgets.demoextensionwidget"
            template="false" slotId="perspectives" title="demoextension.perspective.name">

            </widget>

        </widget-extension>

        <!-- more code -->

    </widgets>
```

Localizing Backoffice Perspective's Name on Hover...

- Now, define the appropriate localization keys for the cockpit name with the `com.hybris.backoffice.perspectiveChooser` prefix. In the `yourextension/resources/yourextension-backoffice-labels/labels_en.properties` directory, add the entry similar to the following:
- `trainingbackoffice/resources/training-backoffice-labels/labels_en.properties`:
- `com.hybris.backoffice.perspectiveChooser.demoextension.perspective.name=The name you want to give to the cockpit`
- Define more labels for other localizations in the same directory: `yourextension/resources/yourextension-backoffice-labels`, for example: `labels_fr.properties`

Localizing Backoffice Perspective's Name on Hover...

- `com.hybris.backoffice.perspectiveChooser.demoextension.perspective.name=Mon nom de cockpit localisée (fr)`
- Redeploy the application by calling `ant clean all` and restart the server by calling `hybrisserver.bat` (Windows) or `./hybrisserver.sh` (Unix).
- Now, when you hover your mouse over the cockpit icon, the name is displayed as per the defined localization keys.

Types Localization in backoffice

➤ Define the node structure & layout of screen in "training-backoffice-config.xml" file. Goto

*\hybris\bin\hybris\bin\custom\trainingbackoffice\resources\training-backoffice-config.xml file and add the xml configuration (define the node and columns) .

Insert the following complete code block in place of the example node structure (remove all the existing code inside the <config> tags)

```
<context component="explorer-tree" merge-by="module">
    <n:explorer-tree xmlns:n="http://www.hybris.com/cockpitng/config/explorertree"> <n:navigation-node id="MyTraining">
        <n:type-node id="Courses" code="Courses" /> </n:navigation-node>
    </n:explorer-tree>
</context>

<context type="Courses" component="listview">
    <list:list-view xmlns:list="http://www.hybris.com/cockpitng/component/listView"> <list:column qualifier="code"/>
    <list:column qualifier="name"/>
    <list:column qualifier="duration"/>
    <list:column qualifier="amount"/> </list:list-view>
</context>
```

➤ Switch to the Application Orchestrator mode by pressing F4 (you'll need to be logged in as an administrator)

➤ Select Reset Everything option under Show cockpit-config.xml

➤ After the page reloads, press again to exit Application Orchestrator mode

➤ Define the table structure in "trainingbackoffice-items.xml" file \hybris\bin\hybris\bin\custom\trainingbackoffice\resources\trainingbackoffice-items.xml file and define the table structure

```
<items xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="items.xsd">

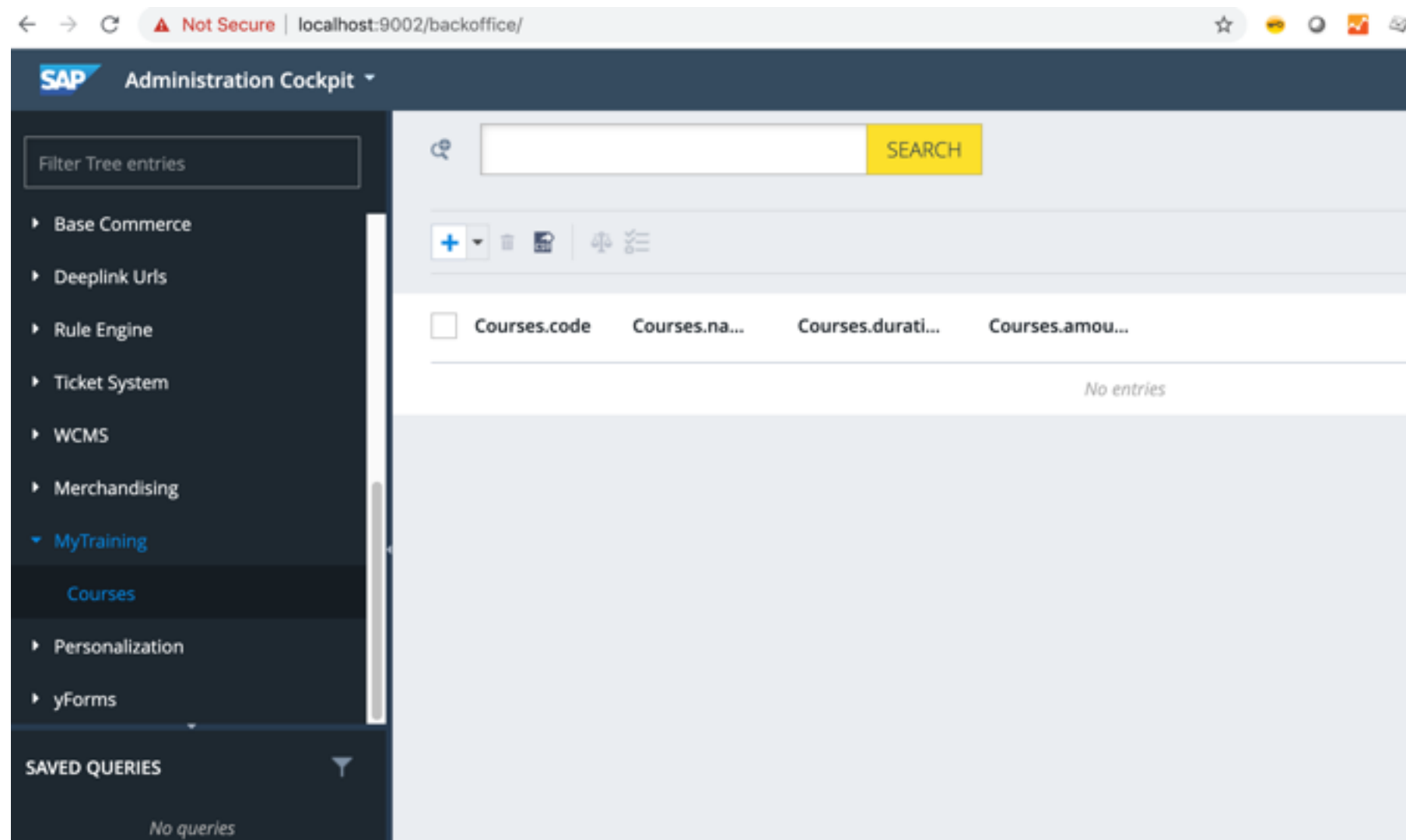
    <itemtypes>
        <itemtype code="Courses" generate="true" autocreate="true">
            <deployment table="TrainingCourses" typecode="10128"/>
            <attributes>
                <attribute qualifier="code" type="java.lang.String">
                    <persistence type="property"/>
                    <modifiers optional="false" unique="true"/>
                </attribute>
                <attribute qualifier="name" type="java.lang.String">
                    <persistence type="property"/>
                    <modifiers optional="false"/>
                </attribute>
                <attribute qualifier="duration" type="java.lang.String">
                    <persistence type="property"/>
                    <modifiers optional="false"/>
                </attribute>
                <attribute qualifier="amount" type="java.lang.Integer">
                    <persistence type="property"/>
                    <modifiers optional="false"/>
                </attribute>
            </attributes>
        </itemtype>
    </itemtypes>
```


Types Localization in backoffice

- Perform the below steps to see the changes
- run the setantenv.bat
- run ant clean all
- Start hybris server hybrisserver.bat
- Open HAC <https://localhost:9002> -> platform->update (by selecting “creating essential data” and Localised type)

Types Localization in backoffice

- Goto backoffice and see the results of the new extension (There should “My Training” node with “Courses” as subnode / element).
- Open Backoffice <https://localhost:9002/backoffice>



Types Localization in backoffice

- Add the new course details by clicking + sign next to the Courses and Save

The screenshot shows the SAP Administration Cockpit interface. On the left is a navigation menu with categories like 'Base Commerce', 'Deeplink Urls', 'Rule Engine', 'Ticket System', 'WCMS', 'Merchandising', 'MyTraining' (expanded), 'Personalization', and 'yForms'. Under 'MyTraining', 'Courses' is selected. Below the menu is a 'SAVED QUERIES' section showing 'No queries'. The main area displays a 'Create New Courses' dialog box. The dialog has a 'MANDATORY VALUES' section with the following fields: 'Time created:' (with a calendar icon), '[duration]:', '[amount]:', '[code]:', and '[name:]'. At the bottom right of the dialog are 'CANCEL' and 'DONE' buttons.

Verify the course details exists in “**TrainingCourses**” table in database or you can get Item Types “Courses ” details by run this flexible search query “select * from {Courses}”

Types Localization in backoffice

- Now stop the server by pressing Ctrl+C
- Add english label in trainingbackoffice-locales_en.properties

```
type.Courses.code.name=Course Code  
type.Courses.name.name=Course Name  
type.Courses.duration.name=Course Duration  
type.Courses.amount.name=Course Amount
```

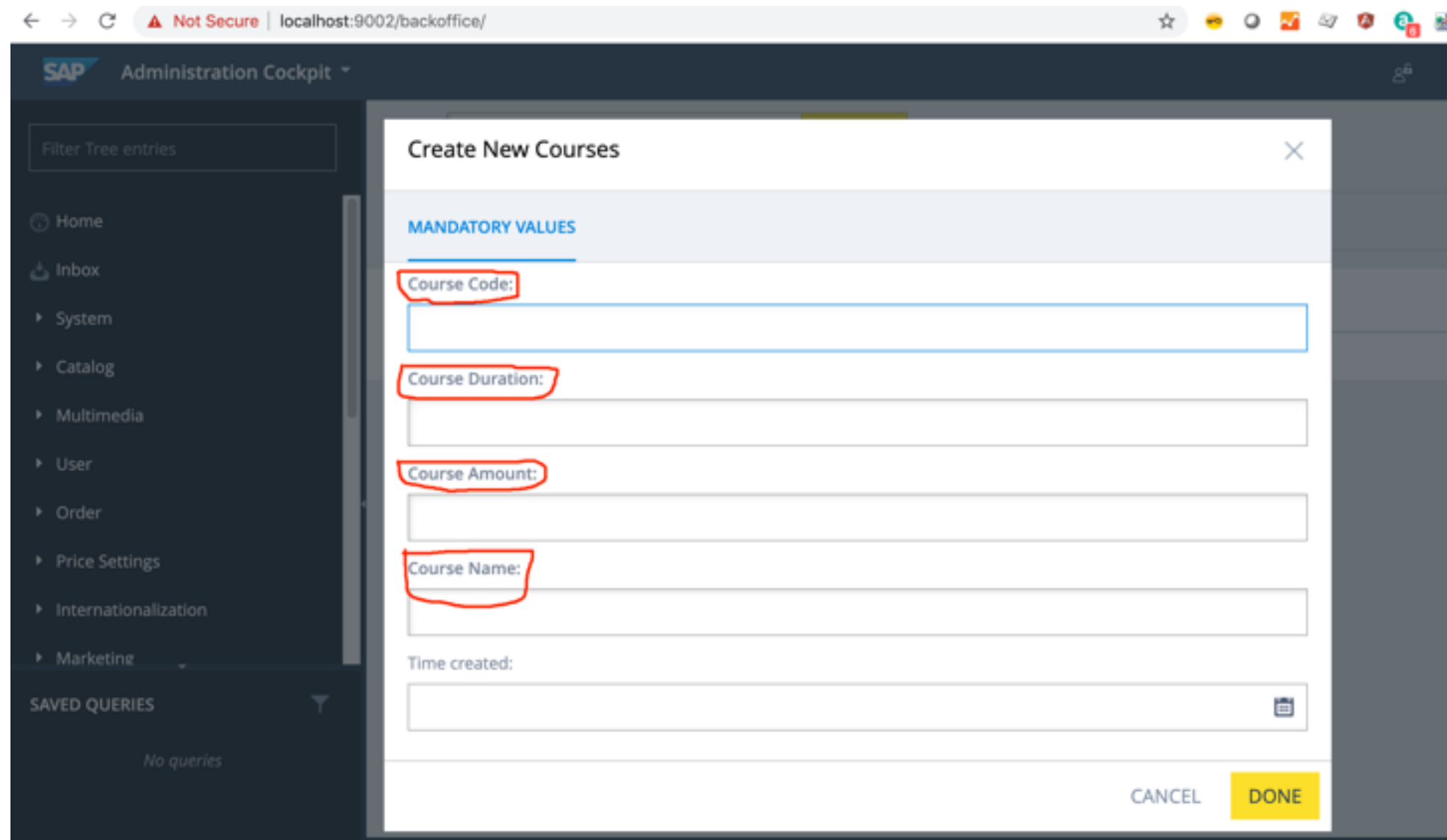
- Add French label in trainingbackoffice-locales_fr.properties

```
type.Courses.code.name=Code de cours  
type.Courses.name.name=Nom du cours  
type.Courses.duration.name=Durée du cours  
type.Courses.amount.name=Montant du cours
```

- Start the server and goto <https://localhost:9002> platform-
>update(select creating essential data and Localised type) -
>update

Types Localization in backoffice

- You can see the Label name



The screenshot shows the SAP Administration Cockpit interface. On the left is a navigation menu with options like Home, Inbox, System, Catalog, Multimedia, User, Order, Price Settings, Internationalization, and Marketing. The main area displays a 'Create New Courses' form. The form has a 'MANDATORY VALUES' section with four input fields: 'Course Code:', 'Course Duration:', 'Course Amount:', and 'Course Name:'. Each of these labels is circled in red. Below these is a 'Time created:' field with a calendar icon. At the bottom right of the form are 'CANCEL' and 'DONE' buttons.

Not Secure | localhost:9002/backoffice/

SAP Administration Cockpit

Filter Tree entries

Home
Inbox
System
Catalog
Multimedia
User
Order
Price Settings
Internationalization
Marketing

SAVED QUERIES
No queries

Create New Courses

MANDATORY VALUES

Course Code:
Course Duration:
Course Amount:
Course Name:
Time created:

CANCEL DONE

“

Thank You

-Akshaya