Exercise 2

Create a new cockpit

INTRODUCTION

This document will guide you through creating a custom Backoffice extension which will contain a new (very simple) Backoffice cockpit. It's going to be the layout configuration that defines this new cockpit. In subsequent chapters, you will develop this cockpit with incrementally more complexity throughout this training.

What you will learn

You will learn how to create a custom Backoffice extension and a layout for a new cockpit inside the Backoffice application.

What's particularly important about this exercise is that you'll also learn how to do some of the basic Backoffice configuration that you'll be doing over and over again in the rest of the training.

What you will need before starting

- Bookstore is installed and initialized (complete exercise 1).

INSTRUCTIONS

1. Stop the SAP Commerce server if it's already running.
2. If you haven't already done these two things in your terminal/console:
   - Navigate to `YOURPATH/workspace/hybris/bin/platform`
     From now on, we will refer to this as your ***platform directory***.
   - Invoke:

     (on Windows) `setantenv.bat`

     (on OSX or Linux) `. ./setantenv.sh`

     This will set your console's environment variables so that the Ant tool being used is the one that is bundled within each installation of the commerce suite.
3. Now it's time to create a brand-new backoffice extension. Execute the following command in your terminal/console from your ***platform directory***:

   `ant extgen`

   You will be prompted with a few questions. The following table contains the answers to use for each question. You are not required to use these values, but for the sake of solution verification and consistency through this training, please use the following values:

Property	*Value*
Template	ybackoffice
Extension's Name	bookstorecustombackoffice
Package	my.bookstore.bookstorecustombackoffice
Register as a SASS extension?	true
Create Sample Widget?	true
Create sample style sheets?	False

4. The new extension will be created under `…/hybris/bin/custom`. Using a stand-alone text editor (e.g., Notepad, vi, ed, emacs, TextPad, etc.), add this extension to `…/hybris/config/localextensions.xml`, then rerun `ant all` from the `…/hybris/bin/platform` directory to incorporate it into the build.
5. Start your server by running `hybrisserver.bat` (on Windows) or `./hybrisserver.sh` (on Mac/Linux).
6. Go to the Backoffice application here: https://bookstore:9002/backoffice/

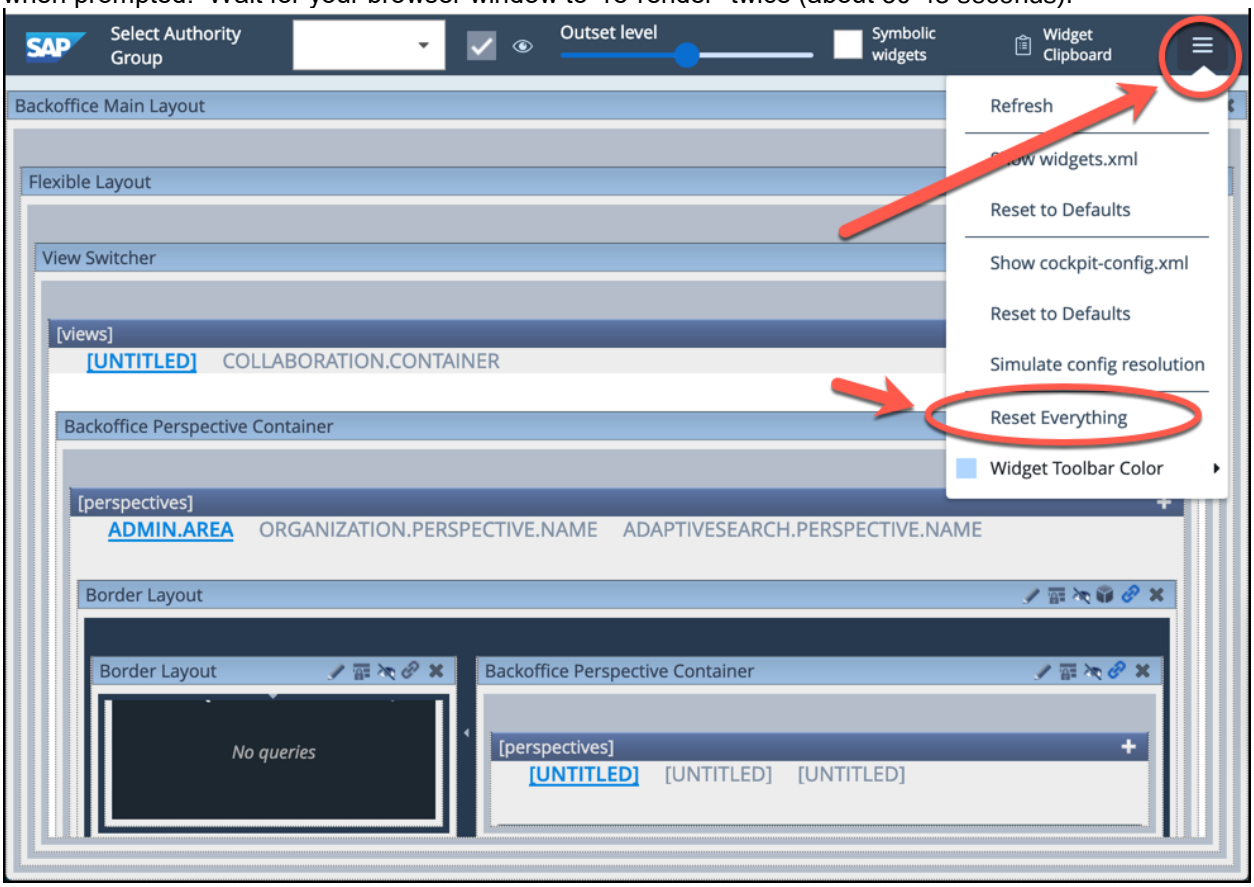
   (If you weren't able to change the content of the `hosts` file in the previous exercise, try this link: https://localhost:9002/backoffice/)
7. Log on to Backoffice using the admin credentials:

User: `admin`
Password: `nimda`

8. Press `F4` to open Application Orchestrator.

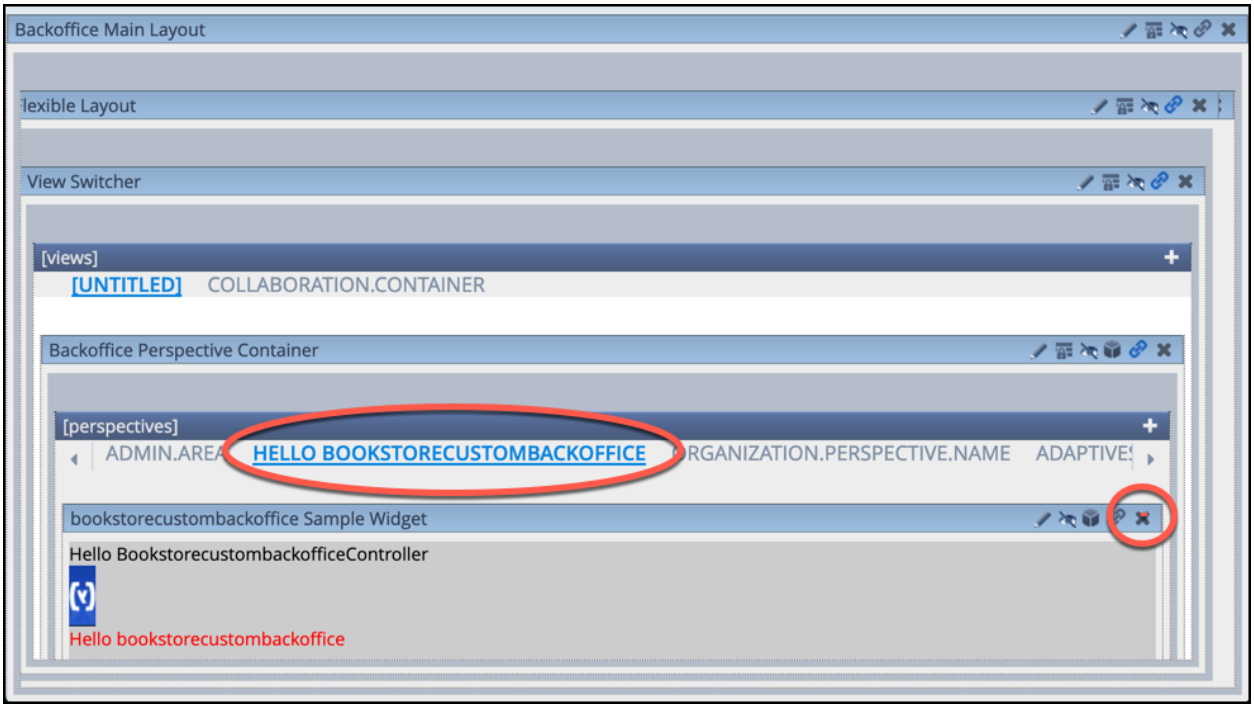
> If you are using a Mac machine, and pressing `F4` is not working for you, you should press `fn+F4`.

9. Because we have just altered Backoffice's widget-layout via the auto-generated contributions in our new `bookstorecustombackoffice` extension that is now part of the build, we need to prompt Backoffice to reset its view by re-reading ALL Backoffice configuration metadata. Do this by clicking on the "stack menu" icon and selecting "Reset Everything", and answering, "Yes" when prompted. Wait for your browser window to "re-render" twice (about 30-45 seconds).

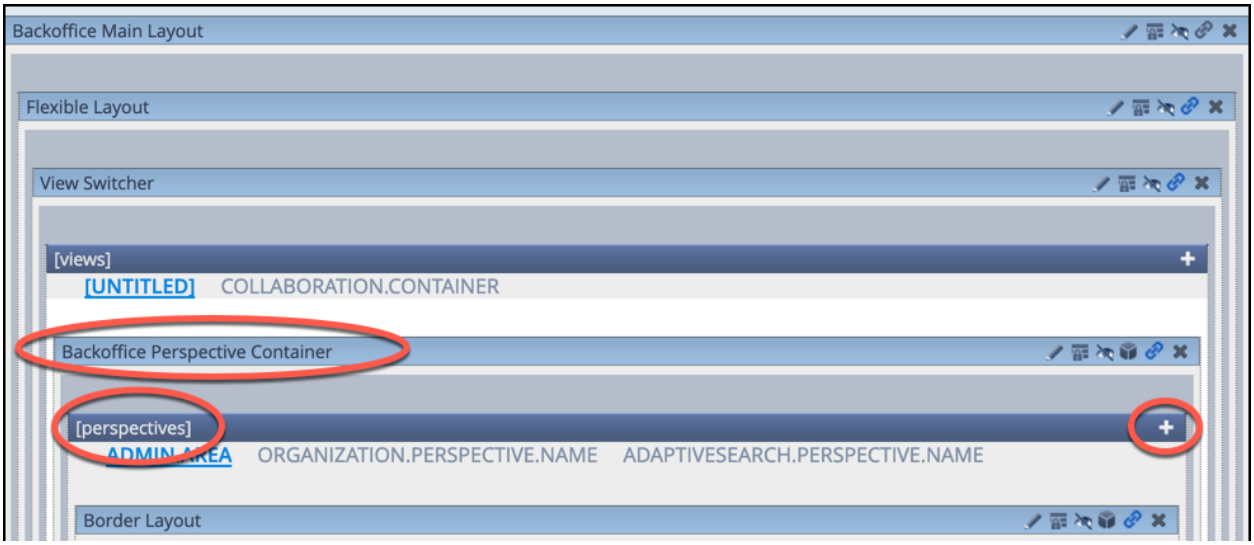


10. In *Backoffice Framework* terminology, *cockpits* were once called *perspectives*. Click on the *cockpit* tab labeled *HELLO BOOKSTORECUSTOMBACKOFFICE* within the *[perspectives]* container. Notice that this cockpit (tab view) contains a widget instance called *bookstorecustombackoffice Sample Widget*. This widget instance was created during our recent

ant extgen operation because we answered `true` to the question, “Create Sample Widget?”. Remove this widget by clicking on the **X** indicated below. This will remove the *HELLO BOOKSTORECUSTOMBACKOFFICE* perspective (tab) as well.



11. Now let's choose the layout for our new cockpit (creating the cockpit at the same time). Find the widget directly inside the outermost *Backoffice Perspective Container* called *[perspectives]* and then press the **"+"** button to open the ***Choose widget*** wizard.



Be careful to perform this step on the outermost *[perspectives]* widget – not some other nested *[perspectives]* widget.

12. Search for "Border Layout" (it's under the Layout category) and continue through the wizard until get to the choice of buttons: "Add & close" or "Add & connect". You should select "**Add & close**", since we don't want to establish any communication between this new widget and other widgets at this point.

    The new layout will be added as another tab next to the already existing cockpits and will be labeled as *[UNTITLED]*.

13. Click on this new *[UNTITLED]* tab (which will be the container for our new cockpit) and then click the pencil icon  (of the *Border Layout* contained immediately within) to open its settings. Within these settings, give the cockpit a name and an intuitive, unique ID, and while in the settings, let's also remove the right-most slot in the layout…

    Use the values in the table below for the new settings, leaving unchanged any properties not mentioned in this table.

Property	*Value*
Title	Book Management
Widget ID	bookstorecustombackoffice-cockpit
eastClosed	true

You may ask yourself how one can find out what these settings are. Remember the link to the list of available widgets on one of the slides? There, you can find links to each widget's complete description.

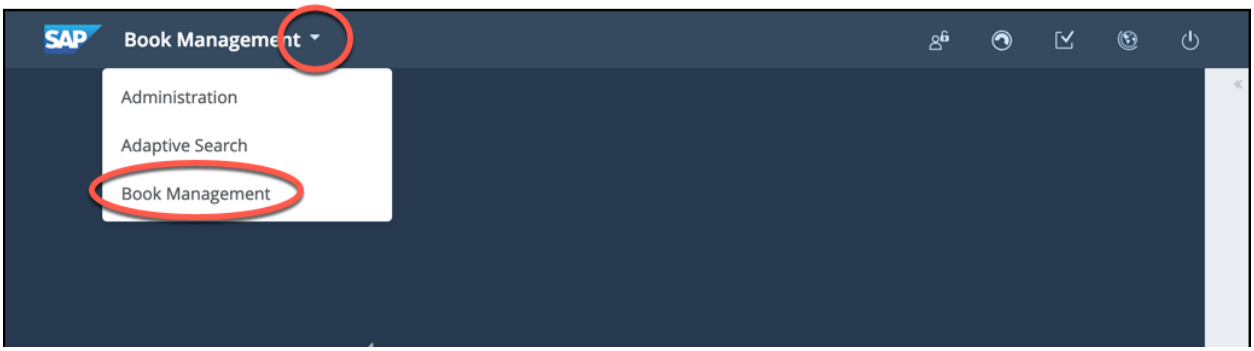
In case of the Border Layout widget, you can find its description here.

For every widget instance (a cockpit is a widget), a unique *widget ID* is automatically generated. You should always provide a more intuitive *widget ID*, though. This will allow you to differentiate your widget instance from any other widget instance of the same type.

The value of the *Title* setting is what gets shown as the name on the tab instead of the automatically generated name, *[UNTITLED]*.

eastClosed is a Boolean property that tells the widget whether you want to keep the slot on the right enabled (opened) or disabled (closed). Closing a slot will leave more space for the remaining open slots.

14. Press `F4` to exit Application Orchestrator and see the effect of your changes. You should now be able to see a new cockpit called, “Book Management”.



So far, we have created a Backoffice extension, *bookstorecustombackoffice*, and a layout for the new cockpit, *Book Management*.

Recall how the *application configuration* holds information about all of the widget instances in Backoffice—coming from different extensions—merged into one file. The layout that you created using Application Orchestrator is preserved in that file, but only temporarily.

This configuration will be lost as soon as you reset the *application configuration* by resetting `widgets.xml` in Application Orchestrator.

15. Press `F4` to enter Application Orchestrator again. Click on the “stack menu” in the upper right-hand corner and select *Reset to Defaults* directly under *Show widgets.xml*. This will reset the widget layout configuration for all of Backoffice to the initial configuration specified via all of your extensions’
*`-backoffice-widgets.xml` files.

Exit Application Orchestrator and confirm that the *Book Management* cockpit has disappeared and that the *HELLO BOOKSTORECUSTOMBACKOFFICE* cockpit has reappeared.

To make this new cockpit a permanent part of Backoffice, we can add its configuration into the *application configuration* of the *bookstorecustombackoffice* extension. Remember to also add this extension to your `localextensions.xml` so that it gets included in the build and loaded to the SAP Commerce server.

16. Using a stand-alone text editor, open the new extension's *widget layout configuration* file located here:

    `${HYBRIS_BIN_DIR}/custom/bookstorecustombackoffice/resources/bookstorecustombackoffice-backoffice-widgets.xml`

    In this file, as you can see below, there is already a widget called "bookstorecustombackoffice-perspective", which is the sample widget that we chose to let `ant extgen` generate during creation of your new Backoffice extension. (Remember, Backoffice *cockpits* used to be called *perspectives*.)

    bookstorecustombackoffice-backoffice-widgets.xml

```
<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="bookstorecustombackoffice-cockpit"
                        widgetDefinitionId="my.backoffice.bookstorecustombackoffice.widgets.bookstorecustombackofficewidget"
                        template="false" slotId="perspectives" title="Hello bookstorecustombackoffice">
                        <setting key="perspectiveImageUrl" type="String" value="/cng/images/book.png" />
                </widget>
        </widget-extension>
        ...
</widgets>
```

17. With minimal effort, we can modify this widget instance's configuration to recreate the same cockpit layout created earlier, except directly through XML this time.

    The changes you should make are:

    - Replace the current ***widget id*** with **"bookstorecustombackoffice-cockpit"**
    - Replace the current ***widgetDefinitonId*** with **"com.hybris.cockpitng.borderlayout"** to tell the system what type (definition) of widget instance we want to create and place here. Recall where to find the descriptions of the OOTB widget definitions via SAP Help.
    - Replace the widget's ***title***, "Hello bookstorecustombackoffice", with **"Book Management"**.
    - Use the *setting* tag to add the ***eastClosed*** setting (with *type*=***Boolean*** and *value*= **true**) that you previously set in Application Orchestrator in step 9. This widget instance already has one setting you can use as an example of a widget setting's XML structure.

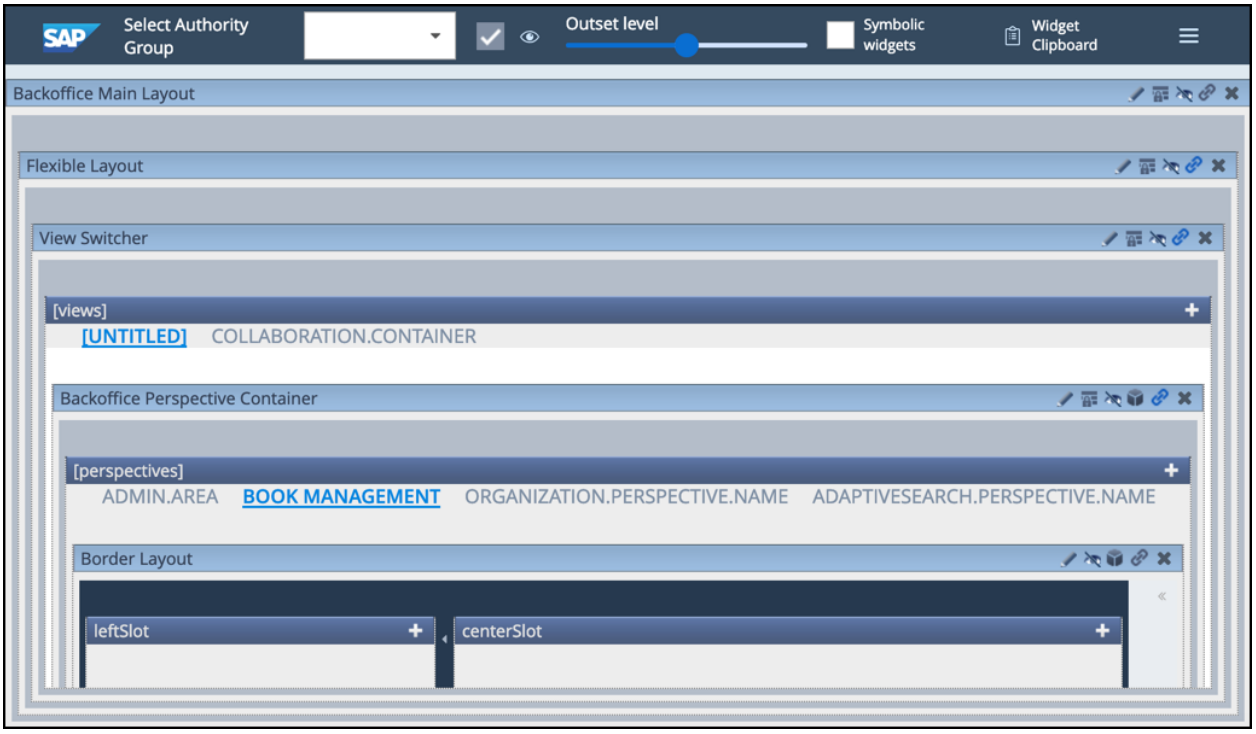
There is another, more commonly used trick to add our new cockpit layout permanently to Backoffice. If we hadn't reset our widget layout (as we did earlier in the exercise), you could go to Application Orchestrator's stack menu, *Show widgets.xml*, and find the XML entry for the border layout widget we created/added via the Application Orchestrator. You would then copy this definition entry from there and paste it into `bookstorecustombackoffice-backoffice-widgets.xml`. Thus, we can use Application Orchestrator as our application configuration generator, and then copy the generated XML elements into your extension's permanent configuration file.

But, for simplicity, we chose to provide you with the data to put into this file.

You *don't* need to restart the server or rebuild when editing an application configuration XML file. Just go into Application Orchestrator and, from the *stack menu* in the upper right-hand corner, select either *Reset to Default*s directly below *Show widgets.xml*, or select *Reset Everything*.

You *DO* need to rebuild and restart the server after adding a new extension to `localextensions.xml` or after creating/modifying Java code.

18. Log into Backoffice using the admin credentials. Go to the Application Orchestrator and *Reset to Defaults* (right under Show widgets.xml). If you can again see the Book Management cockpit as you did in step 14, you've completed this exercise successfully.



Congratulations, you are now ready to work on the rest of the exercises!