Develop SAP Business One extensions for the SAP Business One Web Client

PARTNER

Run Simple

TABLE OF CONTENTS

The objective of this hands on is to put in practice how to develop SAP Business One extensions on SAP Business One Web Client.

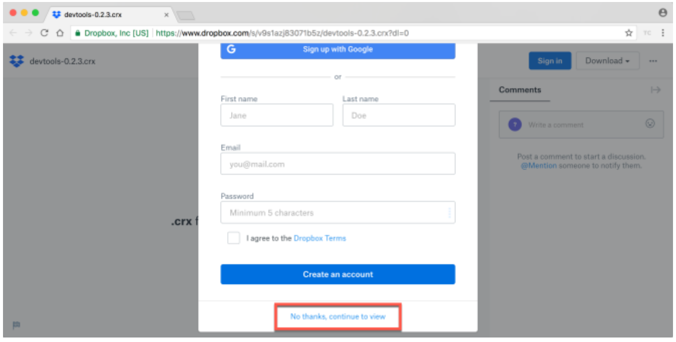
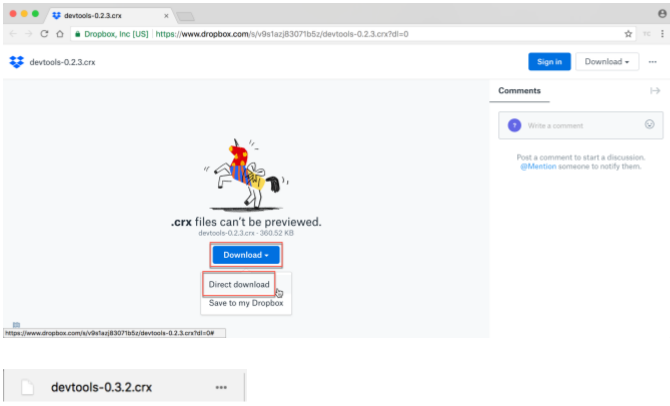
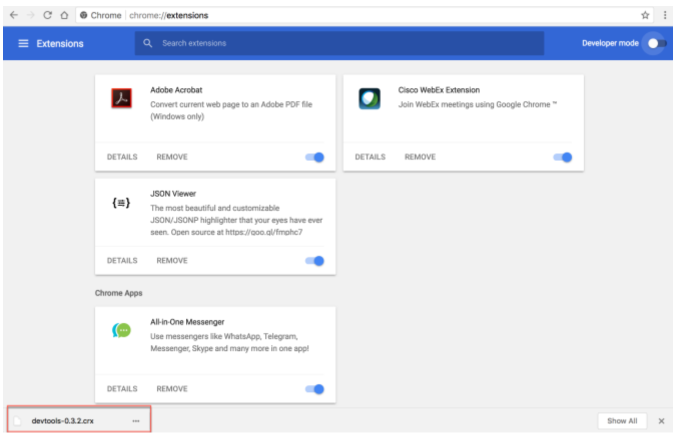
The exercise will be comprised by:

- Step 1: Automatically fill the *“delivery date”* before saving the Sales Order
- Step 2: Make UI field *“Customer Reference Number”* mandatory
- Step 3: Add a button to the Sales Orders UI and handle the button click

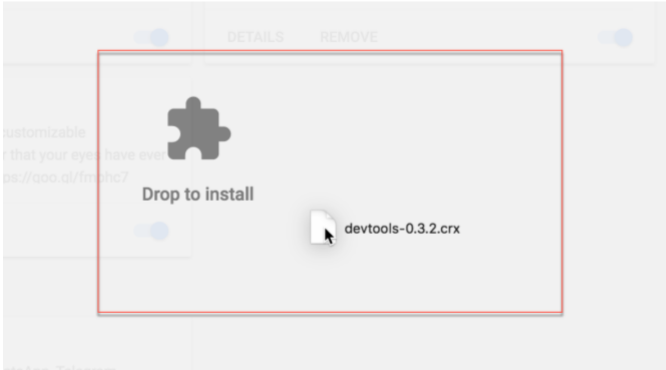
PREREQUISITES

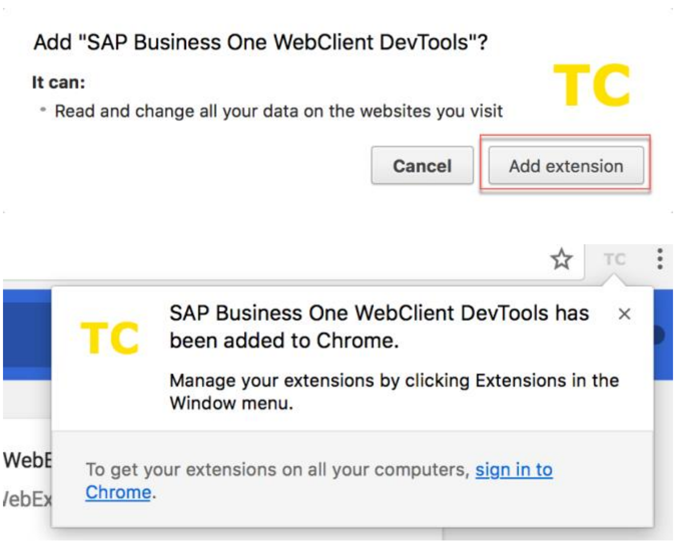
i. SAP Business One Web Client DevTools

This is the SAP Business One Web Client DevTools, the Chrome Extension which enables you to get detailed technical data of the SAP Business One Web Client UI components.

Explanation	Screenshot
Open Google Chrome	
Download and save the latest version of the Chrome extension here: https://goo.gl/qTX1ge	 
In your chrome, open URL "chrome://extensions/".	

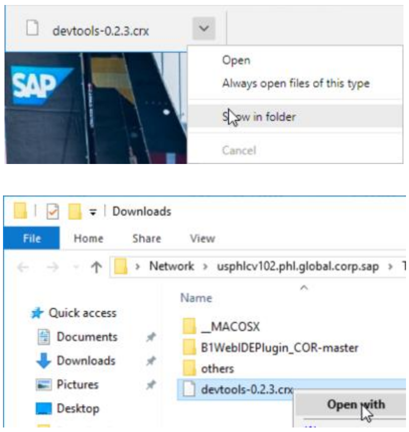
<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td></td><td></td></tr>
<tr><td>Drag the downloaded extension file (*.crx) and drop it into the opened tab.

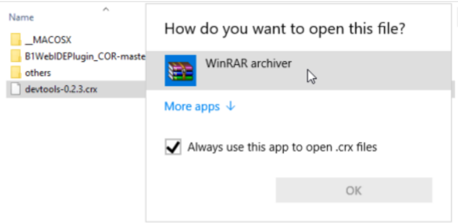
It will automatically install into your chrome extension.</td><td>
</td></tr>
<tr><td>Click “Add extension”

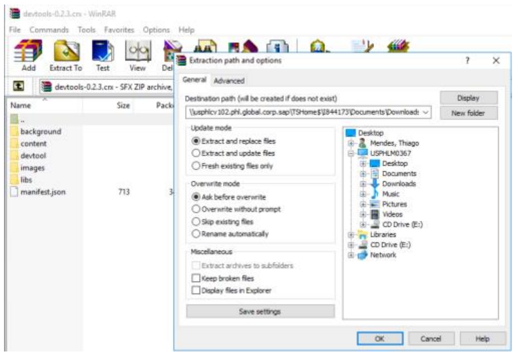
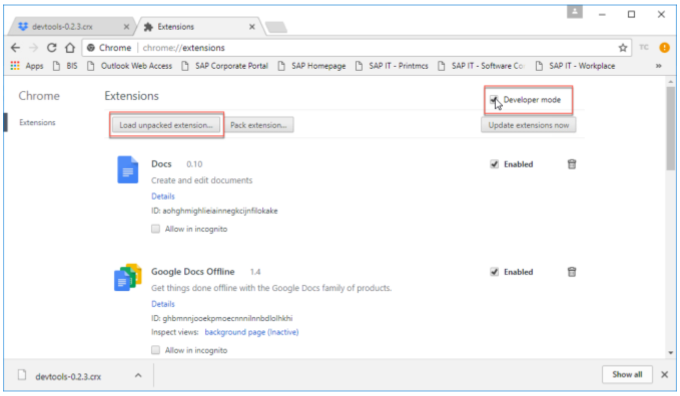
Close the message that will pop-up</td><td>
</td></tr>
<tr><td>Follow the next steps only if you are using windows-based computer.

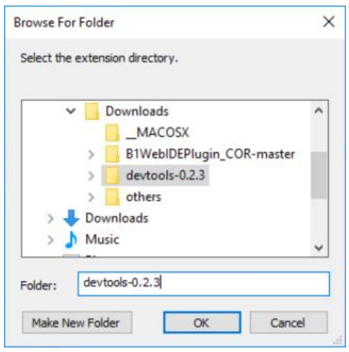
For security reasons, in chrome latest version, the extension will be disabled by default. Here are the steps to enable it:

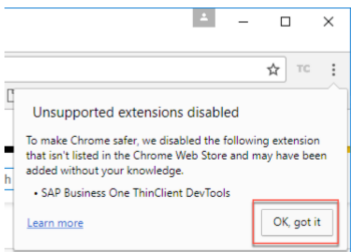
Extract the file you have downloaded (devtools-0.3.2.crx) to a folder.

NOTE: the *.crx file is just a renamed ZIP file, so any archive/compression solution should be able to extract it.</td><td>
</td></tr>
</table>

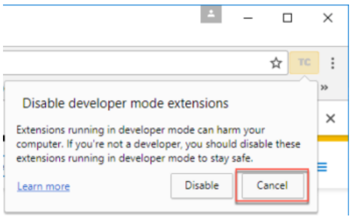
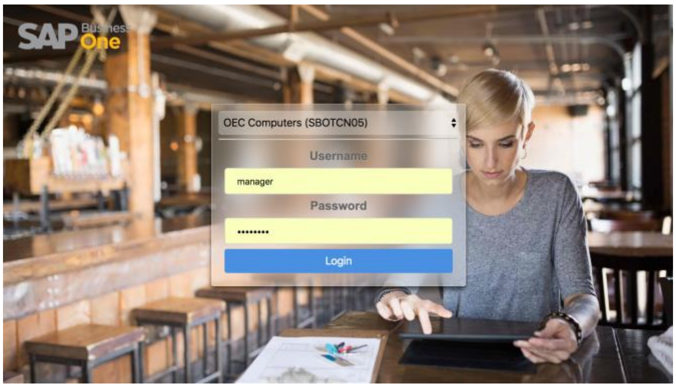
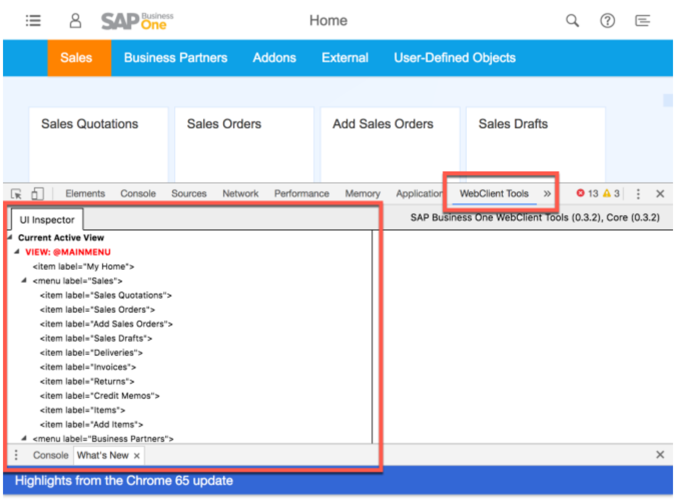
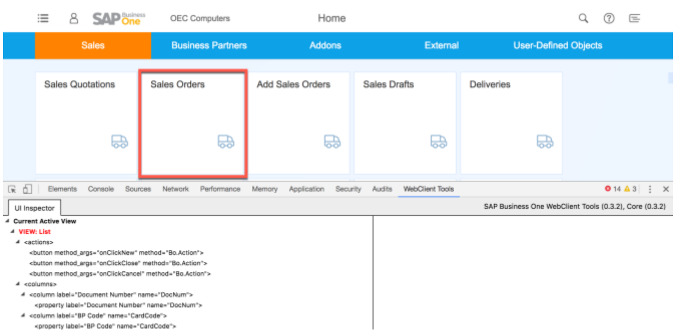
<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td></td><td>


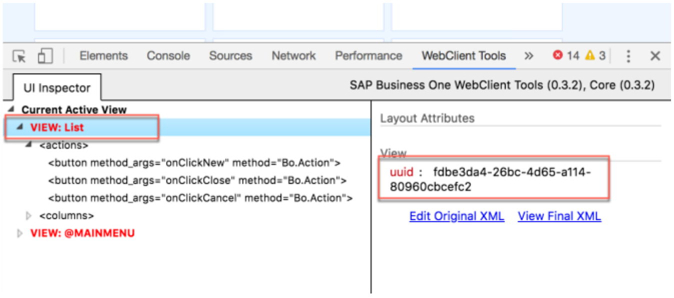

</td></tr>
<tr><td>Open the extension management in Google Chrome
a) enable 'developer mode'
b) click 'Load unpacked extension…'
c) fill the path you extracted in step 1.
d) restart Google Chrome</td><td>




</td></tr>
<tr><td>When the chrome restart, you need to confirm not to disable the DevTools extension.</td><td>


</td></tr>
</table>

Explanation	Screenshot
	
Let's check that the DevTools extension is working properly. Open new window on Google Chrome and browse to the SAP Business One Web Client page: https://goo.gl/GfWDrD Login using below credentials: Company: **Check with your instructor** Username: **manager** Password: **Initial0** Click Login	
Open the Developer Tools: Microsoft Windows: (Ctrl + Shift + I) macOS: (Cmd + Opt + I) Click the "WebClient" Tools" tab. Click the "UI Inspector" tab.	
Click the Sales Orders tile	

Explanation	Screenshot
In the “UI Inspector” tab, select the “VIEW: LIST”. Check the right-frame, the attributes of the selected component are shown there. In this case, the ***Universally Unique Identifier*** for the Sales Order List View is shown here. Select other components on the “UI Inspector” tab and see how they get highlighted in the browser.	

Now follow the steps below.

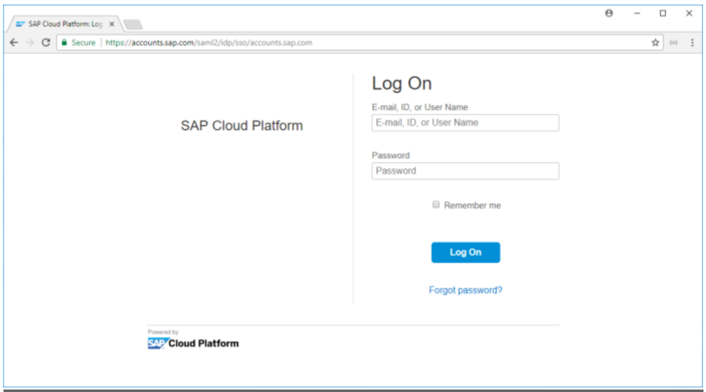
EXERCISE 1: AUTOMATICALLY FILL THE DELIVERY DATE

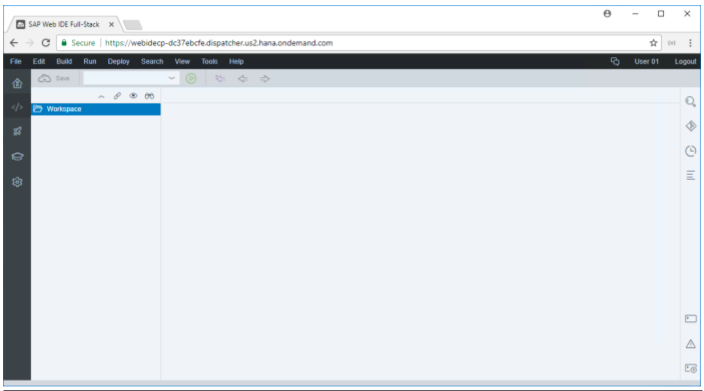
Let's get started, coding your first loosely-coupled application on top of SAP Business One Web Client.

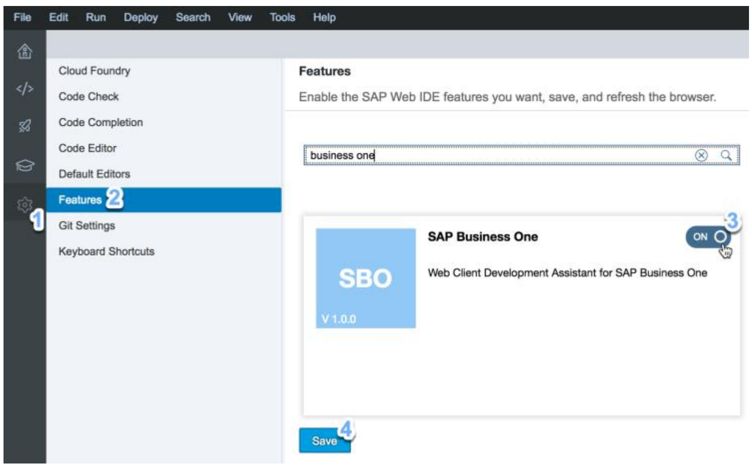
For this first exercise, we want to do something very simple: during the creation of a new Sales Order, automatically fill in the delivery date (in the case it's not yet filled) before saving it.

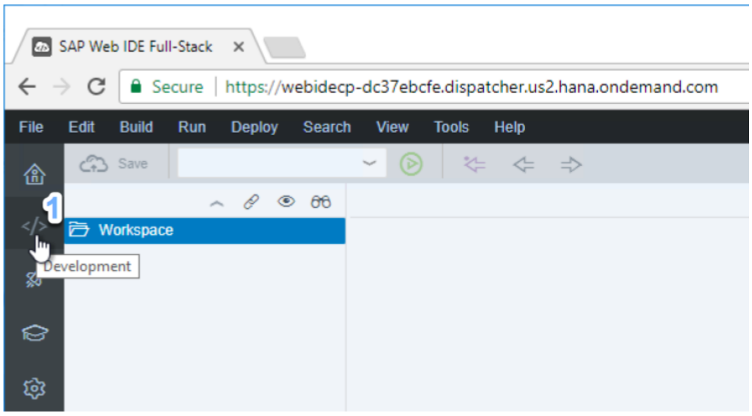
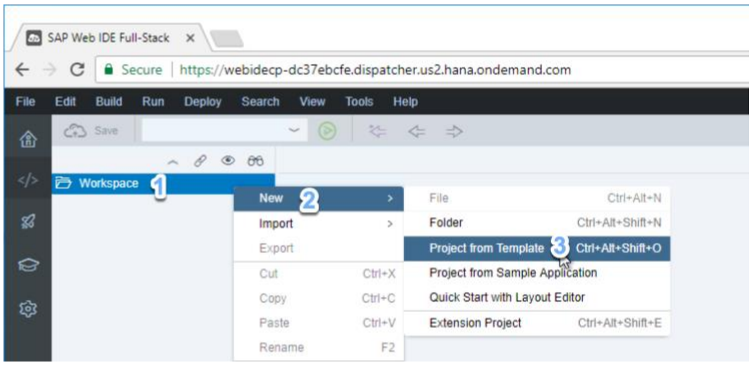
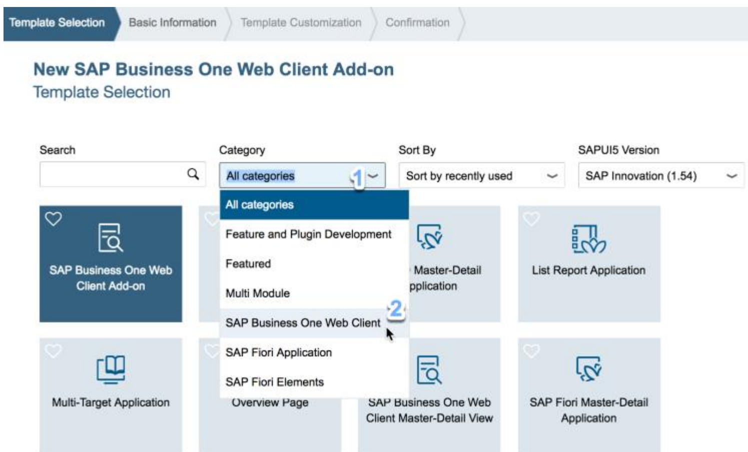
This will cover how to handle an UI event (before button click) of a given view (Sales Order detail) and how to manipulate UI values (current date + 5) of an existing UI component (delivery date field).

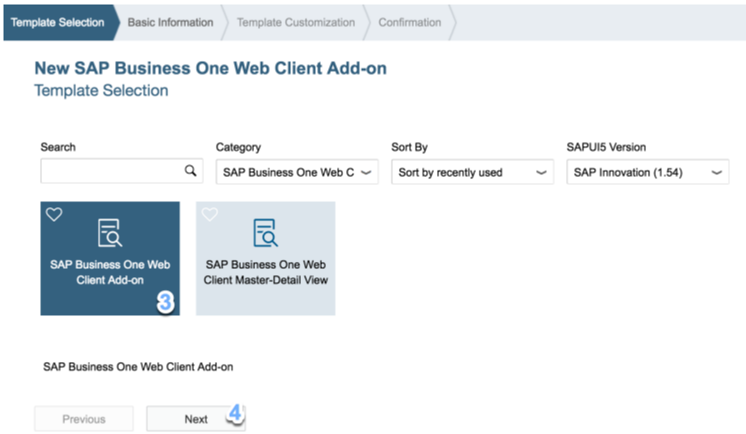
<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Open Chrome and access SCP Web IDE using your S-number credentials.

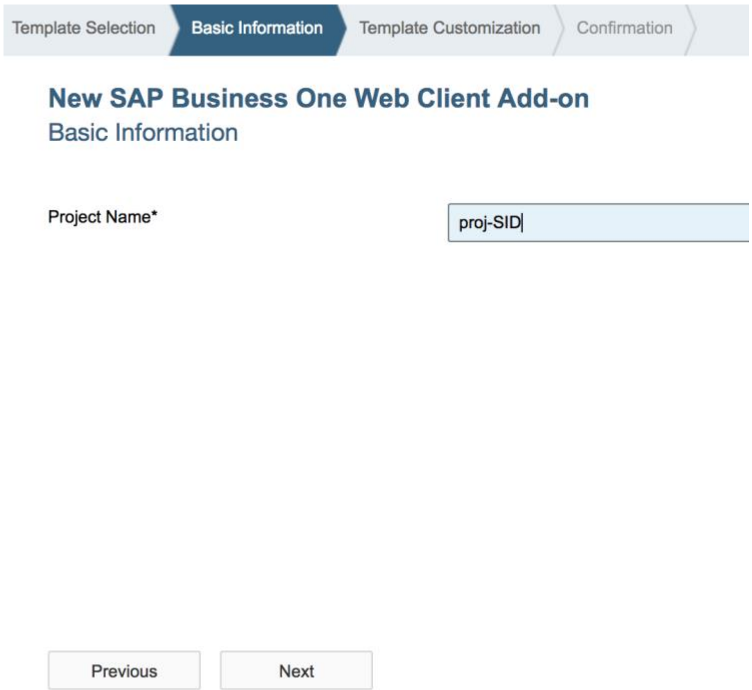
https://goo.gl/PZqa1Y</td><td>

</td></tr>
<tr><td>Select the Preferences perspective;
Click Features;
Search for "business one";
Enable the SAP Business One feature.

DO NOT FORGET TO SAVE AND REFRESH THE PAGE!</td><td>
</td></tr>
</table>

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Select the Development perspective</td><td>
</td></tr>
<tr><td>Right click the Workspace folder, select New > Project from Template</td><td>
</td></tr>
<tr><td>Open the Category dropdown field > select SAP Business One Web Client</td><td>
</td></tr>
</table>

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Select the SAP Business One Web Client Add-on and hit “Next”</td><td>
</td></tr>
<tr><td>Fill in the Basic Information as bellow:

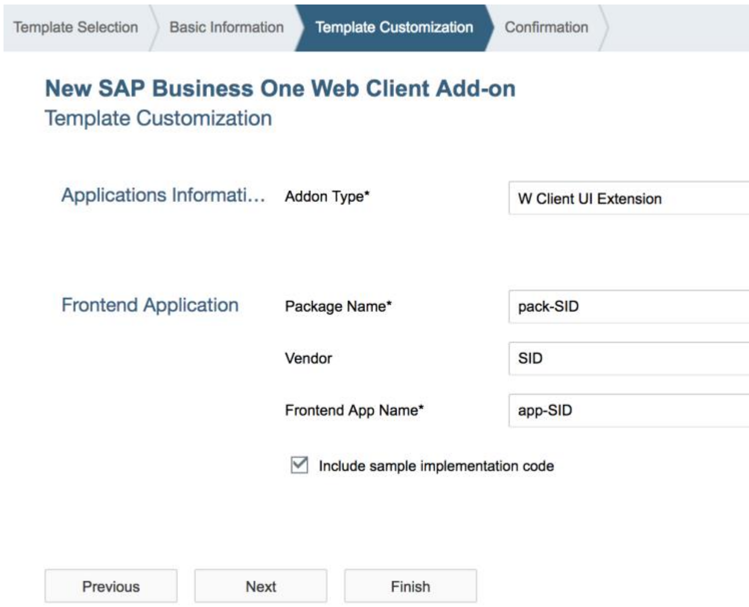
Project Name: proj-<s-number>
Click “Next”</td><td>
</td></tr>
</table>

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>
Fill in the Template Customization as bellow:

Add-on Type: W Client UI Extension
Package Name: pack-<snumber>
Vendor: <snumber>
Frontend App Name:
app-<snumber>

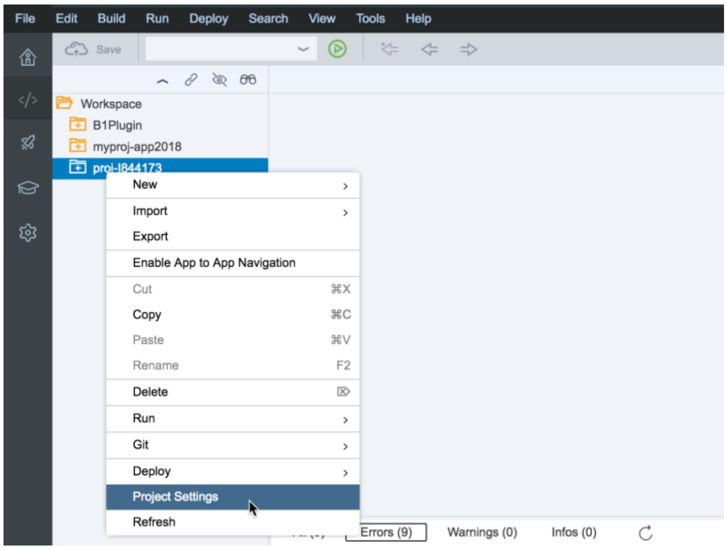
(Note: DO NOT FORGET THE DASH for the Frontend App Name)

Check the “Include sample implementation code” option. (Refer to the appendix section to get further information on the sample code)

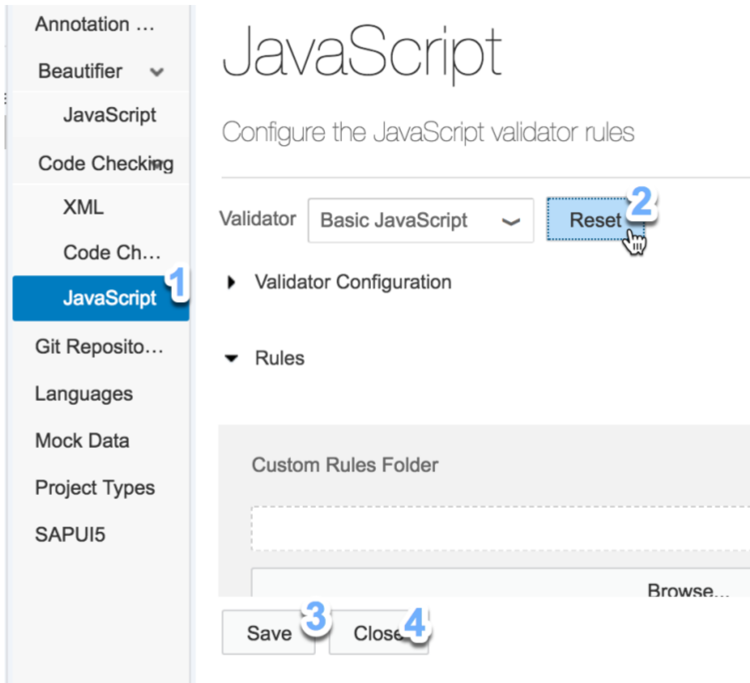
Click “Finish”
</td><td>


</td></tr>
</table>

Right click your recently created project folder, select Project Settings.



Select Code Checking > JavaScript > Reset > Save and Close.



Click the eye-icon to view the hidden files.

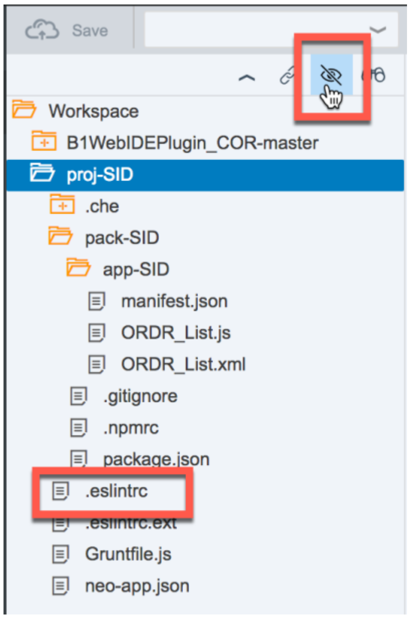
Open the .eslintrc file at the root level of your project.

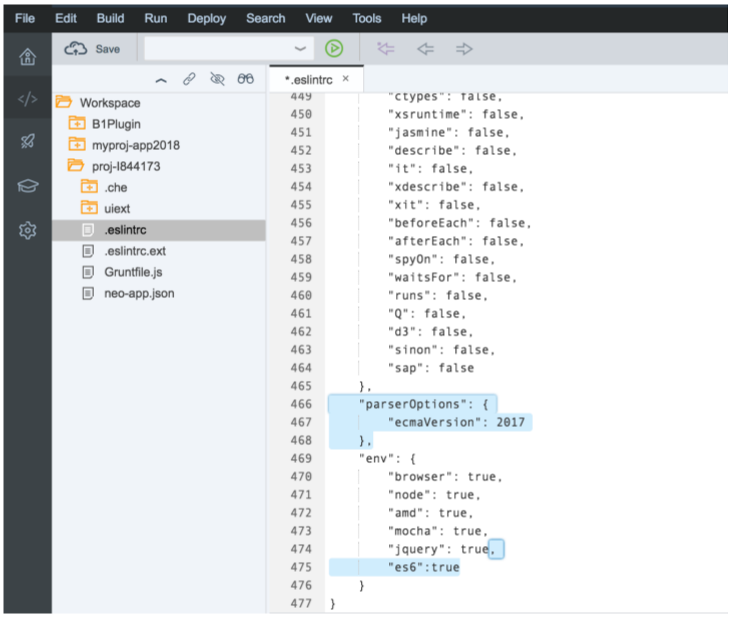
Add the following piece of code to activate the correct version for the code validation, as indicated in the image in the right side:

"parserOptions": {
    "ecmaVersion": 2017
},

,"es6": true

SAVE IT!



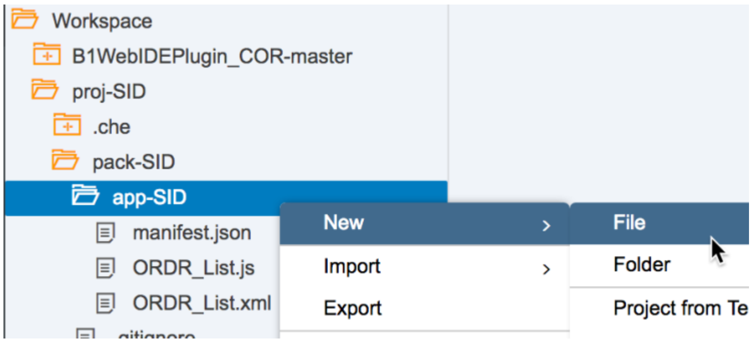


Now let's implement an extension which will automatically fill the delivery date, if it has not yet been filled, before saving the new sales order. To accomplish that we need to handle the button event which creates the sales order and run the custom code.

The custom code will live in a new file named "ORDR_New.js"

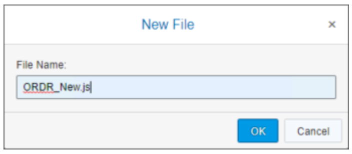
<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Right click the folder proj-<snumber>/pack-<snumber>/app-<snumber>

Select "New" > "File"</td><td>



</td></tr>
<tr><td>Name it "ORDR_New.js" and hit "OK".

This file will handle the Sales Orders Detail View</td><td>



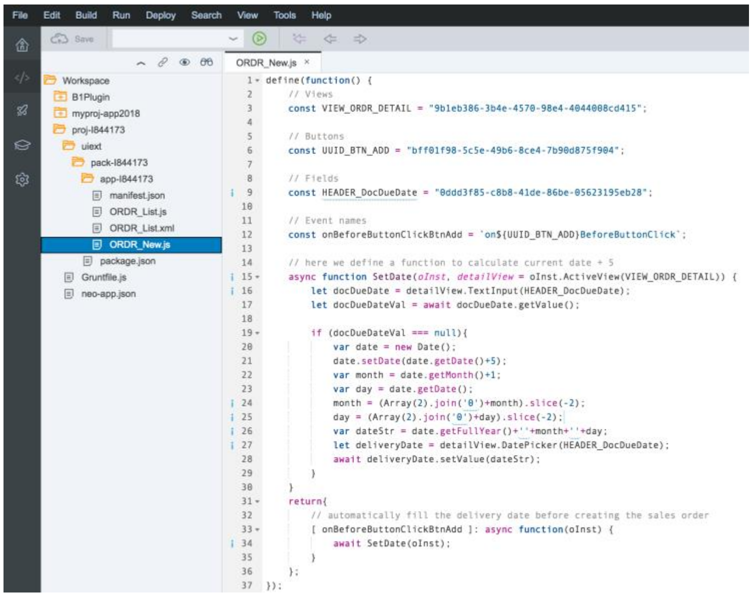
</td></tr>
<tr><td>Time to write the JavaScript with your custom logic.

Open the "ORDR_New.js" file that you've recently created.

Copy and paste the code from the link below:

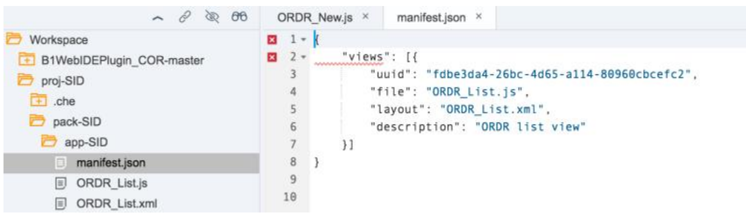
https://goo.gl/iQD82Q

Save your changes.</td><td>

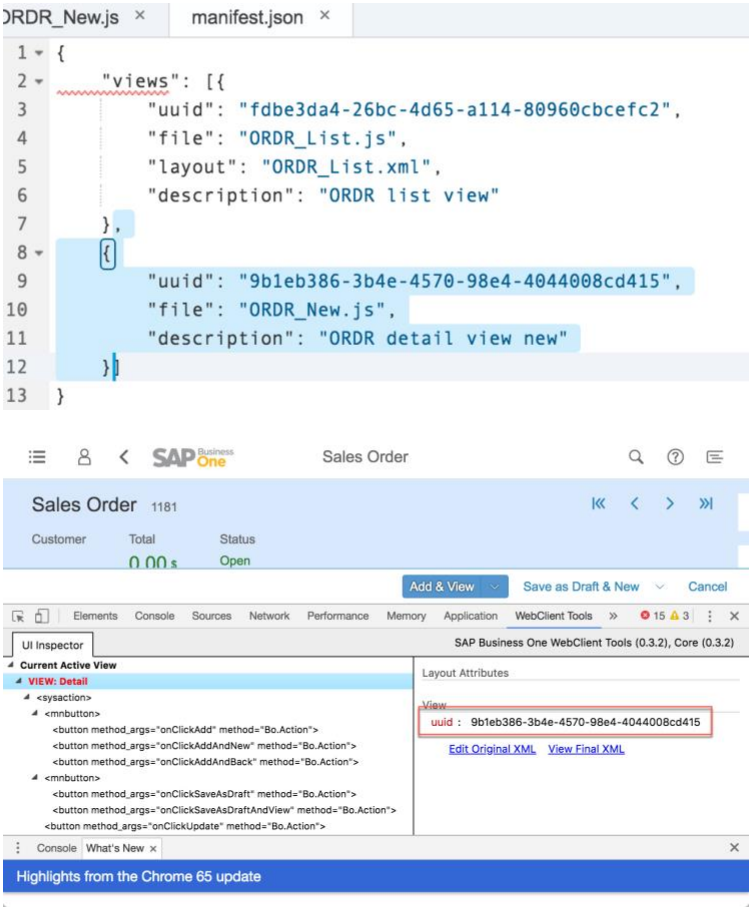


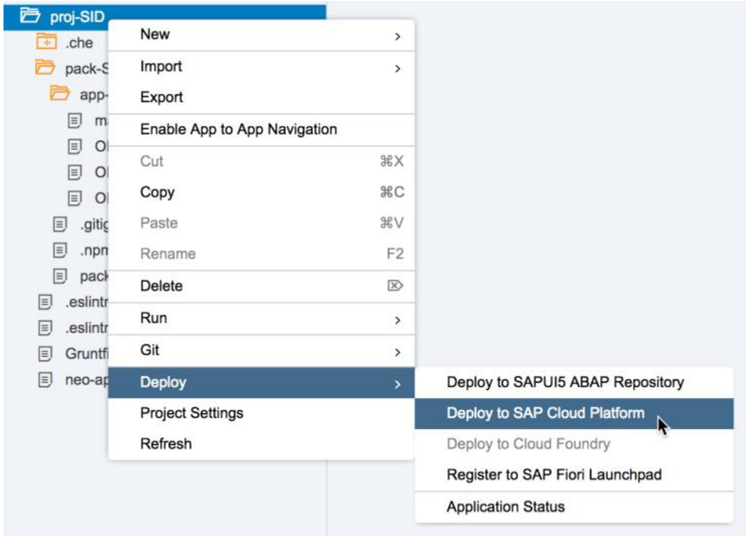
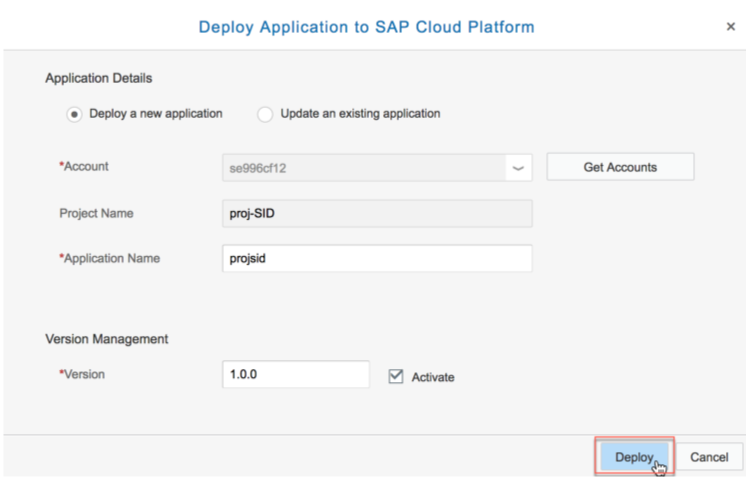
</td></tr>
<tr><td>We need to let our application know that this new file will handle the Sales Order Detail View. To accomplish that, open the manifest.json file and switch to the "Code Editor".

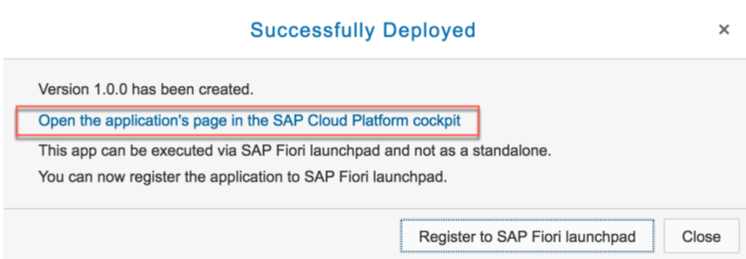
Note: just ignore the errors displayed in the manifest.json – this is an unreleased version of the web client.</td><td>



</td></tr>
</table>

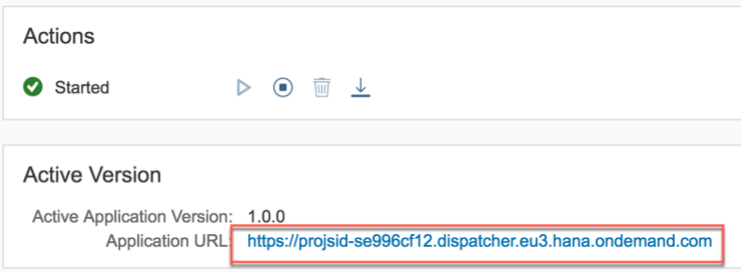
Explanation	Screenshot
Access below link to copy and paste the code to the manifest.json file “views” array: https://goo.gl/7dMHB3 Let’s understand each field: **“uuid”:** this is the universally unique identifier of the order detail view *(you saw how to get this data during the prerequisites configuration steps)* **“file”:** this is the JavaScript file where you will code your customization **“description”:** this is a description for the new object Save the file.	

<table>
<tr><td>Your first extension is ready to be deployed and tested.
Right click the project main folder (myproj-app2018) > Deploy > Deploy to SAP Cloud Platform</td><td>
</td></tr>
<tr><td>Hit the Deploy button</td><td>
</td></tr>
<tr><td>You will see the message box saying it’s successfully deployed, don’t close it.

Instead, click the link “Open the application’s page in the SAP Cloud Platform cockpit”.</td><td>
</td></tr>
<tr><td>If the “News and Announcements” pops up on your screen, just hit close.</td><td></td></tr>
</table>

The cockpit will open with the runtime details of your application deployed on SAP Cloud Platform.

Copy the application URL.



In order to confirm the application is up and running, open a new tab on Chrome and paste the URL.

Append the path to the ***package.json*** file of your project (*/pack-SID/package.json*) into the URL and hit “enter”.

Copy the full URL from the browser address bar.

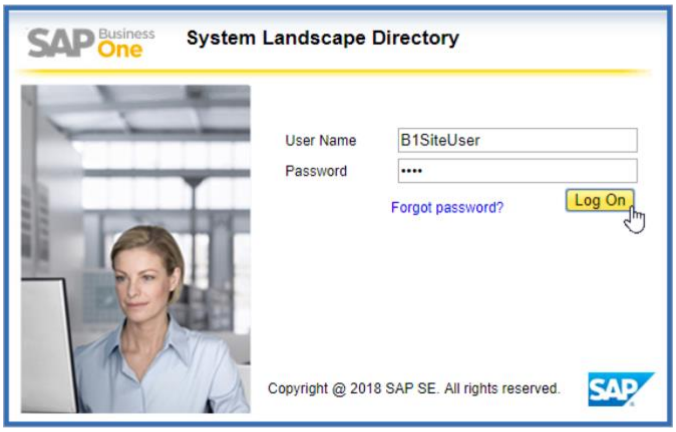
You will need it to register the app into SAP Business One Web Client.

Open the Extension Manager:

http://goo.gl/9UtVW6

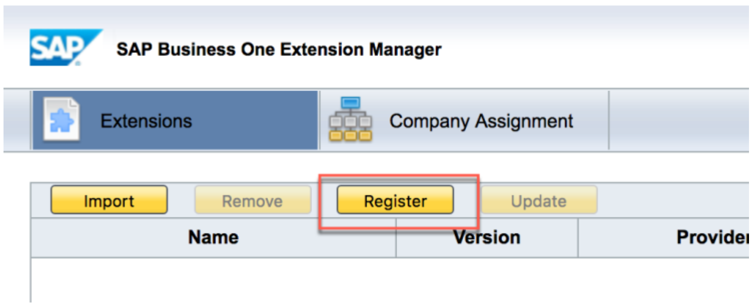
Use the following credentials:

User Name: B1SiteUser
Password: 1234



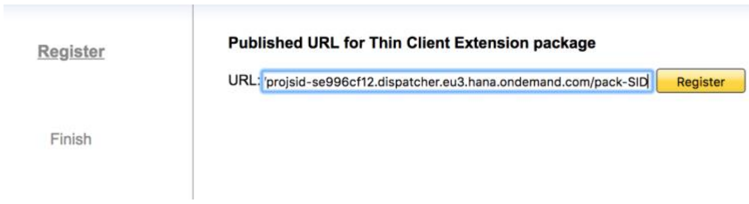
You will see this form.

Click “Register”

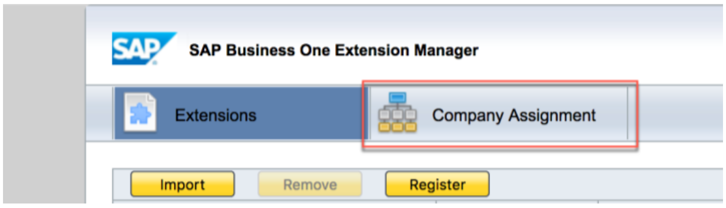


Paste the URL you copied before and remove the following piece: **"/package.json"**

click "**Register**"
click "**Next**"
click "**Finish**"



Select "Company Assignment"

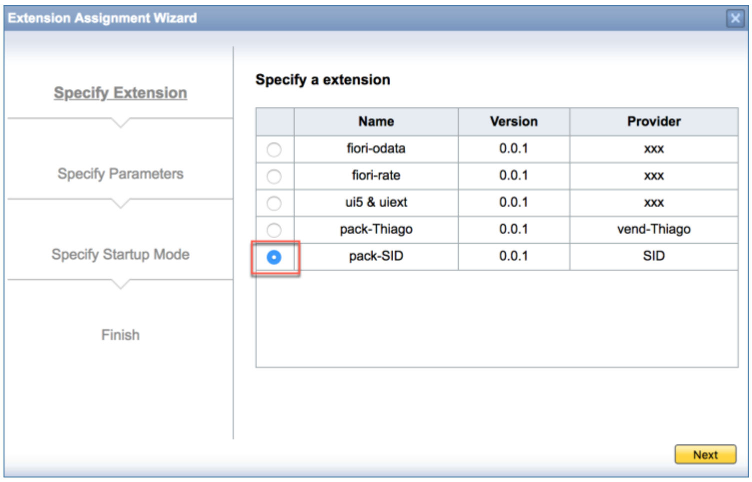


Choose your tenant

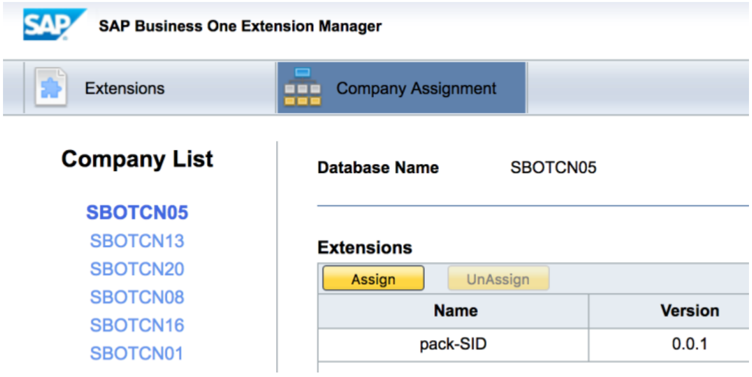
Click "Assign"

Select your extension.

Keep all the default options for the following wizard screens, until it finishes.



You should see that your extension is registered and assigned to the chosen company.



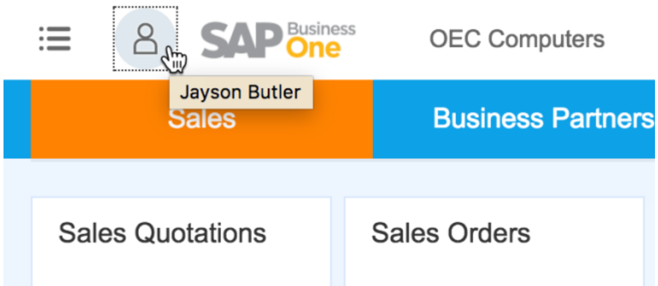
Browse to the B1 Web Client.

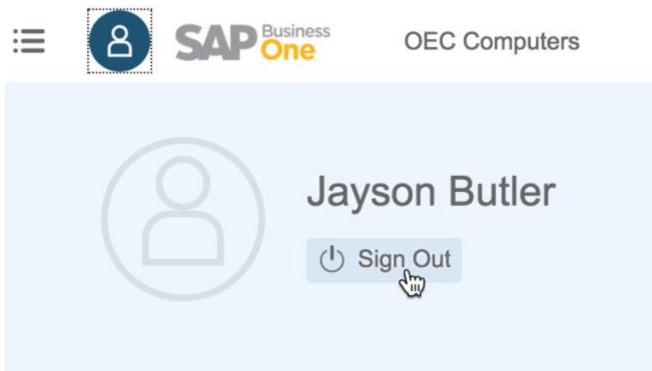
https://goo.gl/GfWDrD

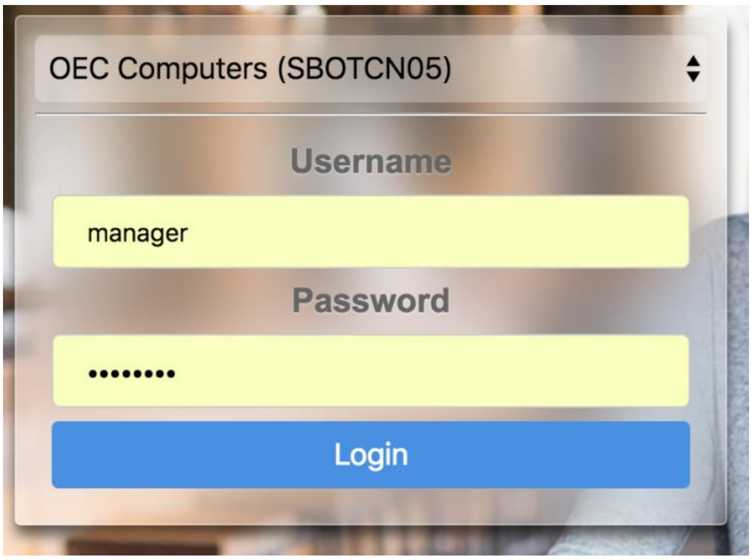
If you're already logged in, please sign off.

Select your tenant (NOTE: IT SHOULD BE THE SAME OF THE ONE YOU HAVE ASSIGNED YOUR APP TO, IN THE PREVIOUS STEPS).

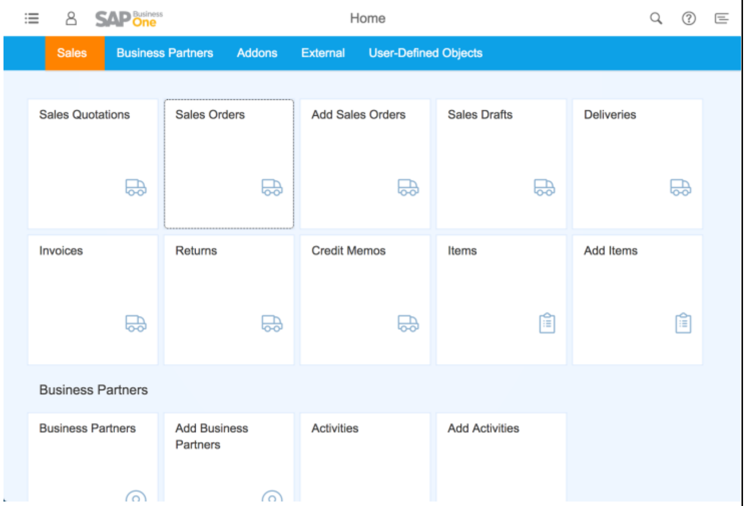
Login using below credentials:
Username: manager
Password: Initial0



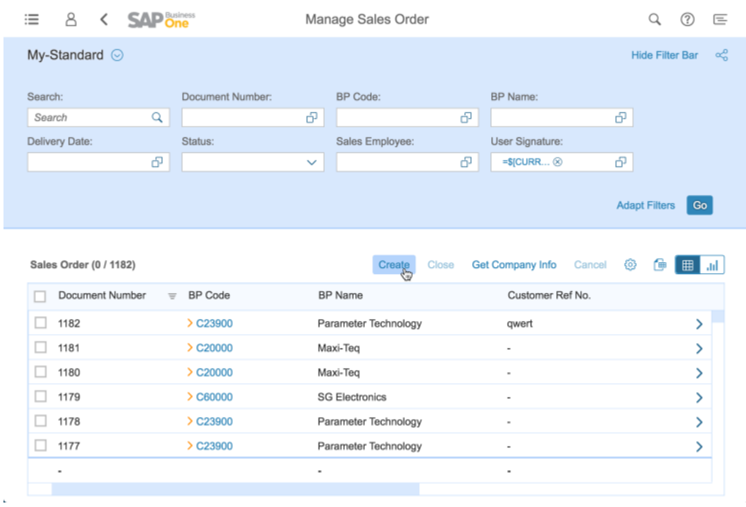




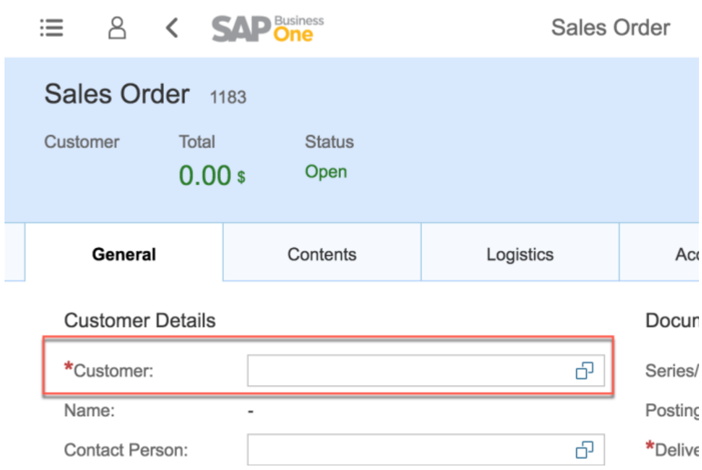
Click the “Sales Orders” tile



Click “Create”

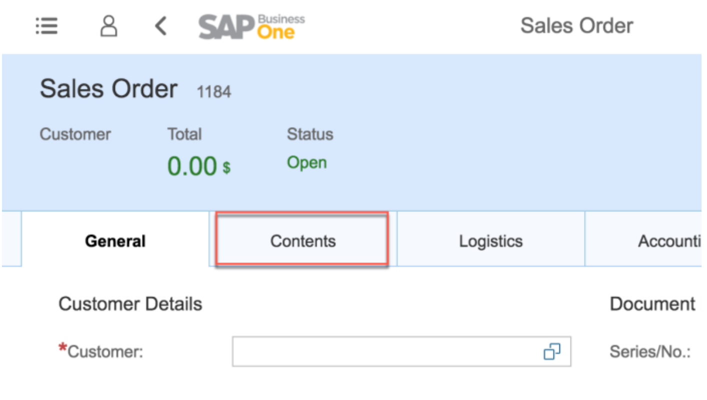


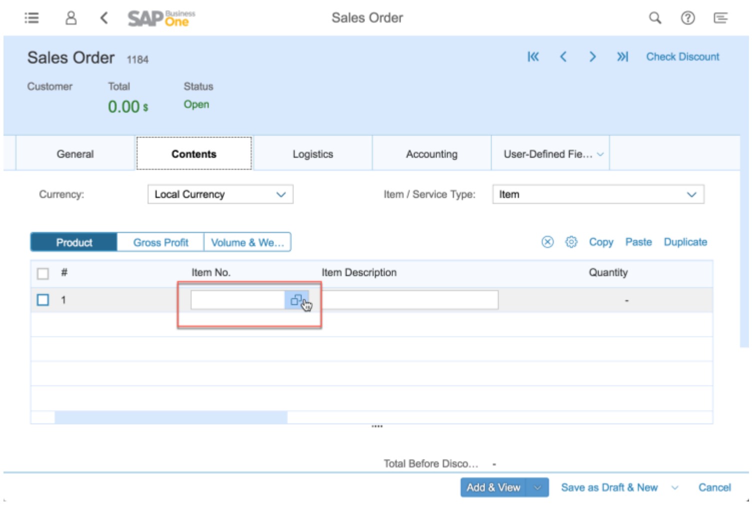
Select the customer from the choose from list (suggested: Maxi-Teq or Parameter Technology)



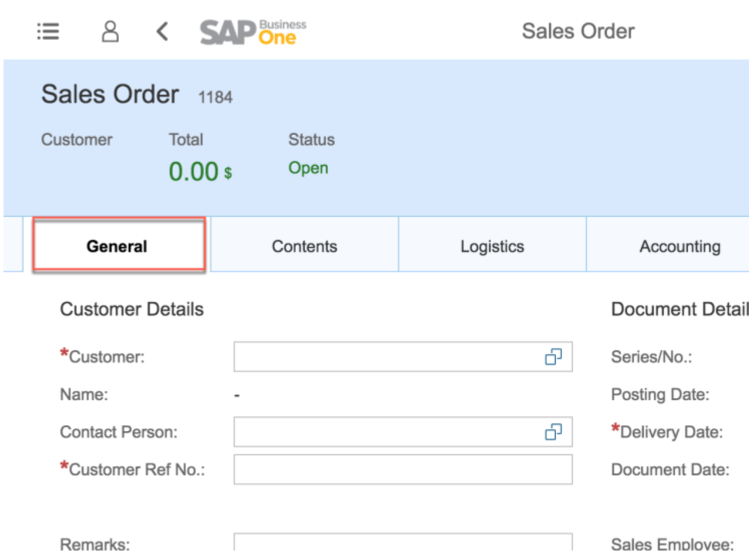
Browse to the “**Contents**” tab, add an item selecting the choose from list

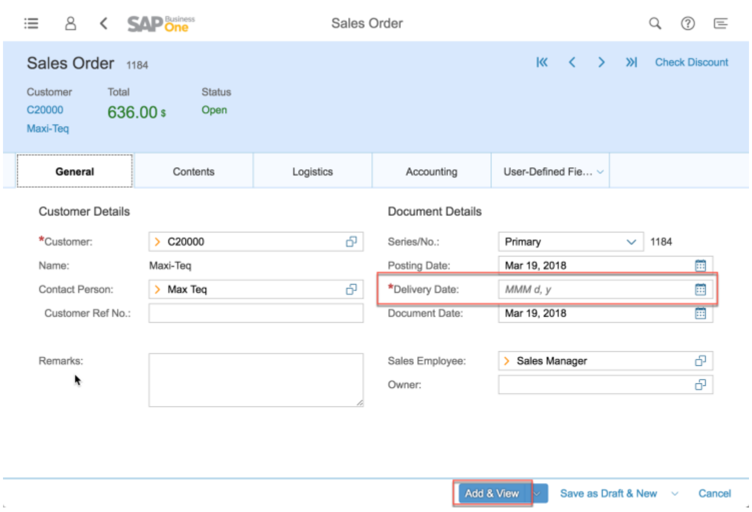
Click “**Select**”

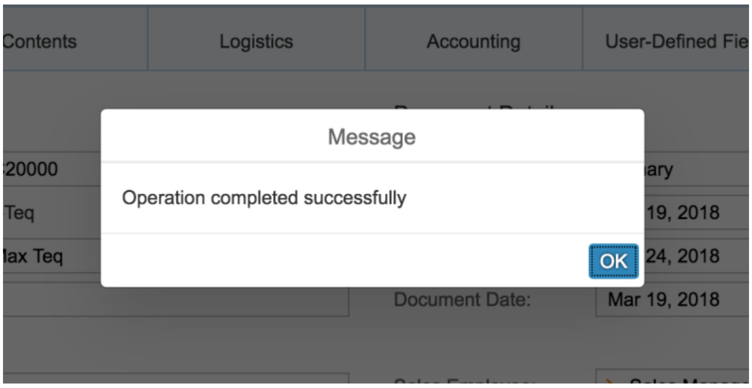




Browse back to the “General” tab



<table>
<tr>
<td>Check that the Delivery Date is empty and click the “Add” button to create the Sales Order</td>
<td>

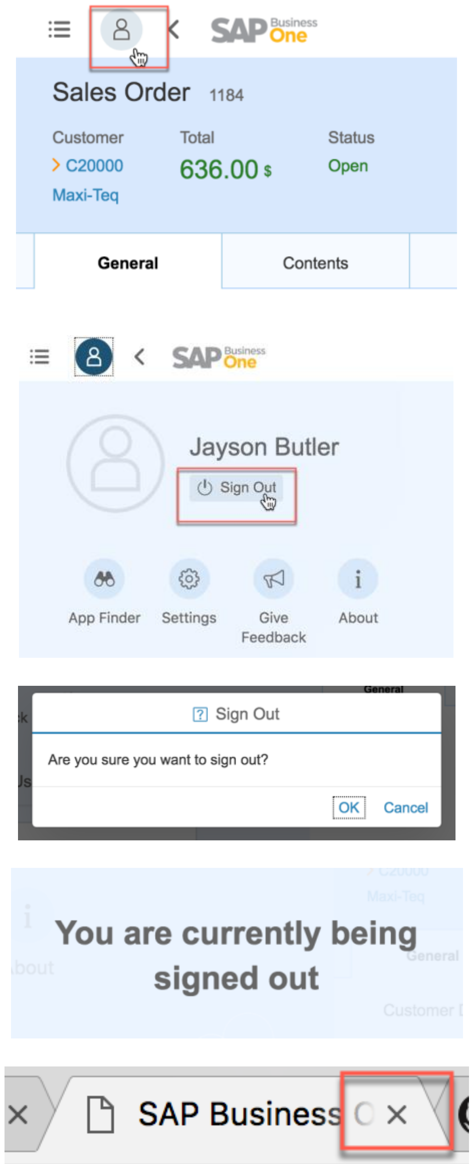

</td>
</tr>
<tr>
<td>Validate that the Sales Order has been successfully created and that the Delivery Date is the current date + 5 days.</td>
<td>
</td>
</tr>
<tr>
<td>Congratulations, you have finished the first exercise!</td>
<td></td>
</tr>
</table>

Before continuing, please log out from SAP Business One Web Client, and close the window:

<table>
<tr>
<td>Click the user icon

Select “Sign Out”

Confirm the message

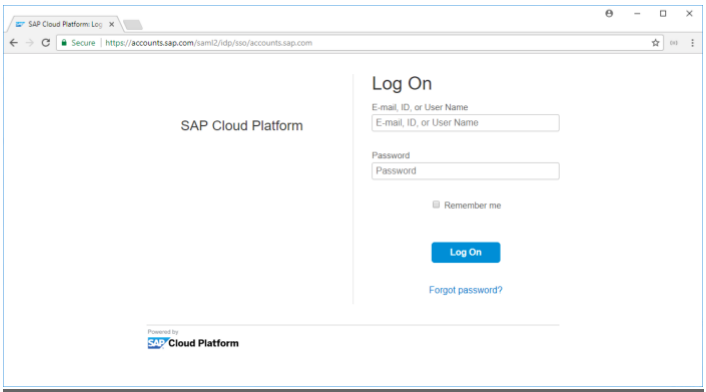
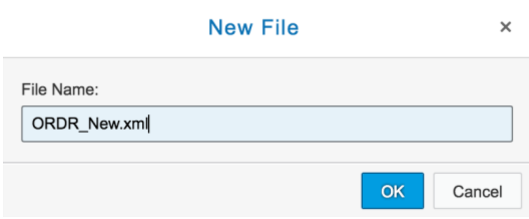
Close the browser</td>
<td>
</td>
</tr>
</table>

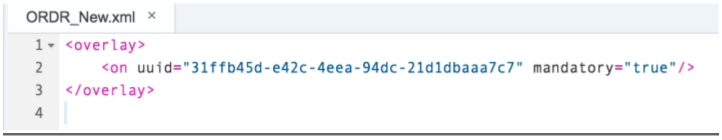
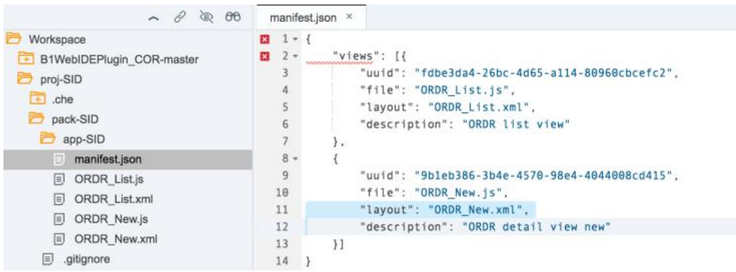
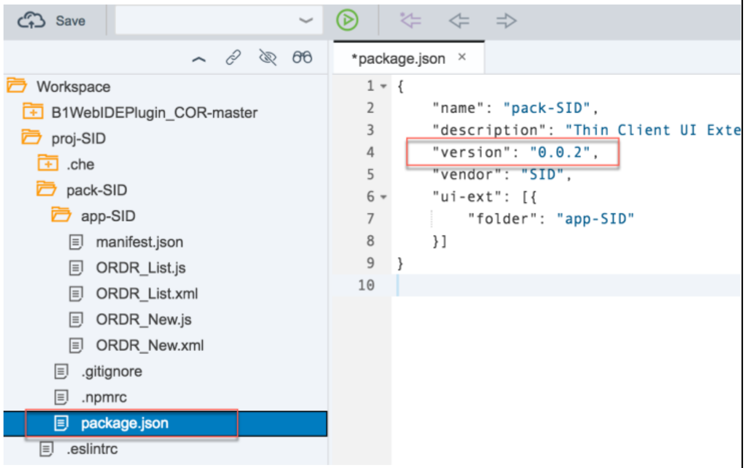
EXERCISE 2: MAKE FIELD “CUSTOMER REFERENCE NUMBER” MANDATORY

For this second exercise, we want to make an existing field mandatory.

To accomplish changes in the UI components, we need an XML file with the corresponding components and properties.

In this case, we will create a file named “ORDR_New.xml”, reference this file in the “manifest.json” and deploy a new version of our existing app.

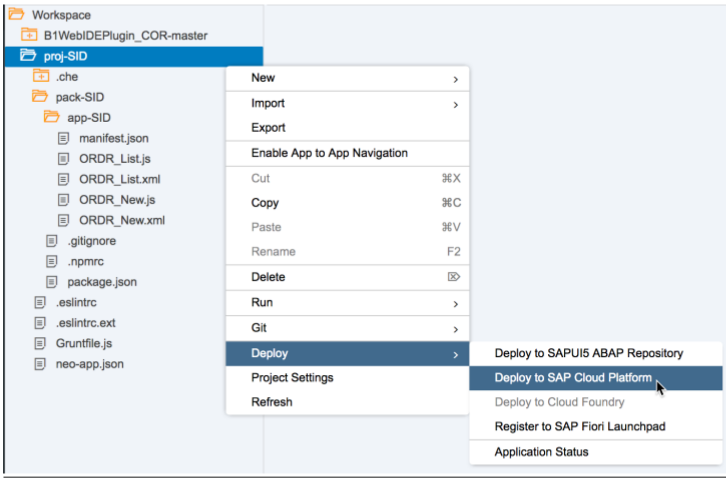
<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Open Google Chrome.</td><td></td></tr>
<tr><td>Access the SCP Web IDE using your S-number credentials.
https://goo.gl/PZqa1Y</td><td>
</td></tr>
<tr><td>At the app level, create a new xml file to handle the view changes.</td><td>
</td></tr>
<tr><td>Name it ORDR_New.xml</td><td>
</td></tr>
</table>

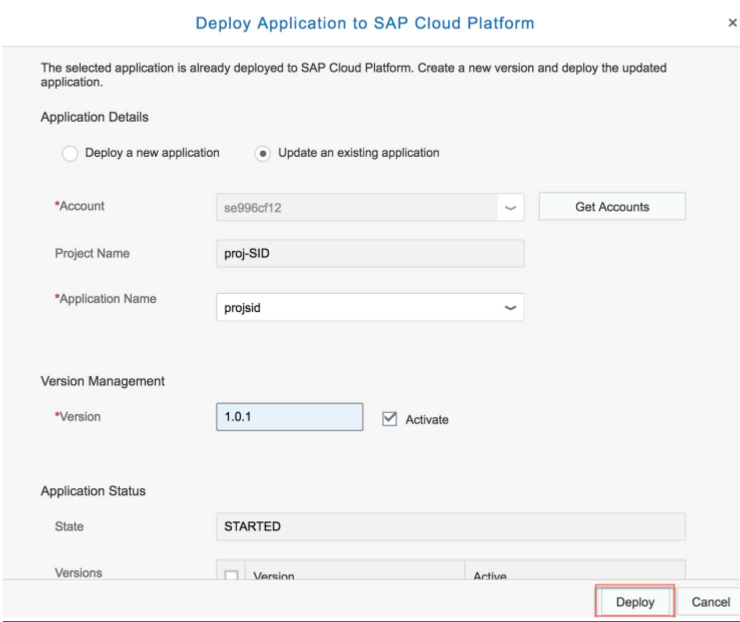
Explanation	Screenshot
Add the XML to make the field “Customer Reference Number” mandatory https://goo.gl/597DbH Save it.	
Now open the manifest.json file and add a reference to the new xml file in the existing object. https://goo.gl/tVFCQ2 Save it.	
Open the package.json file and update the version of your app, in this case it will be 0.0.2 Save it.	

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>
Your changes are ready to be deployed and tested.

Right click the project main folder > Deploy > Deploy to SAP Cloud Platform

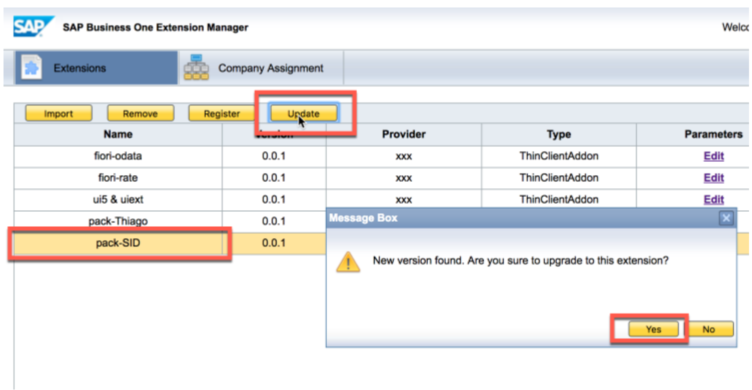
It will update the existing application.

Click the “Deploy” button.
</td><td>


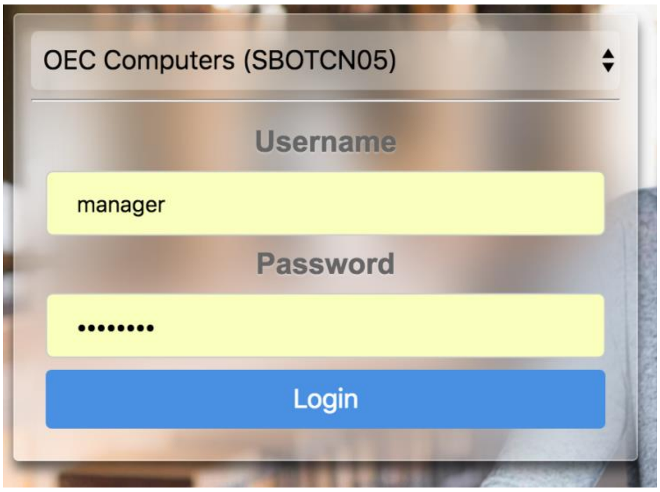
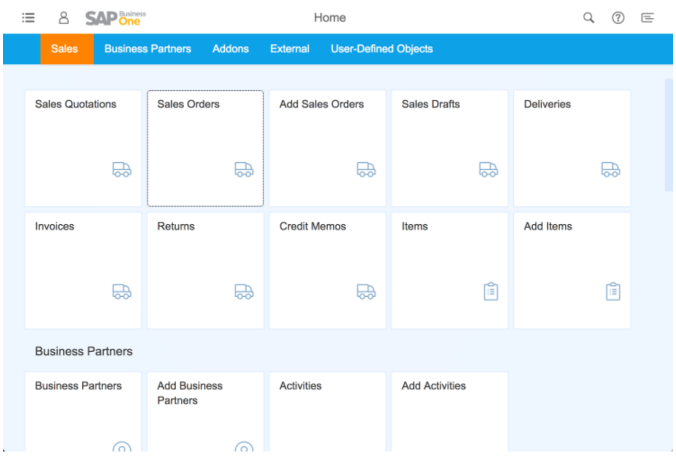
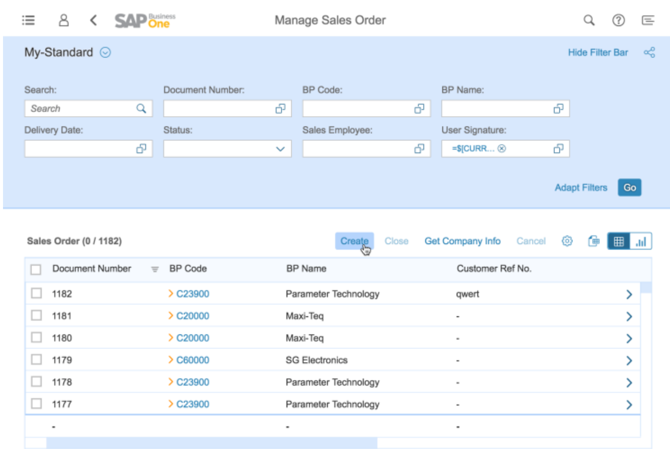


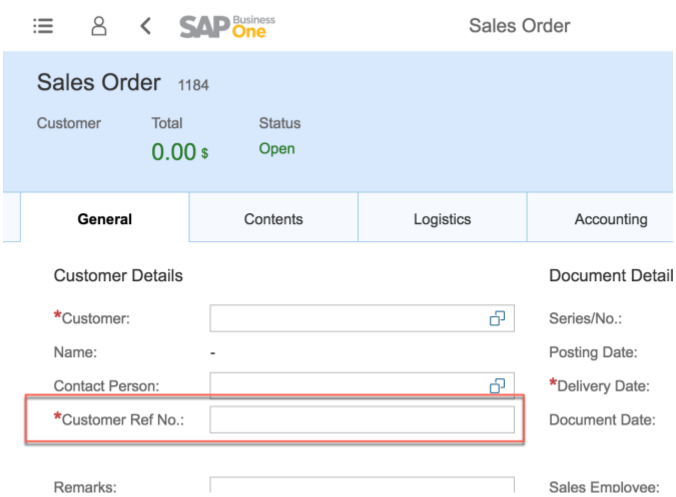
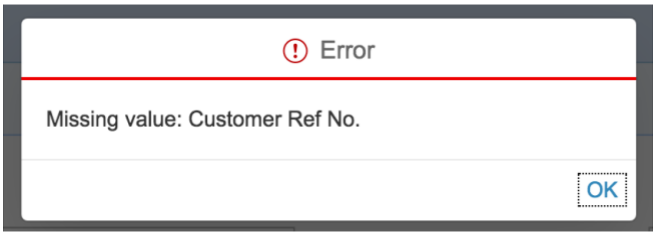
</td></tr>
<tr><td>
Now, before testing the changes, you need to tell SAP Business One, that your app is updated.

Open the Extension Manager, select your app and click “Update”.

Once it’s updated, the screen will refresh and the Version must match with the version you’ve informed in the package.json file.
</td><td>


</td></tr>
</table>

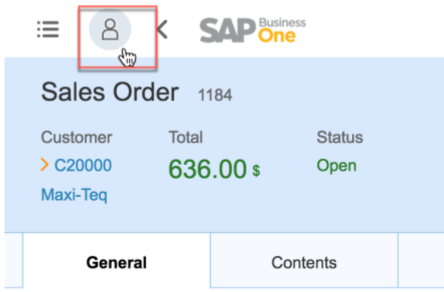
Explanation	Screenshot
Open the SAP Business One Web Client >> https://goo.gl/GfWDrD Company: make sure you are connecting to the correct tenant Username: **manager** Password: **Initial0** Click Login	
Click the Sales Orders tile.	
Click "Create"	

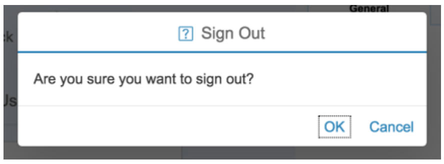
<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Check that the field “Customer Reference Number” is now marked with the red asterisk, indicating it’s now a mandatory field.</td><td>
</td></tr>
<tr><td>Try to create a Sales Order without the Customer Ref No. and you will receive an error.</td><td>
</td></tr>
</table>

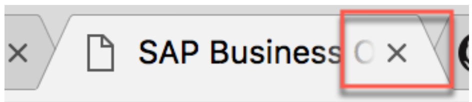
Congratulations!

Before continuing, please log out from SAP Business One Web Client, and close the window:

<table>
<tr>
<td>
Click the user icon
Select “Sign Out”
Confirm the message
Close the browser
</td>
<td>





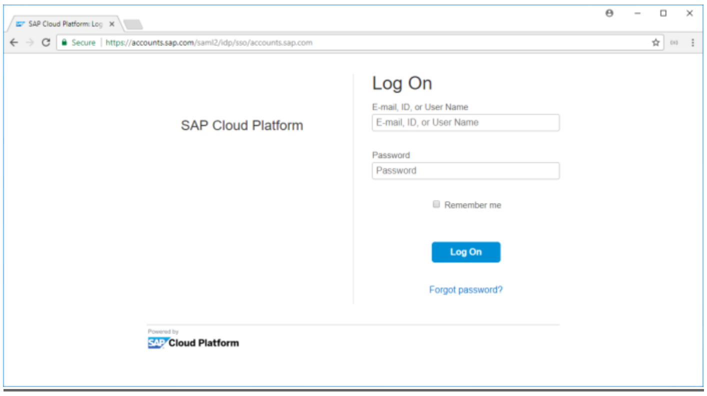


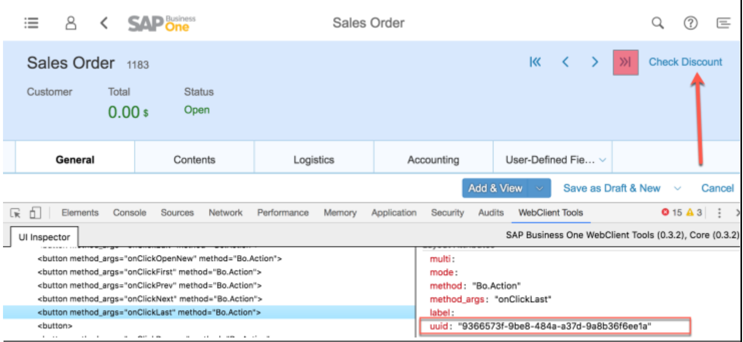
</td>
</tr>
</table>

EXERCISE 3: ADD A BUTTON TO THE SALES ORDER TO CHECK BUSINESS PARTNER'S DEBTS

As a last exercise, you will create a new button to the existing Sales Order UI.
You will handle the button click to check if the customer has pending debts. If there's no pending debts, alert the user and apply 20% discount to the current Sales Order.

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Open Google Chrome.</td><td></td></tr>
<tr><td>Access the SCP Web IDE using your S-number credentials.

https://goo.gl/PZqa1Y</td><td>
</td></tr>
<tr><td>Let's add a new button to the existing Sales Order UI.

It should be called "Check Discount" and placed after the "Last Record" navigation button.</td><td>
</td></tr>
<tr><td>Open the ORDR_New.xml and add the new button to the form.

https://goo.gl/h9MBC3</td><td>ORDR_New.xml

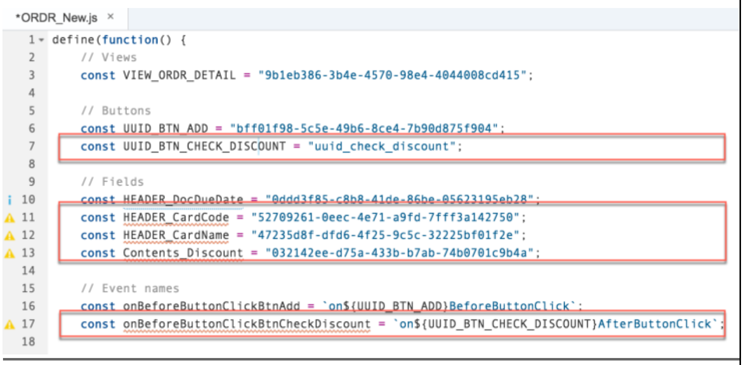
```
<overlay>
    <on uuid="31ffb45d-e42c-4eea-94dc-21d1dbaaa7c7" mandatory="true"/>
    <after uuid="9366573f-9be8-484a-a37d-9a8b36f6ee1a">
        <button text="Check Discount" uuid="uuid_check_discount"></button>
    </after>
</overlay>
```
</td></tr>
</table>

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Open the ORDR_New.js file and declare a constant for the new button.

Declare a constant for the event you need to handle.

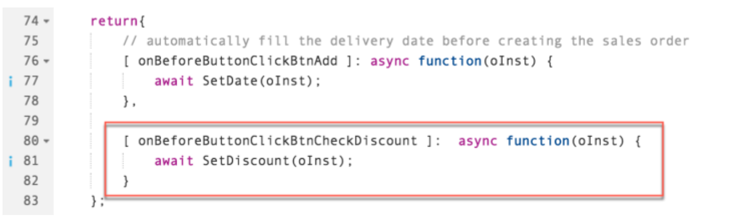
Declare a constant for the discount, CardCode and CardName fields.

You can copy and paste the new code lines from here:

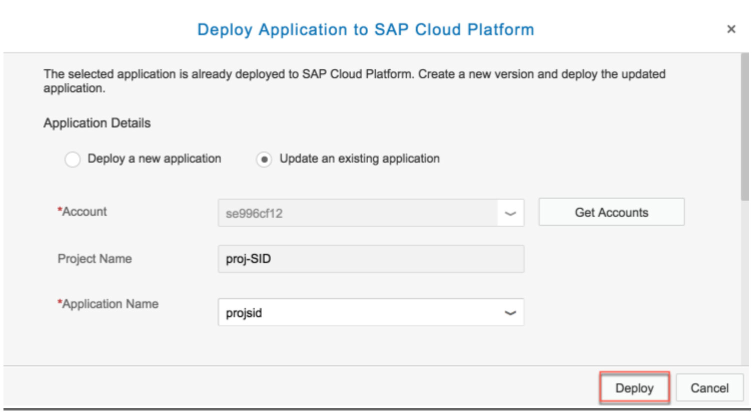
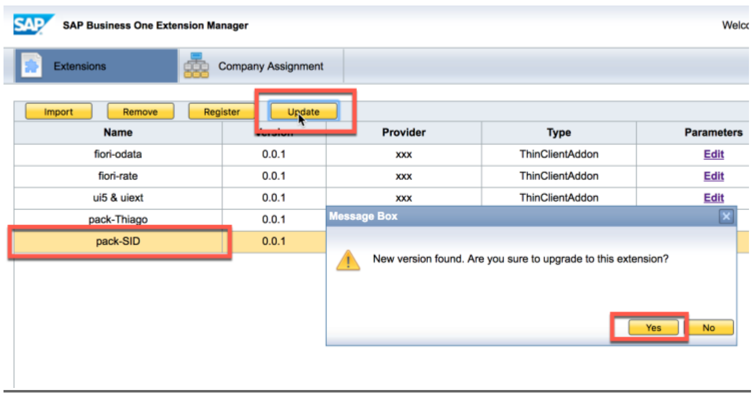
https://goo.gl/AxokcH</td><td>
</td></tr>
<tr><td>Let's start coding the decision logic for the discount. Append the following function, just after the SetDate function (note the call to the Service Layer)

https://goo.gl/x8BRhc</td><td>
</td></tr>
<tr><td>Now append the event handler of the new button

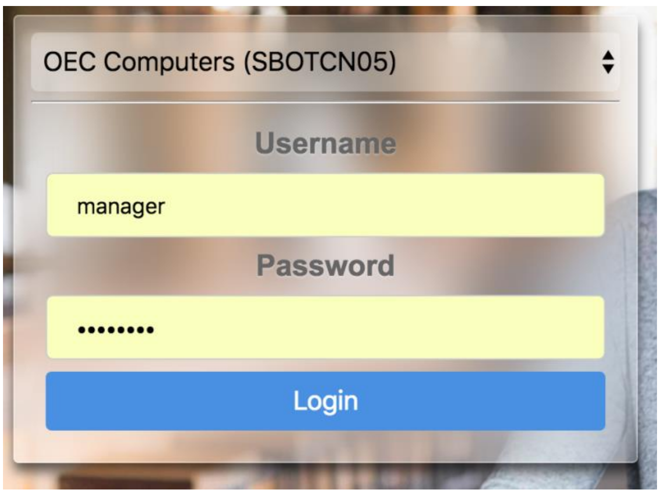
https://goo.gl/RxPNMc

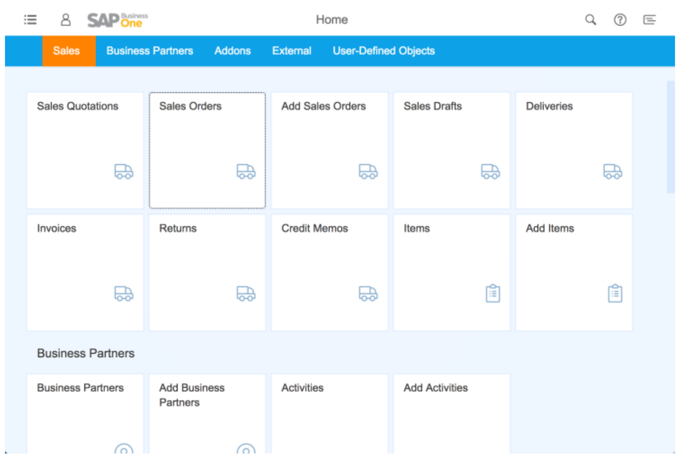
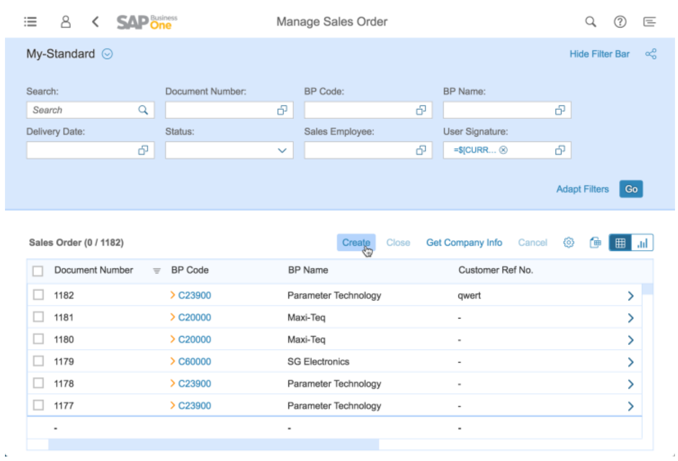
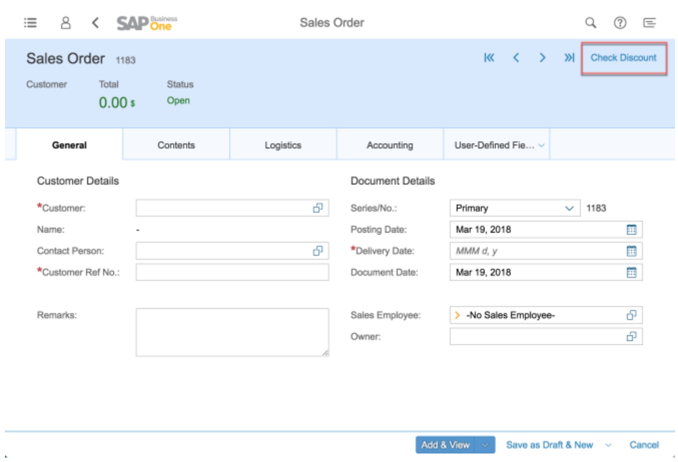
Save it!</td><td>
</td></tr>
<tr><td>Open the package.json file and update the version of your app, in this case it will be 0.0.3

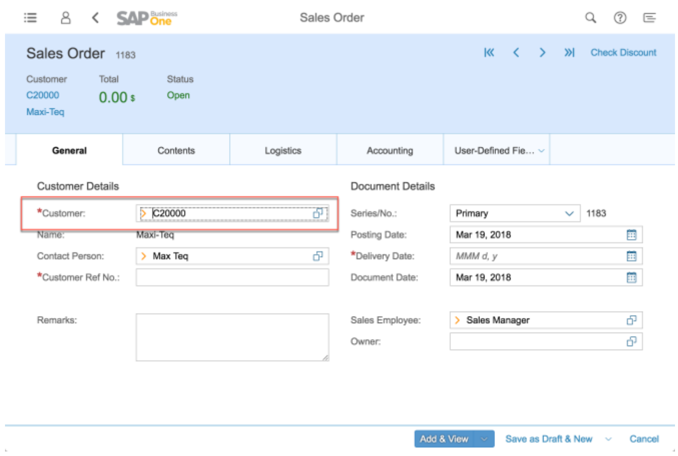
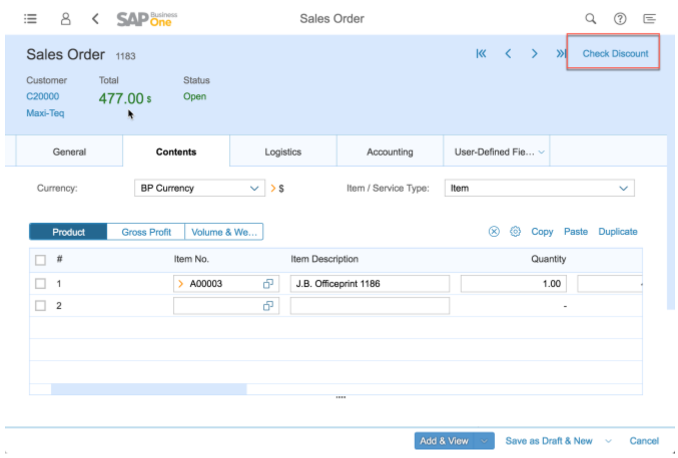
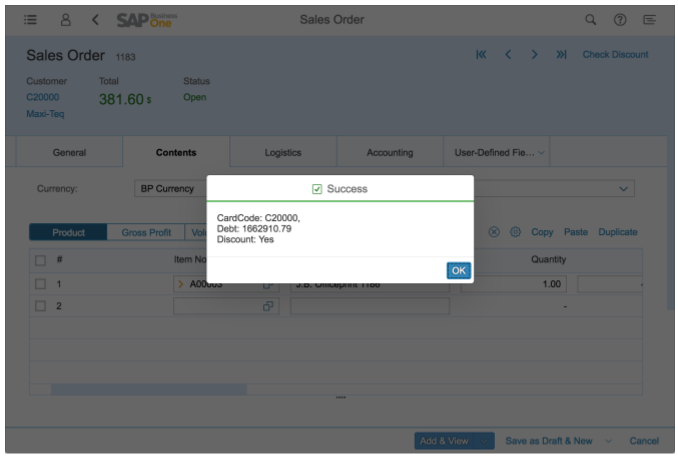
Save it.</td><td>
</td></tr>
</table>

Explanation	Screenshot
Your changes are ready to be deployed and tested. Right click the project main folder > Deploy > Deploy to SAP Cloud Platform. Click the “Deploy” button.	
Now, before testing the changes, you need to tell SAP Business One, that your app is updated. Open the Extension Manager, select your app and click “Update”. Once it’s updated, the frame will auto-refresh and the Version must match with the version you’ve informed in the package.json file.	

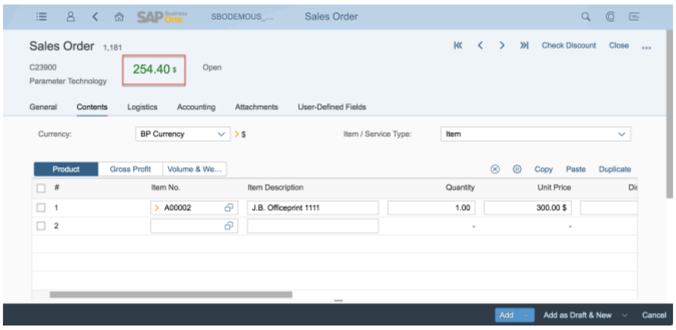
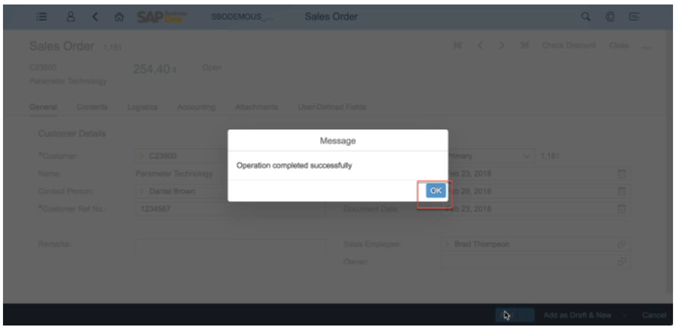
Now, test the changes.

Explanation	Screenshot
Open Google Chrome	
Open SAP Business One Web Client >> https://goo.gl/GfWDrD Company: make sure you are connecting to the correct tenant Username: **manager** Password: **Initial0** Click Login	

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Click the Sales Orders tile.</td><td>
</td></tr>
<tr><td>Click “Create”</td><td>
</td></tr>
<tr><td>Validate that the new button “Check Discount” is displayed, don’t click it.</td><td>
</td></tr>
</table>

Explanation	Screenshot
Select a customer	
Click “Contents” tab Select an Item Look at the Sales Order total amount Click “Check Discount”	
A message will be displayed showing whether the discount is applicable. Click “OK” Look at the Total amount	

<table>
<tr><th>Explanation</th><th>Screenshot</th></tr>
<tr><td>Note the Sales Order amount, it is reduced by 20%

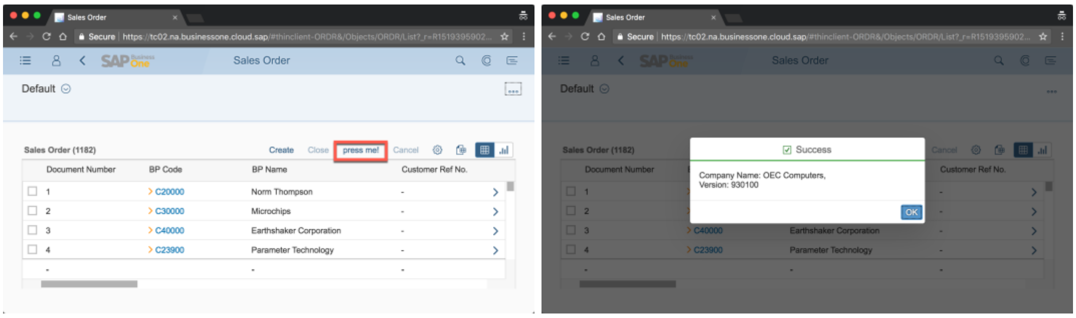
Click “Add”</td><td>
</td></tr>
<tr><td>Click “OK”</td><td>
</td></tr>
</table>

Congratulations! You have finished the exercises on the extensibility of SAP Business One Web Client. Please, give us your feedback.

APPENDIX:

The project you have just created derives from a template which comes with a very simple customization to the existing sales order UI view:

- It adds a button labeled "***press me!***", to the existing Sales Order view, which calls the service layer to retrieve the company information, see below:



If you are curious, go to folder "*proj-<snumber>*" → "*pack-<snumber>*" → "*app-<snumber>*" and open the file ***ORDR_List.js***. Review the code.

You will also see the ***manifest.json*** defining the views and files associated with each extension.

www.sap.com/contactsap

Run Simple