



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

PARTNER





TABLE OF CONTENTS

PREREQUISITES	4
i. SAP Business One Web Client DevTools.....	4
EXERCISE 1: AUTOMATICALLY FILL THE DELIVERY DATE	9
EXERCISE 2: MAKE FIELD “CUSTOMER REFERENCE NUMBER” MANDATORY.....	25
EXERCISE 3: ADD A BUTTON TO THE SALES ORDER TO CHECK BUSINESS PARTNER’S DEBTS ..	31



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

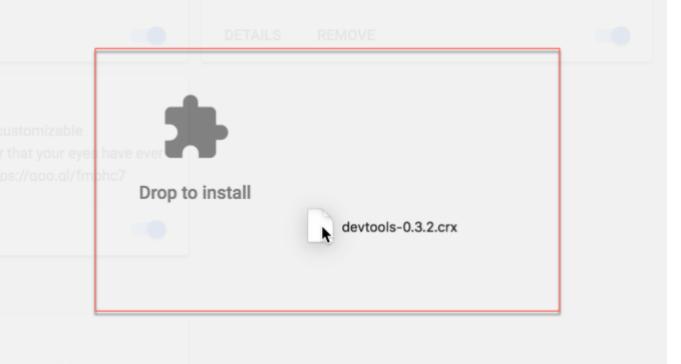
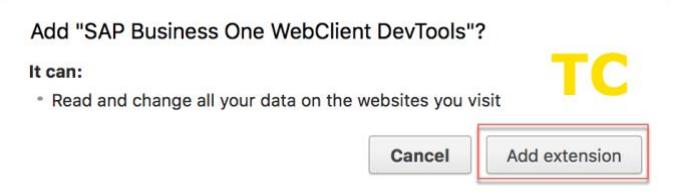
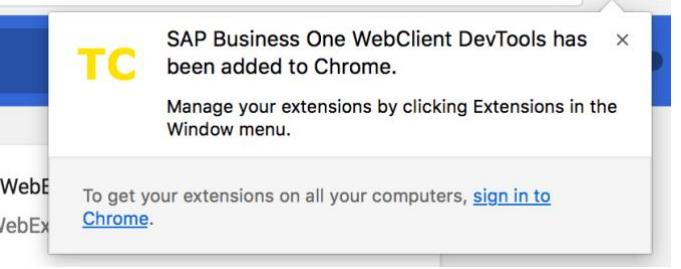
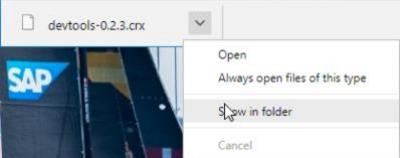
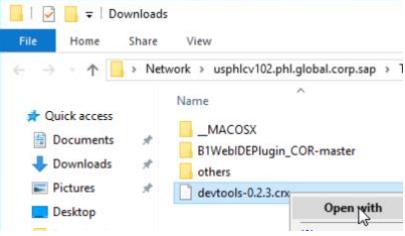


PREREQUISITES

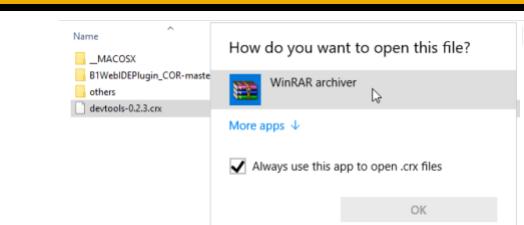
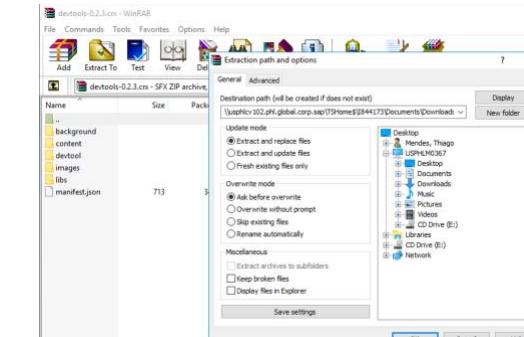
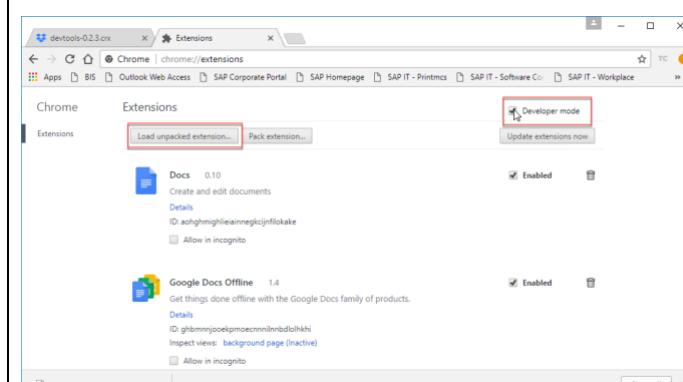
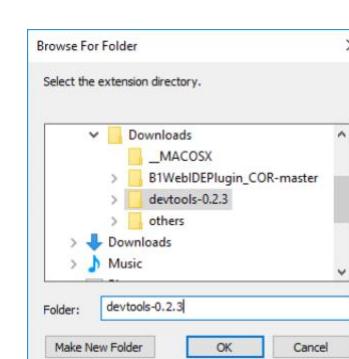
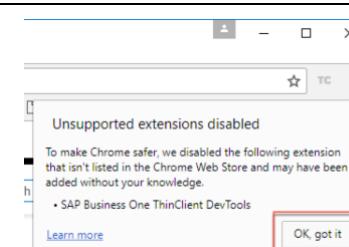
i. SAP Business One Web Client DevTools

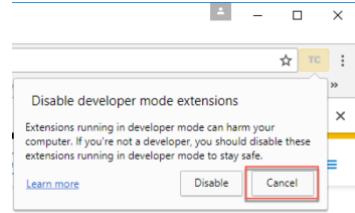
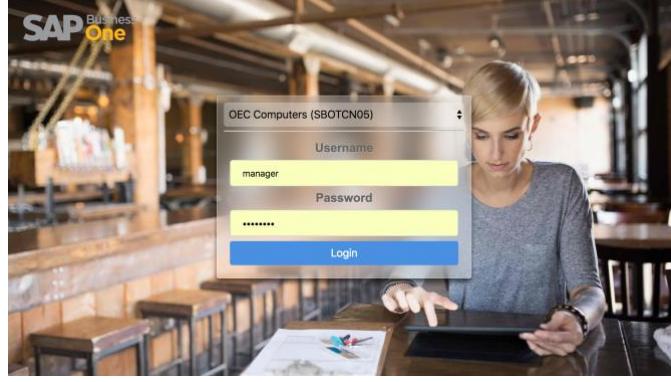
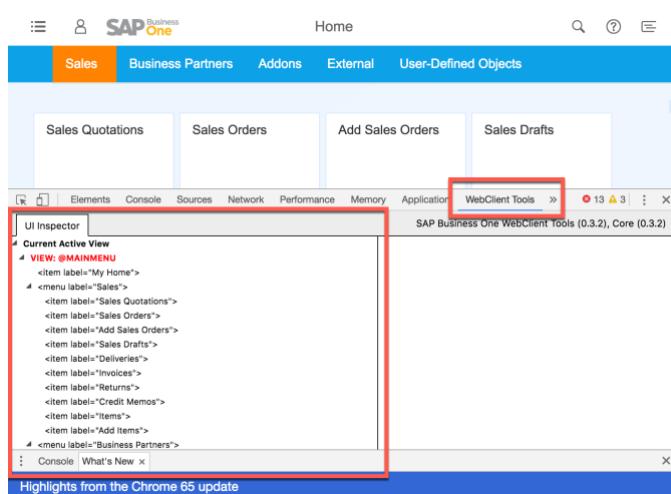
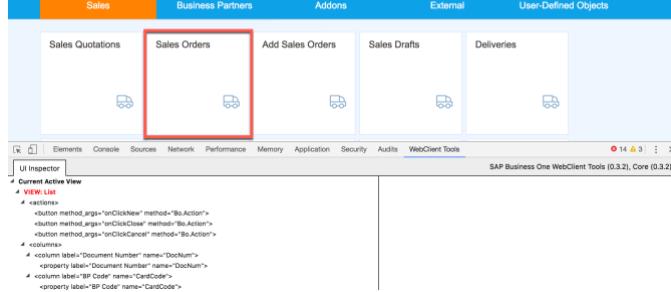
This is the SAP Business One Web Client DevTools, the Chrome Extension which enables you to get detailed technical data of the SAP Business One Web Client UI components.

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/".	

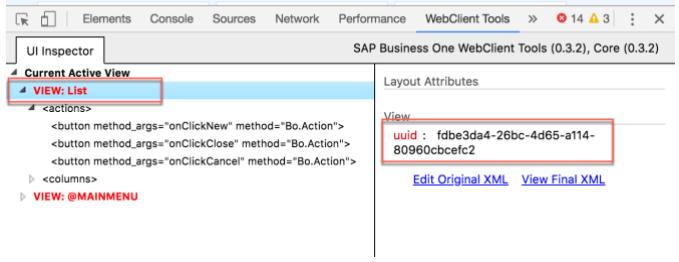
Explanation	Screenshot
<p>Drag the downloaded extension file (*.crx) and drop it into the opened tab.</p> <p>It will automatically install into your chrome extension.</p>	
<p>Click "Add extension"</p> <p>Close the message that will pop-up</p>	 
<p>Follow the next steps only if you are using windows-based computer.</p> <p>For security reasons, in chrome latest version, the extension will be disabled by default. Here are the steps to enable it:</p> <p>Extract the file you have downloaded (devtools-0.3.2.crx) to a folder.</p> <p>NOTE: the *.crx file is just a renamed ZIP file, so any archive/compression solution should be able to extract it.</p>	 



Explanation	Screenshot
	 
<p>Open the extension management in Google Chrome</p> <ol style="list-style-type: none">enable 'developer mode'click 'Load unpacked extension...'fill the path you extracted in step 1.restart Google Chrome	 
When the chrome restart, you need to confirm not to disable the DevTools extension.	

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



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 <i>Universally Unique Identifier</i> 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.



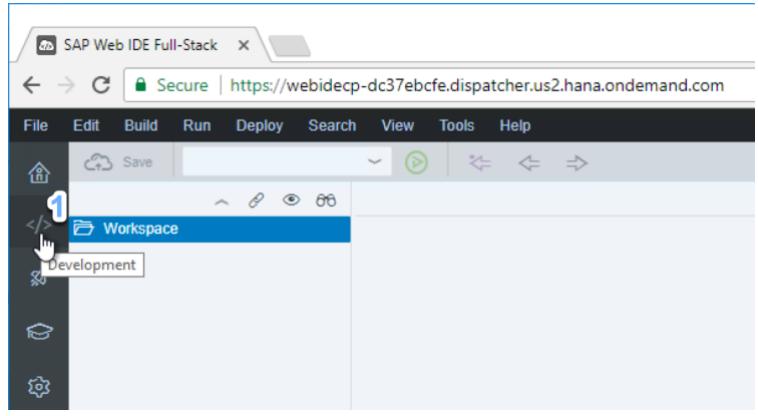
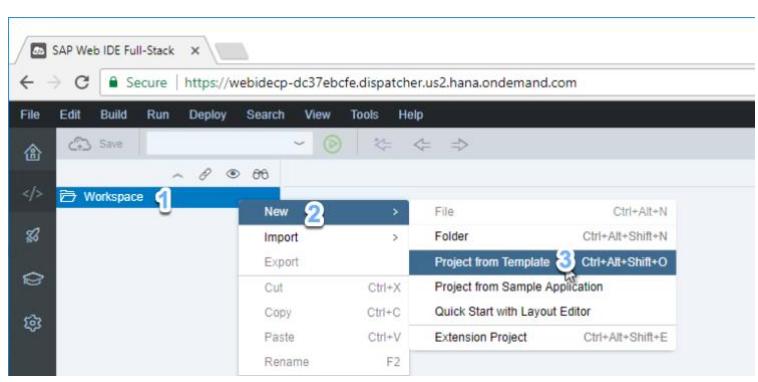
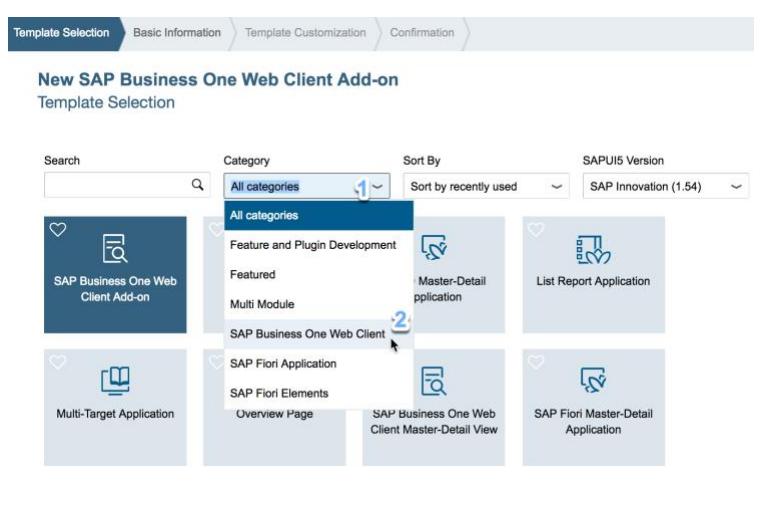
EXERCISE 1: AUTOMATICALLY FILL THE DELIVERY DATE

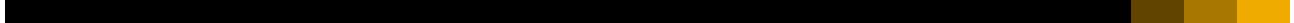
Let's get started, coding your first loosely-coupled application on top of SAP Business One Web Client.

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).

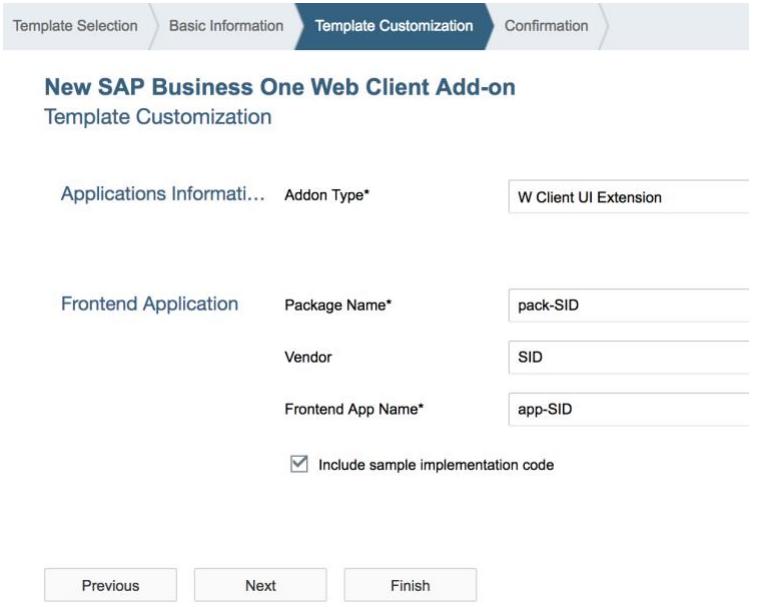
Explanation	Screenshot
<p>Open Chrome and access SCP Web IDE using your S-number credentials. https://goo.gl/PZqa1Y</p>	
<p>Select the Preferences perspective; Click Features; Search for “business one”; Enable the SAP Business One feature.</p> <p>DO NOT FORGET TO SAVE AND REFRESH THE PAGE!</p>	

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

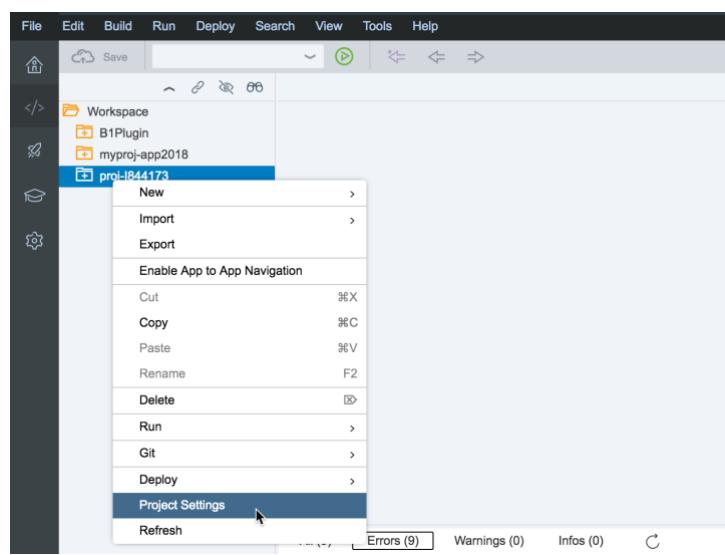


Explanation	Screenshot
<p>Select the SAP Business One Web Client Add-on and hit “Next”</p>	<p>New SAP Business One Web Client Add-on Template Selection</p> <p>Search: SAP Business One Web C</p> <p>Category: SAP Business One Web C</p> <p>Sort By: Sort by recently used</p> <p>SAPUI5 Version: SAP Innovation (1.54)</p> <p>SAP Business One Web Client Add-on</p> <p>3</p> <p>4</p>
<p>Fill in the Basic Information as bellow:</p> <p>Project Name: <i>proj-<s-number></i> Click “Next”</p>	<p>Template Selection Basic Information Template Customization Confirmation</p> <p>New SAP Business One Web Client Add-on Basic Information</p> <p>Project Name*</p> <p>proj-SID</p> <p>Previous</p> <p>Next</p>

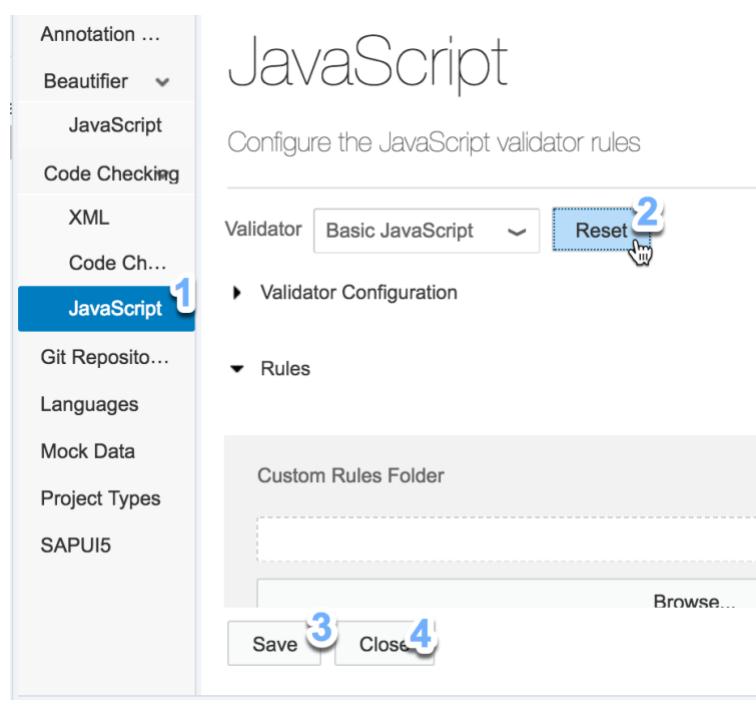


Explanation	Screenshot
<p>Fill in the Template Customization as bellow:</p> <p>Add-on Type: W Client UI Extension Package Name: pack-<snumber> Vendor: <snumber> Frontend App Name: app-<snumber></p> <p>(Note: DO NOT FORGET THE DASH for the Frontend App Name)</p> <p>Check the “Include sample implementation code” option. (Refer to the appendix section to get further information on the sample code)</p> <p>Click “Finish”</p>	

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.

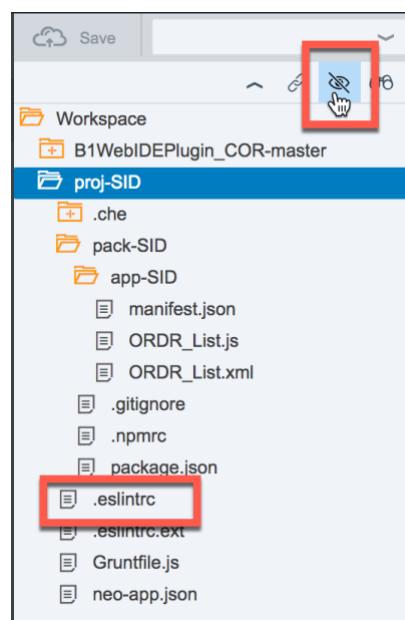
Open the .eslintrc file at the root level of your project.

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!

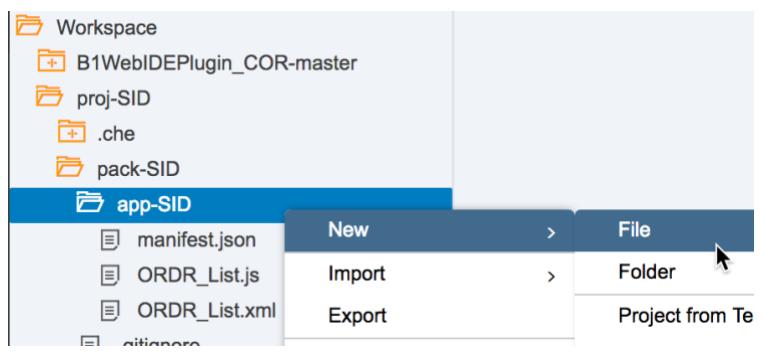
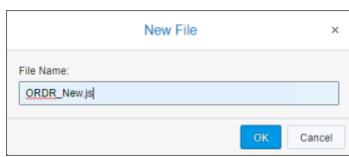
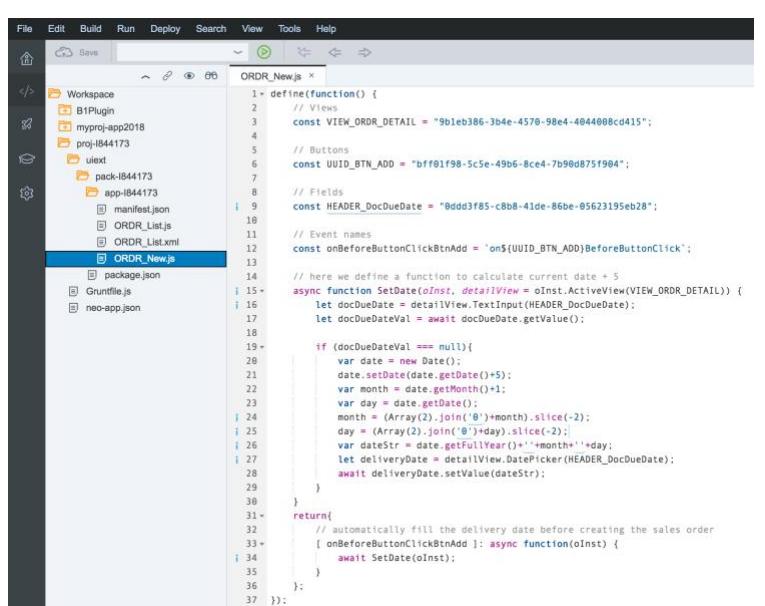
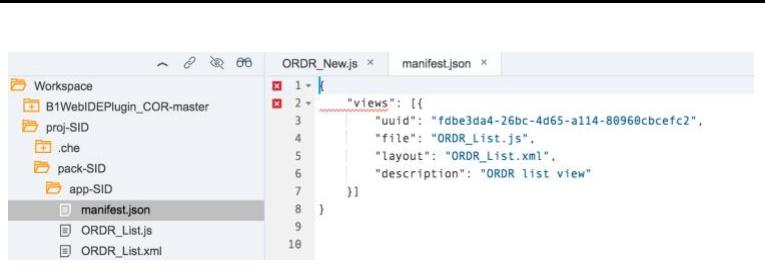


```
449     "ctypes": false,  
450     "xsruntime": false,  
451     "jasmine": false,  
452     "describe": false,  
453     "it": false,  
454     "xdescribe": false,  
455     "xit": false,  
456     "beforeEach": false,  
457     "afterEach": false,  
458     "spyOn": false,  
459     "waitsFor": false,  
460     "runs": false,  
461     "Q": false,  
462     "d3": false,  
463     "sinon": false,  
464     "sap": false  
465   },  
466   "parserOptions": {  
467     "ecmaVersion": 2017  
468   },  
469   "env": {  
470     "browser": true,  
471     "node": true,  
472     "amd": true,  
473     "mocha": true,  
474     "jquery": true,  
475     "es6": true  
476   }  
477 }
```



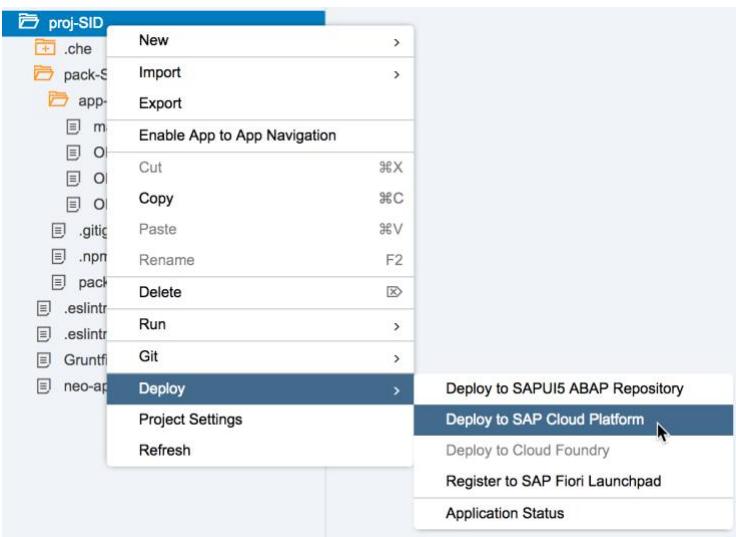
Now let's implement an extension which will automatically fill the delivery date, if it has not yet been filled, before saving the new sales order. To accomplish that we need to handle the button event which creates the sales order and run the custom code.

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

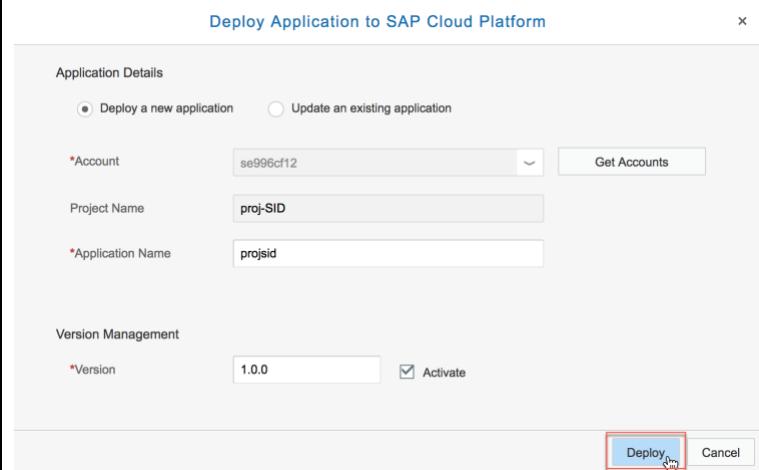
Explanation	Screenshot
Right click the folder <i>proj-<snumber>/pack-<snumber>/app-<snumber></i> Select "New" > "File"	
Name it " ORDR_New.js " and hit " OK ". This file will handle the Sales Orders Detail View	
Time to write the JavaScript with your custom logic. Open the " ORDR_New.js " file that you've recently created. Copy and paste the code from the link below: https://goo.gl/iQD82Q Save your changes.	 <pre> File Edit Build Run Deploy Search View Tools Help ORDR_New.js x 1+ define(function() { 2 // Views 3 const VIEW_ORDR_DETAIL = "9b1eb386-3b4e-4570-98e4-4044088cd415"; 4 5 // Buttons 6 const UUID_BTN_ADD = "bff81f98-5c5e-49b6-8ce4-7b90d875f984"; 7 8 // Fields 9 const HEADER_DocDueDate = "8dd3f85-c8b8-41de-86be-05623195eb28"; 10 11 // Event names 12 const onBeforeButtonClickBtnAdd = "on\$({UUID_BTN_ADD})BeforeButtonClick"; 13 14 // here we define a function to calculate current date + 15 async function SetDate(oInst, detailView = oInst.ActiveView(VIEW_ORDR_DETAIL)) { 16 let docDueDate = detailView.TextInput(HEADER_DocDueDate); 17 let docDueDateVal = await docDueDate.getValue(); 18 19 if (docDueDateVal === null) { 20 var date = new Date(); 21 date.setDate(date.getDate() + 5); 22 var month = date.getMonth() + 1; 23 var day = date.getDate(); 24 month = (Array(2).join('0') + month).slice(-2); 25 day = (Array(2).join('0') + day).slice(-2); 26 var dateStr = date.getFullYear() + '-' + month + '-' + day; 27 let deliveryDate = detailView.DatePicker(HEADER_DocDueDate); 28 await deliveryDate.setValue(dateStr); 29 } 30 return; 31 } // automatically fill the delivery date before creating the sales order 32 [onBeforeButtonClickBtnAdd]: async function(oInst) { 33 await SetDate(oInst); 34 } 35 36 }; 37 }); </pre>
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.	 <pre> 1 [2 { 3 "views": [4 { 5 "uuid": "fdbe3da4-26bc-4d65-a114-80960cbcefcc", 6 "file": "ORDR_List.js", 7 "layout": "ORDR_List.xml", 8 "description": "ORDR list view" 9 } 10] 11 } 12] </pre>

Explanation	Screenshot
<p>Access below link to copy and paste the code to the manifest.json file “views” array:</p> <p>https://goo.gl/7dMHB3</p> <p>Let's understand each field:</p> <ul style="list-style-type: none"> “uuid”: this is the universally unique identifier of the order detail view (<i>you saw how to get this data during the prerequisites configuration steps</i>) “file”: this is the JavaScript file where you will code your customization “description”: this is a description for the new object <p>Save the file.</p>	

Your first extension is ready to be deployed and tested.
 Right click the **project main folder** (myproj-app2018) > **Deploy** > **Deploy to SAP Cloud Platform**

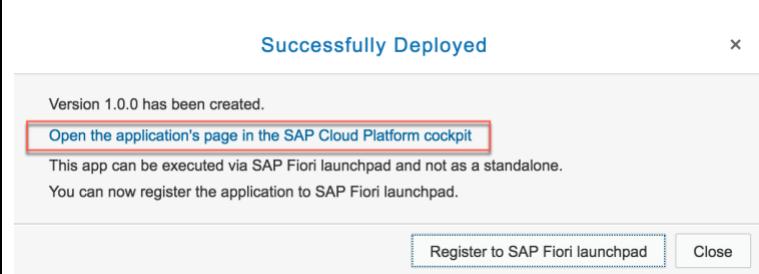


Hit the Deploy button

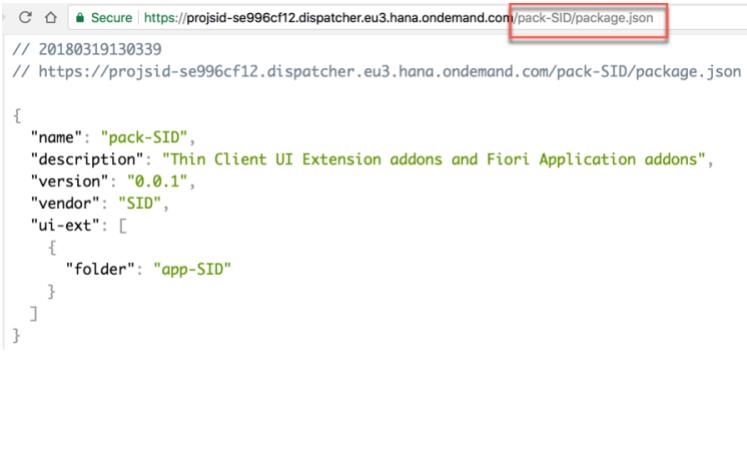
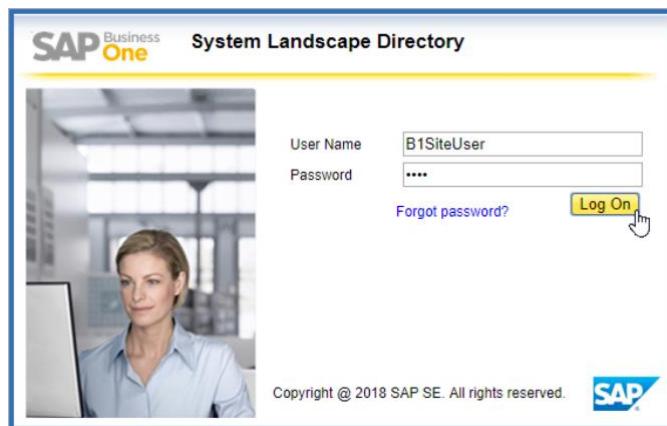
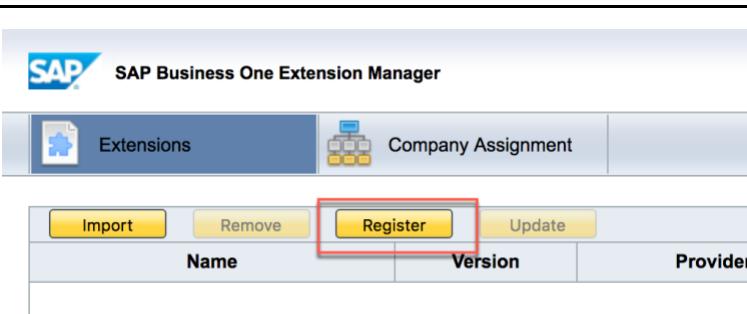


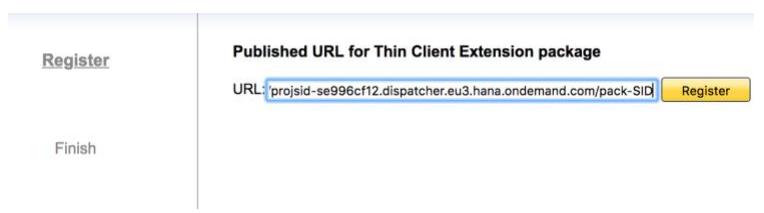
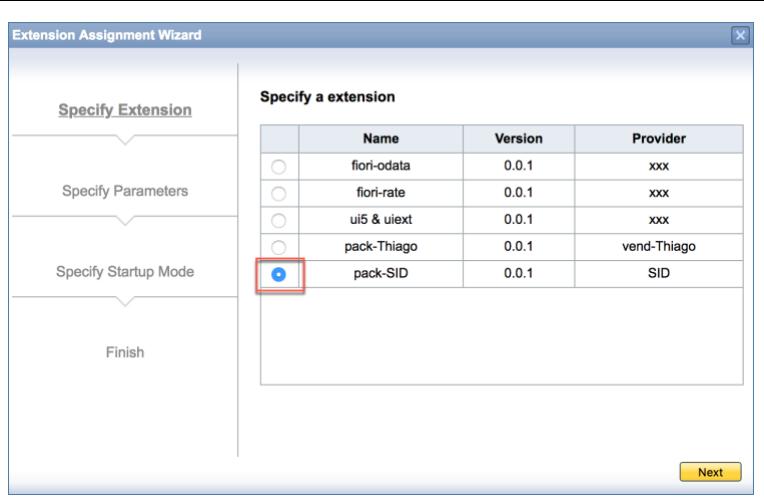
You will see the message box saying it's successfully deployed, don't close it.

Instead, click the link "**Open the application's page in the SAP Cloud Platform cockpit**".



If the "News and Announcements" pops up on your screen, just hit close.

<p>The cockpit will open with the runtime details of your application deployed on SAP Cloud Platform.</p> <p>Copy the application URL.</p>	 <p>The screenshot shows the SAP Cloud Platform cockpit interface. In the top right corner, there is a small decorative bar with three colored squares: black, brown, and yellow. Below this, the cockpit displays the application's status as "Started" with a green checkmark. There are also icons for actions like "Edit" and "Delete". Under the "Active Version" section, it shows the "Active Application Version: 1.0.0" and the "Application URL: https://projsid-se996cf12.dispatcher.eu3.hana.ondemand.com". The URL is highlighted with a red box.</p>
<p>In order to confirm the application is up and running, open a new tab on Chrome and paste the URL.</p> <p>Append the path to the package.json file of your project (/pack-SID/package.json) into the URL and hit "enter".</p> <p>Copy the full URL from the browser address bar.</p> <p>You will need it to register the app into SAP Business One Web Client.</p>	 <p>The screenshot shows a browser window displaying the contents of a package.json file. The URL in the address bar is https://projsid-se996cf12.dispatcher.eu3.hana.ondemand.com/pack-SID/package.json. The JSON content includes fields like name, description, version, vendor, and ui-ext.</p> <pre> { "name": "pack-SID", "description": "Thin Client UI Extension addons and Fiori Application addons", "version": "0.0.1", "vendor": "SID", "ui-ext": [{ "folder": "app-SID" }] } </pre>
<p>Open the Extension Manager:</p> <p>http://goo.gl/9UtVW6</p> <p>Use the following credentials:</p> <p>User Name: B1SiteUser Password: 1234</p>	 <p>The screenshot shows the SAP Business One System Landscape Directory login page. It features a SAP logo and a woman's profile picture. The login form has fields for "User Name" (B1SiteUser) and "Password" (1234). There are links for "Forgot password?" and "Log On". The SAP logo is also present at the bottom right.</p>
<p>You will see this form.</p> <p>Click "Register"</p>	 <p>The screenshot shows the SAP Business One Extension Manager interface. At the top, it says "SAP Business One Extension Manager". Below that is a navigation bar with "Extensions" and "Company Assignment". A toolbar below the navigation bar has buttons for "Import", "Remove", "Register" (which is highlighted with a red box), and "Update". The main area is a table with columns "Name", "Version", and "Provider".</p>

<p>Paste the URL you copied before and remove the following piece: “/package.json”</p> <p>click “Register” click “Next” click “Finish”</p>	
<p>Select “Company Assignment”</p>	
<p>Choose your tenant</p> <p>Click “Assign”</p>	
<p>Select your extension.</p> <p>Keep all the default options for the following wizard screens, until it finishes.</p>	

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

The screenshot shows the SAP Business One Extension Manager interface. In the top navigation bar, there are tabs for 'Extensions' and 'Company Assignment'. The 'Company Assignment' tab is selected. On the left, under 'Company List', the database name 'SBOTCN05' is listed. On the right, the 'Database Name' is shown as 'SBOTCN05'. Below this, under 'Extensions', there is a table with columns 'Name' and 'Version'. A single row is present with 'Name' as 'pack-SID' and 'Version' as '0.0.1'. There are 'Assign' and 'UnAssign' buttons above the table.

Browse to the B1 Web Client.

<https://goo.gl/GfWDrD>

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

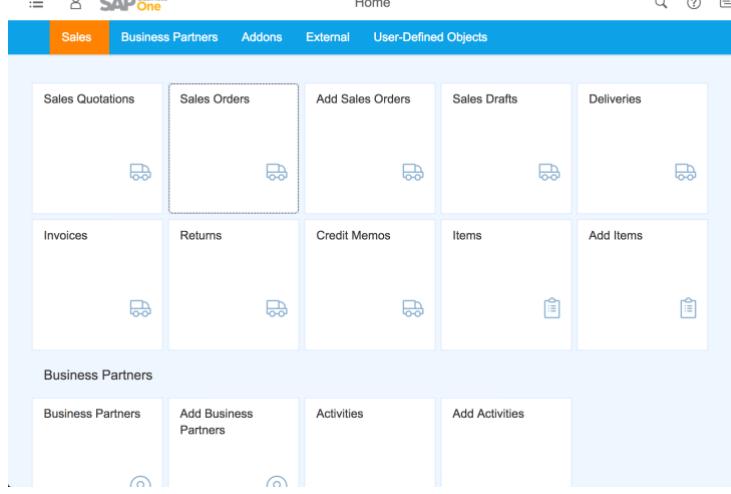
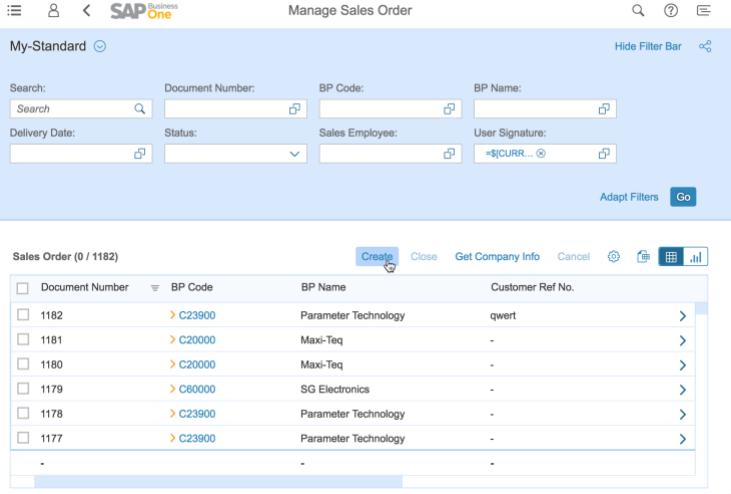
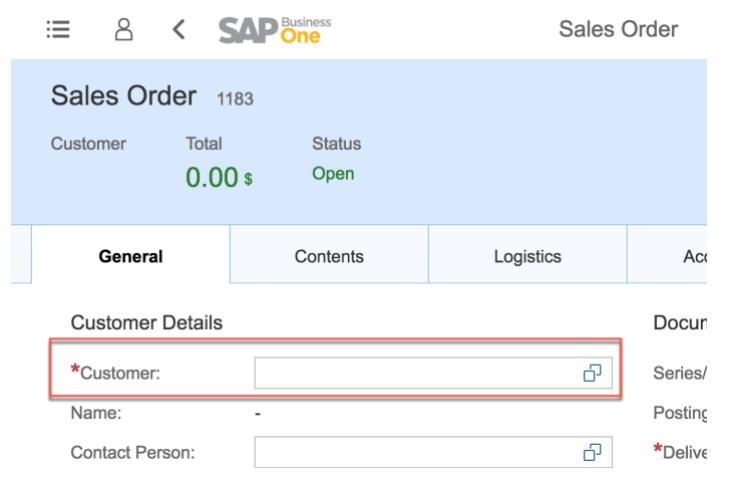
Select your tenant (**NOTE: IT SHOULD BE THE SAME OF THE ONE YOU HAVE ASSIGNED YOUR APP TO, IN THE PREVIOUS STEPS.**)

Login using below credentials:

Username: manager
Password: Initial0

The screenshot shows the SAP Business One Web Client interface. At the top, it displays 'SAP Business One' with a user icon and the name 'Jayson Butler'. Below this, there are two main sections: 'Sales Quotations' and 'Sales Orders'. Further down, it shows a large user profile picture of 'Jayson Butler' and a 'Sign Out' button.

The screenshot shows the SAP Business One login screen. The title bar says 'OEC Computers (SBOTCN05)'. It has fields for 'Username' containing 'manager' and 'Password' containing '.....'. A blue 'Login' button is at the bottom.

<p>Click the “Sales Orders” tile</p>	
<p>Click “Create”</p>	
<p>Select the customer from the choose from list (suggested: Maxi-Teq or Parameter Technology)</p>	

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

Click “Select”

SAP Business One Sales Order

Sales Order 1184

Customer Total Status
0.00 \$ Open

General Contents Logistics Accounting

Customer Details Document

*Customer:

Sales Order 1184

Customer Total Status
0.00 \$ Open

General Contents Logistics Accounting User-Defined F...

Currency: Local Currency Item / Service Type: Item

Product Gross Profit Volume & We... Copy Paste Duplicate

#	Item No.	Item Description	Quantity
1	<input type="text"/>		-

Total Before Disco... Add & View Save as Draft & New Cancel

Browse back to the “General” tab

SAP Business One Sales Order

Sales Order 1184

Customer Total Status
0.00 \$ Open

General Contents Logistics Accounting

Customer Details Document Detail

*Customer: Series/No.:
Name: - Posting Date:
Contact Person: *Delivery Date:
*Customer Ref No.: Document Date:
Remarks: Sales Employee:

Check that the Delivery Date is empty and click the “Add” button to create the Sales Order

Sales Order 1184

Customer Total Status
C20000 636.00 \$ Open

General Contents Logistics Accounting User-Defined Fie...

Customer Details Document Details

*Customer: C20000
Name: Maxi-Teq
Contact Person: Max Teq
Customer Ref No.:
Remarks:
Posting Date: Mar 19, 2018
Series/No.: Primary 1184
Delivery Date: MMM d, y
Document Date: Mar 19, 2018
Sales Employee: Sales Manager
Owner:

Add & View Save as Draft & New Cancel

Message
Operation completed successfully
OK

Validate that the Sales Order has been successfully created and that the Delivery Date is the current date + 5 days.

Document Details

Series/No.: Primary 1184
Posting Date: 2018/03/19
Delivery Date: 2018/03/24
Document Date: 2018/03/19

Congratulations, you have finished the first exercise!



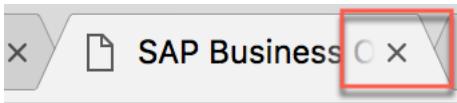
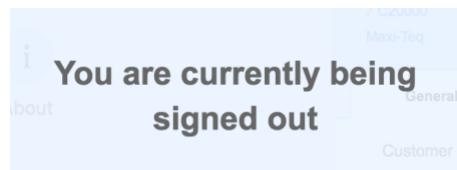
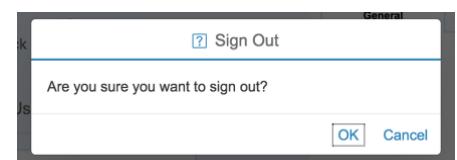
Before continuing, please log out from SAP Business One Web Client, and close the window:

Click the user icon

Select “Sign Out”

Confirm the message

Close the browser





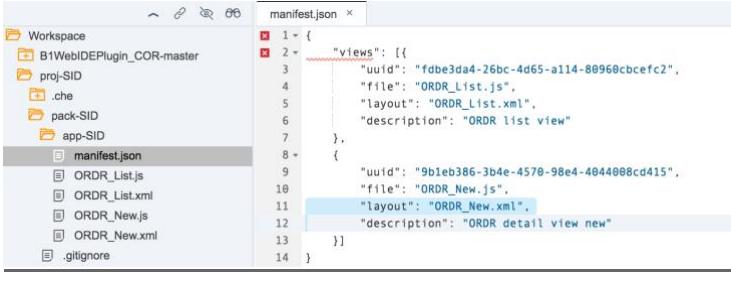
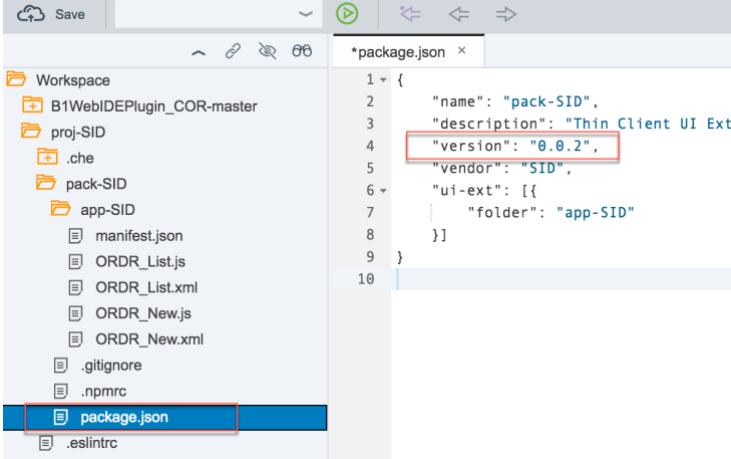
EXERCISE 2: MAKE FIELD “CUSTOMER REFERENCE NUMBER” MANDATORY

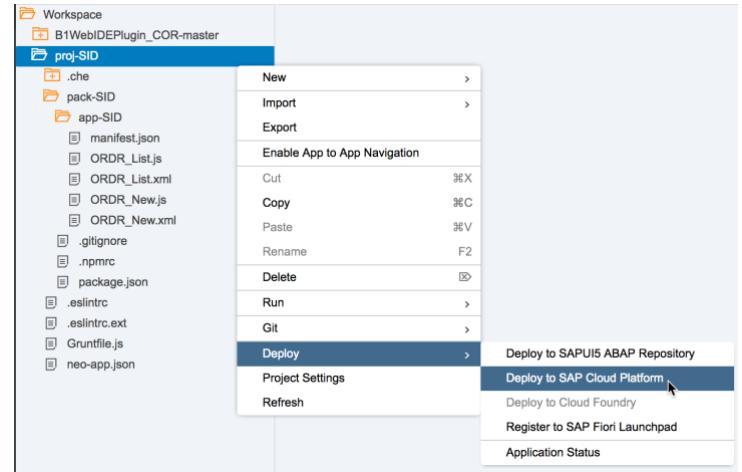
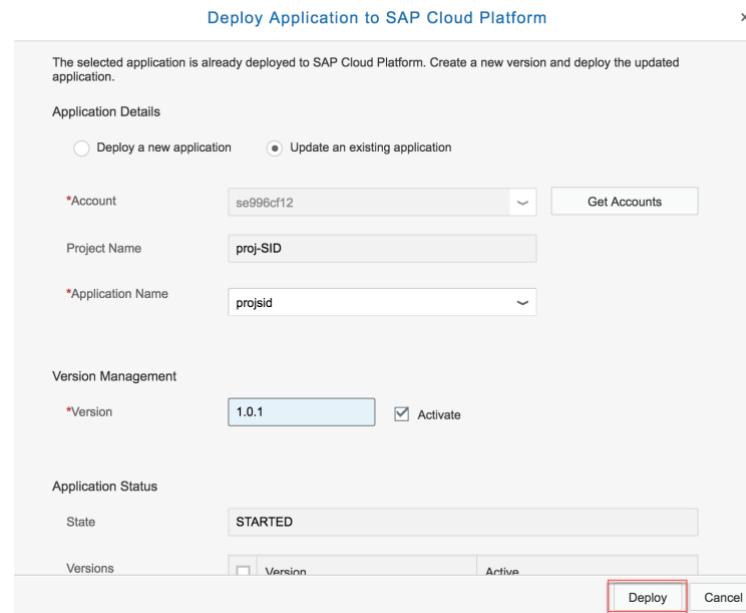
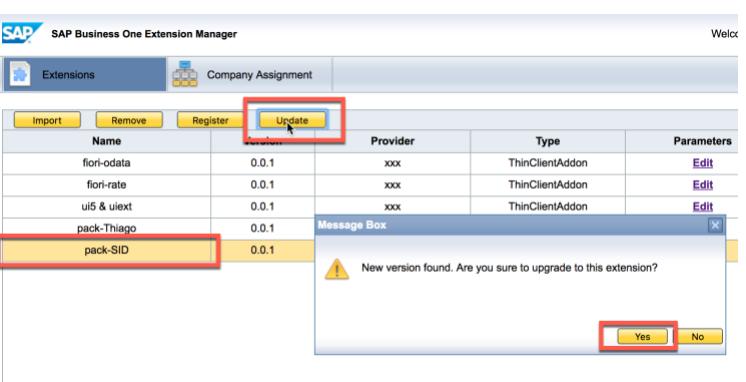
For this second exercise, we want to make an existing field mandatory.

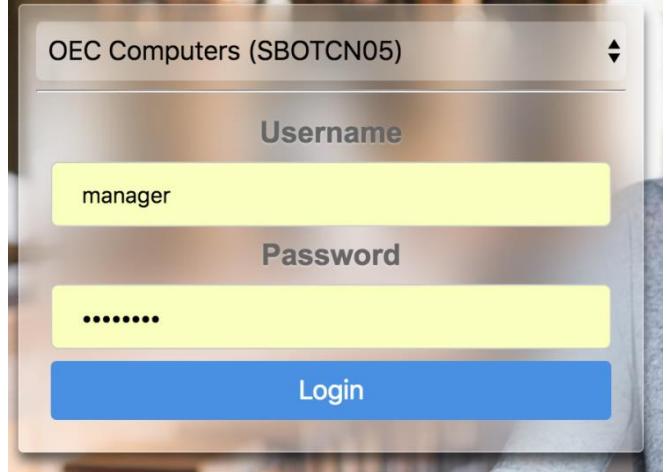
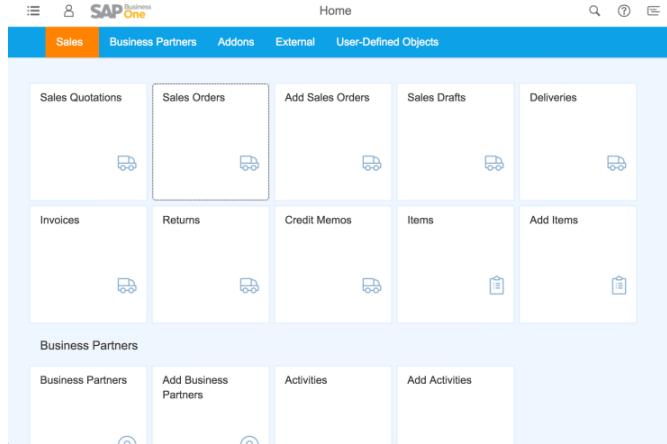
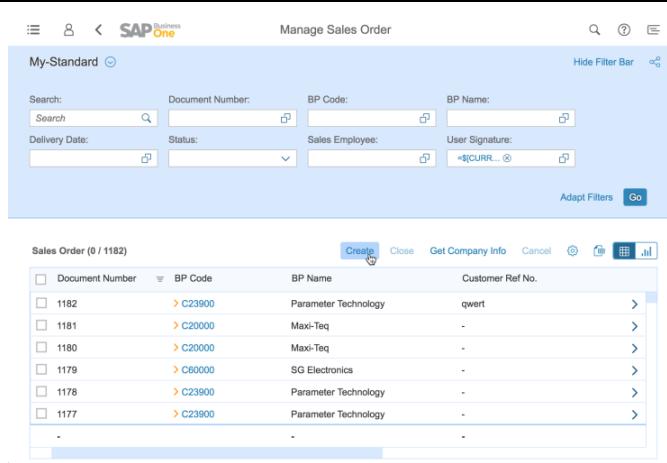
To accomplish changes in the UI components, we need an XML file with the corresponding components and properties.

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.

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

Explanation	Screenshot
<p>Add the XML to make the field "Customer Reference Number" mandatory</p> <p>https://goo.gl/597DbH Save it.</p>	 <pre data-bbox="720 332 1399 444"> 1 <overlay> 2 <on uuid="31ffb45d-e42c-4eea-94dc-21d1dbaaa7c7" mandatory="true"/> 3 </overlay> 4 </pre>
<p>Now open the manifest.json file and add a reference to the new xml file in the existing object.</p> <p>https://goo.gl/tVFCQ2 Save it.</p>	 <pre data-bbox="720 579 1378 819"> 1 <! 2 "views": [3 { 4 "uuid": "fdbbe3da4-26bc-4d65-a114-80960cbcef2", 5 "file": "ORDR_List.js", 6 "layout": "ORDR_List.xml", 7 "description": "ORDR list view" 8 }, 9 { 10 "uuid": "9bleb386-3b4e-4570-98e4-4044008cd415", 11 "file": "ORDR_New.js", 12 "layout": "ORDR_New.xml", 13 "description": "ORDR detail view new" 14 }] </pre>
<p>Open the package.json file and update the version of your app, in this case it will be 0.0.2</p> <p>Save it.</p>	 <pre data-bbox="720 945 1430 1185"> 1 <! 2 "name": "pack-SID", 3 "description": "Thin Client UI Extensibility Sample Application", 4 "version": "0.0.2", 5 "vendor": "SID", 6 "ui-ext": [7 { 8 "folder": "app-SID" 9 } 10] </pre>

Explanation	Screenshot
<p>Your changes are ready to be deployed and tested.</p>	 <p>The screenshot shows a context menu for a project folder named 'proj-SID'. The 'Deploy' option is selected, revealing a submenu with options: 'Deploy to SAPUI5 ABAP Repository', 'Deploy to SAP Cloud Platform' (which is highlighted), 'Deploy to Cloud Foundry', 'Register to SAP Fiori Launchpad', and 'Application Status'.</p>
<p>Right click the project main folder > Deploy > Deploy to SAP Cloud Platform</p>	
<p>It will update the existing application.</p>	
<p>Click the "Deploy" button.</p>	 <p>The dialog box is titled 'Deploy Application to SAP Cloud Platform'. It contains the following fields:</p> <ul style="list-style-type: none"> Application Details: <ul style="list-style-type: none"> Radio button for 'Update an existing application' is selected. *Account: se996cf12 Project Name: proj-SID *Application Name: projsid Version Management: <ul style="list-style-type: none"> *Version: 1.0.1 <input checked="" type="checkbox"/> Activate Application Status: <ul style="list-style-type: none"> State: STARTED Versions table: Shows a single row for 'proj-SID' with Version 0.0.1 and Active status. A 'Deploy' button is highlighted with a red box.
<p>Now, before testing the changes, you need to tell SAP Business One, that your app is updated.</p>	 <p>The screenshot shows the SAP Business One Extension Manager interface. An 'Update' button is highlighted with a red box. A 'Message Box' dialog is open, asking if the user wants to upgrade the extension 'pack-SID' from version 0.0.1 to 0.0.1. The 'Yes' button is highlighted with a red box.</p>
<p>Open the Extension Manager, select your app and click "Update".</p>	
<p>Once it's updated, the screen will refresh and the Version must match with the version you've informed in the package.json file.</p>	

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



Explanation	Screenshot
<p>Check that the field “Customer Reference Number” is now marked with the red asterisk, indicating it's now a mandatory field.</p>	<p>The screenshot shows the SAP Business One Sales Order interface. At the top, there are navigation icons and the SAP Business One logo. Below that, the title "Sales Order 1184" is displayed. Underneath the title, there are fields for "Customer" (containing "0.00 \$"), "Total" (containing "0.00 \$"), and "Status" (containing "Open"). A horizontal menu bar below these fields includes "General", "Contents", "Logistics", and "Accounting". The main area is titled "Customer Details" and contains several input fields: "Customer:" (with a red asterisk), "Name:", "Contact Person:", and "Customer Ref No.". The "Customer Ref No." field is highlighted with a red border. To the right of these fields are "Document Detail" fields: "Series/No.:", "Posting Date:", "Delivery Date:", and "Document Date:". Below the customer details, there are fields for "Remarks:" and "Sales Employee:". The entire screenshot is framed by a thick black border.</p>
<p>Try to create a Sales Order without the Customer Ref No. and you will receive an error.</p>	<p>The screenshot shows a modal error dialog box. The title bar says "Error" with a red exclamation mark icon. The main message area contains the text "Missing value: Customer Ref No.". In the bottom right corner of the dialog box, there is a blue "OK" button.</p>

Congratulations!



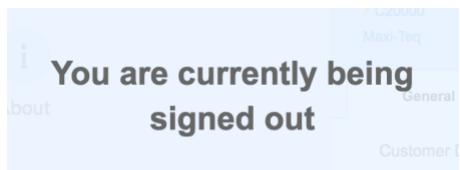
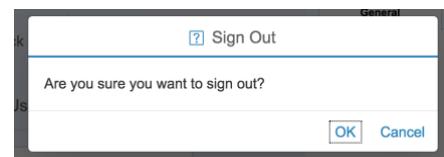
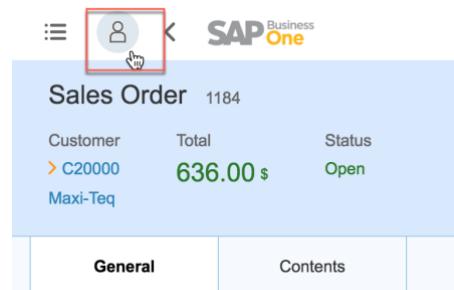
Before continuing, please log out from SAP Business One Web Client, and close the window:

Click the user icon

Select “Sign Out”

Confirm the message

Close the browser

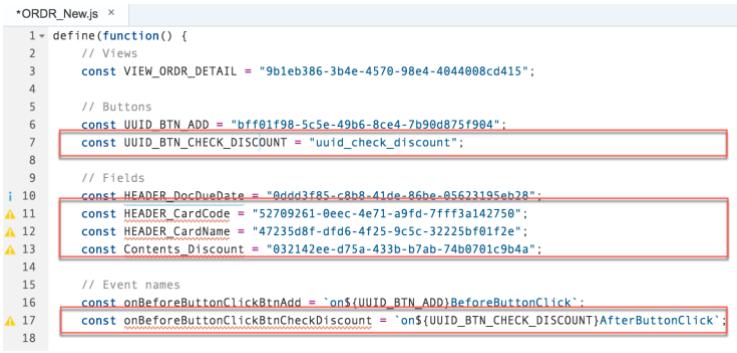
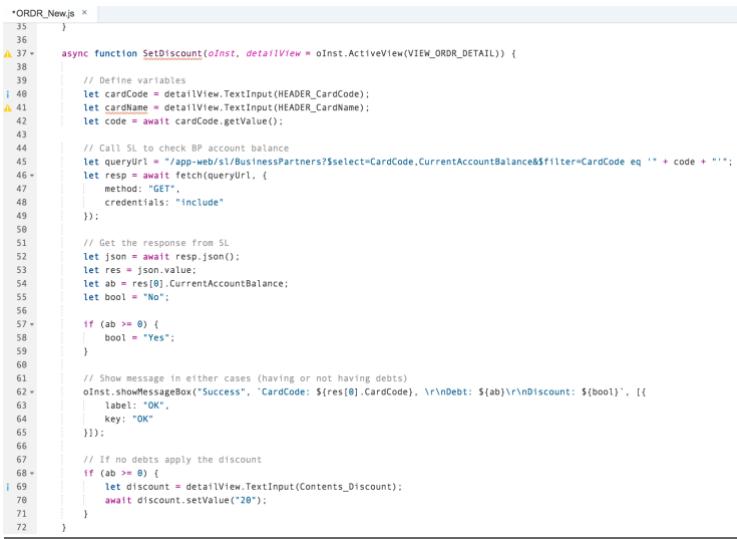


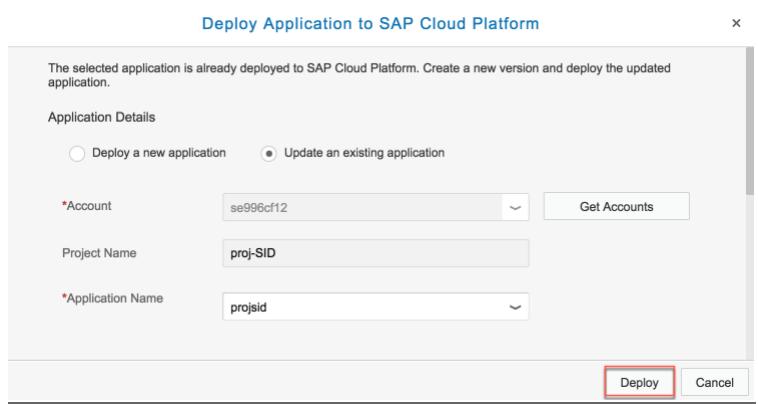
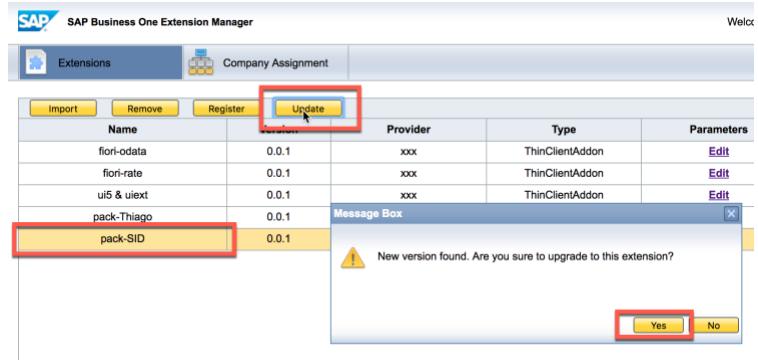
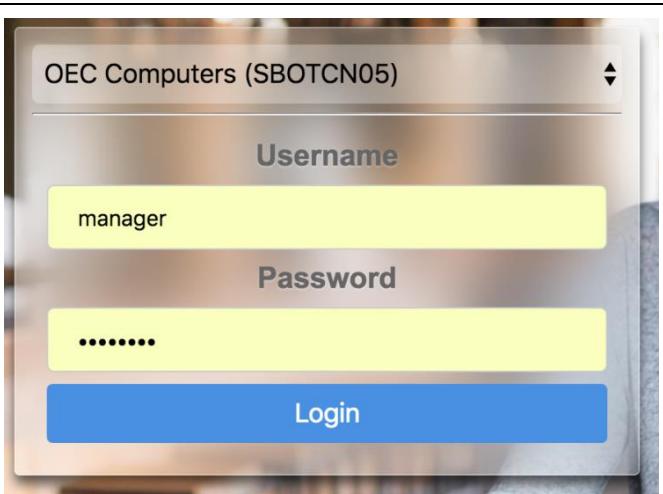


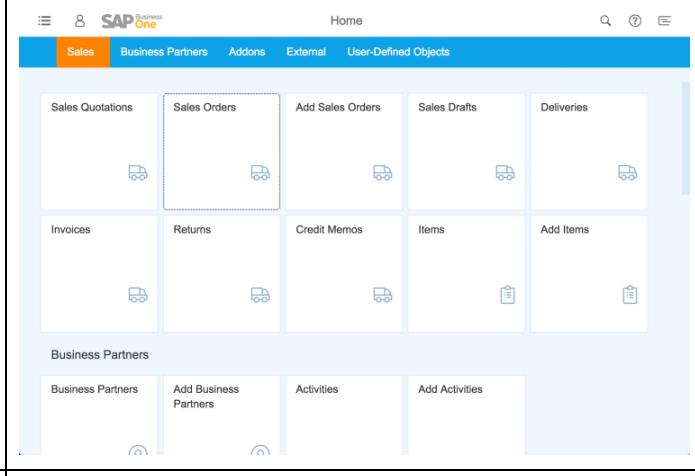
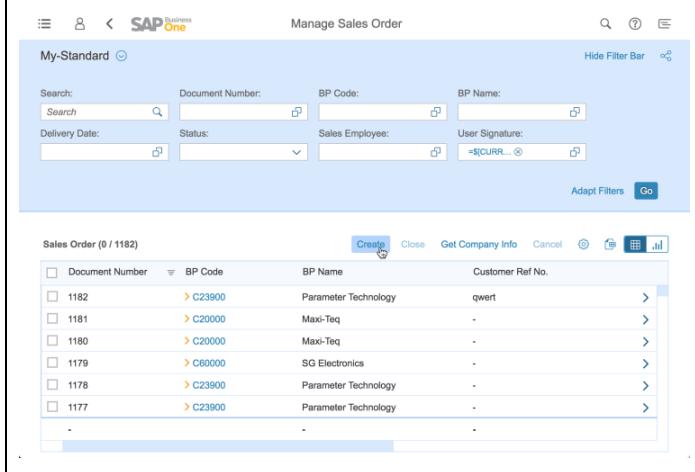
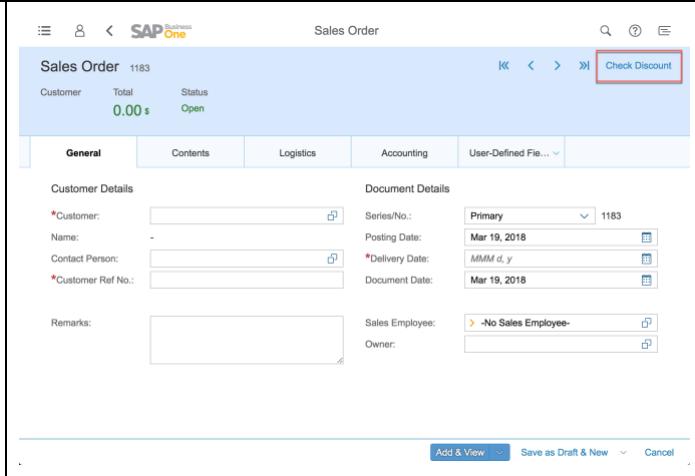
EXERCISE 3: ADD A BUTTON TO THE SALES ORDER TO CHECK BUSINESS PARTNER'S DEBTS

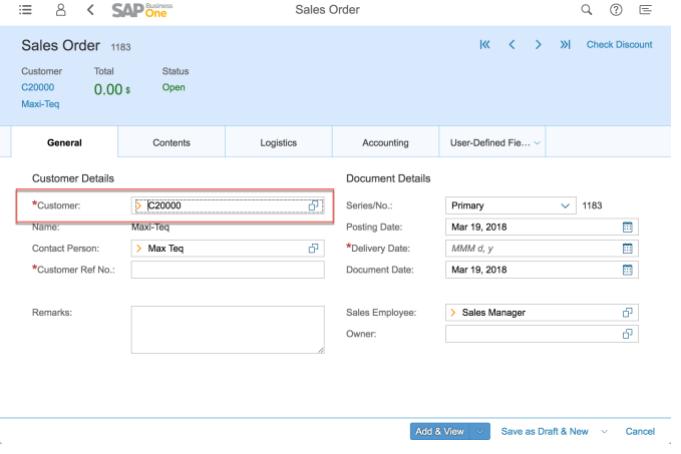
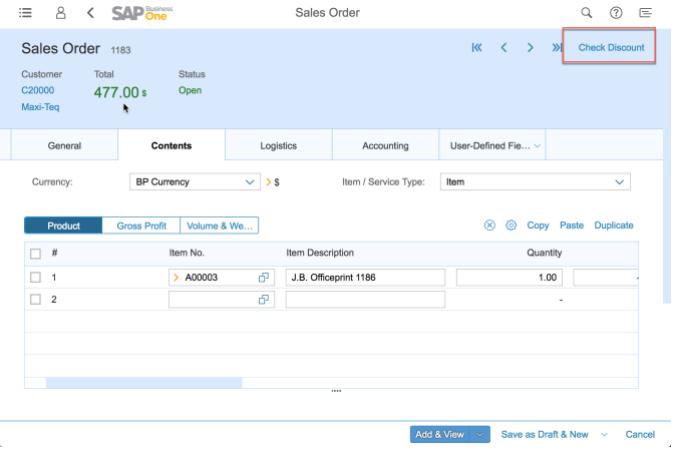
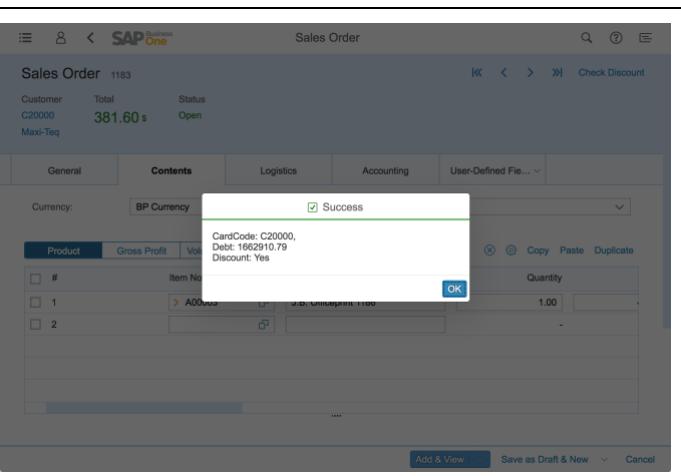
As a last exercise, you will create a new button to the existing Sales Order UI. You will handle the button click to check if the customer has pending debts. If there's no pending debts, alert the user and apply 20% discount to the current Sales Order.

Explanation	Screenshot
Open Google Chrome.	
Access the SCP Web IDE using your S-number credentials. https://goo.gl/PZqa1Y	
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.	 The UI Inspector XML code for the button is: <button method_args="onClickOpenNew" method="Bo.Action"> <button method_args="onClickFirst" method="Bo.Action"> <button method_args="onClickPrev" method="Bo.Action"> <button method_args="onClickNext" method="Bo.Action"> <button method_args="onClickLast" method="Bo.Action"> <button method_args="onClickLast" method="Bo.Action"> </button>
Open the ORDR_New.xml and add the new button to the form. https://goo.gl/h9MBC3	 The XML code in the ORDR_New.xml file is: 1 <overlay> 2 <on uuid="31ffb45d-e42c-4eea-94dc-21d1dbaaa7c7" mandatory="true"/> 3 <after uuid="9366573f-9be8-484a-a37d-9a8b36f6ee1a"> 4 <button text="Check Discount" uuid="uuid_check_discount"></button> 5 </after> 6 </overlay>

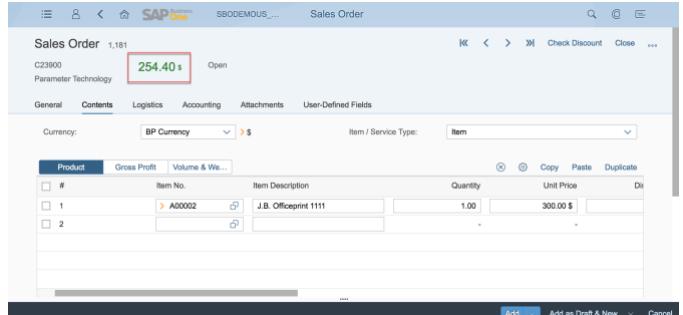
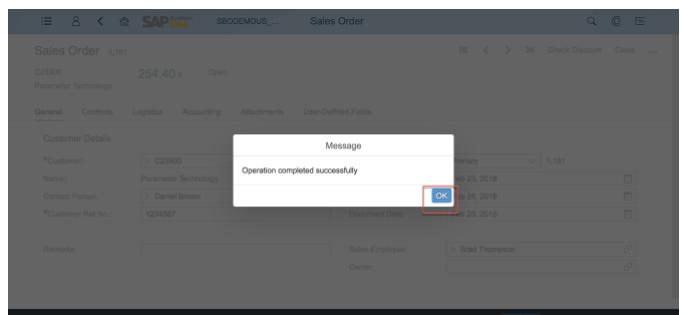
Explanation	Screenshot						
<p>Open the ORDR_New.js file and declare a constant for the new button.</p> <p>Declare a constant for the event you need to handle.</p> <p>Declare a constant for the discount, CardCode and CardName fields.</p> <p>You can copy and paste the new code lines from here:</p> <p>https://goo.gl/Axokch</p>	 <pre>*ORDR_New.js × 1 define(function() { 2 // Views 3 const VIEW_ORDR_DETAIL = "9b1eb386-3b4e-4570-98e4-4044008cd415"; 4 5 // Buttons 6 const UUID_BTN_ADD = "bff01f98-5c5e-49b6-8ce4-7b90d875f904"; 7 const UUID_BTN_CHECK_DISCOUNT = "uuid_check_discount"; 8 9 // Fields 10 const HEADER_DocDueDate = "0ddd3fb5-c8b8-41de-86be-05623195eb28"; 11 const HEADER_CardCode = "52709261-9ec-4e71-a9fd-7ff3a142750"; 12 const HEADER_CardName = "47235d8f-dfd6-4f25-9c5c-32225bf01f2e"; 13 const Contents_Discount = "032142ee-d75a-433b-b7ab-74b0701c9b4a"; 14 15 // Event names 16 const onBeforeButtonClickBtnAdd = `on\${UUID_BTN_ADD}BeforeButtonClick`; 17 const onBeforeButtonClickBtnCheckDiscount = `on\${UUID_BTN_CHECK_DISCOUNT}AfterButtonClick`;</pre>						
<p>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)</p> <p>https://goo.gl/x8BRhc</p>	 <pre>*ORDR_New.js × 35 } 36 37 + async function SetDiscount(oInst, detailView = oInst.ActiveView(VIEW_ORDR_DETAIL)) { 38 39 // Define variables 40 let cardCode = detailView.TextInput(HEADER_CardCode); 41 let cardName = detailView.TextInput(HEADER_CardName); 42 let code = await cardCode.getValue(); 43 44 // Call SL to check BP account balance 45 let queryUrl = `/app/websl/BusinessPartners?\$select=CardCode,CurrentAccountBalance&\$filter=CardCode eq '\${code}'`; 46 let resp = await fetch(queryUrl, { 47 method: "GET", 48 credentials: "include" 49 }); 50 51 // Get the response from SL 52 let json = await resp.json(); 53 let res = json.value; 54 let ab = res[0].CurrentAccountBalance; 55 let bool = "No"; 56 57 if (ab >= 0) { 58 bool = "Yes"; 59 } 60 61 // Show message in either cases (having or not having debts) 62 oInst.showMessageBox("Success", `CardCode: \${res[0].CardCode}, \r\nDebt: \${ab}\r\nDiscount: \${bool}`.trim(), 63 { 64 label: "OK", 65 key: "OK" 66 }); 67 68 // If no debts apply the discount 69 if (ab >= 0) { 70 let discount = detailView.TextInput(Contents_Discount); 71 await discount.setValue("20"); 72 } 73 }</pre>						
<p>Now append the event handler of the new button</p> <p>https://goo.gl/RxPNMc</p> <p>Save it!</p>	 <pre>74 + 75 return{ 76 // automatically fill the delivery date before creating the sales order 77 [onBeforeButtonClickBtnAdd]: async function(oInst) { 78 await SetDate(oInst); 79 }, 80 81 [onBeforeButtonClickBtnCheckDiscount]: async function(oInst) { 82 await SetDiscount(oInst); 83 } 84 }; 85 }</pre>						
<p>Open the package.json file and update the version of your app, in this case it will be 0.0.3</p> <p>Save it.</p>	 <table border="1" data-bbox="695 1578 1432 1949"> <tr> <td data-bbox="695 1578 901 1612">package.json</td> <td data-bbox="901 1578 1108 1612">ORDR_New.js</td> <td data-bbox="1108 1578 1432 1612">ORDR_New.</td> </tr> <tr> <td data-bbox="695 1612 901 1949"> <pre>1 { 2 "name": "pack-SID", 3 "description": "Thin Client UI Ext", 4 "version": "0.0.3", highlighted 5 "vendor": "SID", 6 "ui-ext": [7 { 8 "folder": "app-SID" 9 } 10]</pre> </td> <td data-bbox="901 1612 1108 1949"></td> <td data-bbox="1108 1612 1432 1949"></td> </tr> </table>	package.json	ORDR_New.js	ORDR_New.	<pre>1 { 2 "name": "pack-SID", 3 "description": "Thin Client UI Ext", 4 "version": "0.0.3", highlighted 5 "vendor": "SID", 6 "ui-ext": [7 { 8 "folder": "app-SID" 9 } 10]</pre>		
package.json	ORDR_New.js	ORDR_New.					
<pre>1 { 2 "name": "pack-SID", 3 "description": "Thin Client UI Ext", 4 "version": "0.0.3", highlighted 5 "vendor": "SID", 6 "ui-ext": [7 { 8 "folder": "app-SID" 9 } 10]</pre>							

Explanation	Screenshot
<p>Your changes are ready to be deployed and tested. Right click the project main folder > Deploy > Deploy to SAP Cloud Platform.</p> <p>Click the “Deploy” button.</p>	
<p>Now, before testing the changes, you need to tell SAP Business One, that your app is updated.</p> <p>Open the Extension Manager, select your app and click “Update”.</p> <p>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.</p>	
<p>Now, test the changes.</p>	
Explanation	Screenshot
<p>Open Google Chrome</p>	
<p>Open SAP Business One Web Client >></p> <p>https://goo.gl/GfWDrD</p> <p>Company: make sure you are connecting to the correct tenant</p> <p>Username: manager Password: Initial0</p> <p>Click Login</p>	

Explanation	Screenshot
<p>Click the Sales Orders tile.</p>	
<p>Click "Create"</p>	
<p>Validate that the new button "Check Discount" is displayed, don't click it.</p>	

Explanation	Screenshot
<p>Select a customer</p>	 <p>The screenshot shows the SAP Business One Sales Order interface. In the 'Customer' field, the value 'C20000 Maxi-Teq' is selected. The total amount is listed as '0.00 \$' and the status is 'Open'. The 'General' tab is selected. The 'Customer Details' section contains fields for Customer, Name, Contact Person, and Customer Ref No., all of which are populated with the values from the previous step. The 'Document Details' section includes fields for Series/No., Posting Date, Delivery Date, and Document Date, all set to their default values. At the bottom right, there are buttons for 'Add & View', 'Save as Draft & New', and 'Cancel'.</p>
<p>Click "Contents" tab</p> <p>Select an Item</p> <p>Look at the Sales Order total amount</p> <p>Click "Check Discount"</p>	 <p>The screenshot shows the SAP Business One Sales Order interface. The 'Contents' tab is selected. The total amount is now listed as '477.00 \$'. The 'Check Discount' button is highlighted with a red box. The 'Item' service type is selected. The item list table shows two items: 'A00003 J.B. Officeprint 1186' with a quantity of '1.00'. At the bottom right, there are buttons for 'Add & View', 'Save as Draft & New', and 'Cancel'.</p>
<p>A message will be displayed showing whether the discount is applicable.</p> <p>Click "OK"</p> <p>Look at the Total amount</p>	 <p>The screenshot shows the SAP Business One Sales Order interface. The total amount is now listed as '381.60 \$'. A modal dialog box titled 'Success' displays the message: 'CardCode: C20000, Debt: 1662910.79, Discount: Yes'. The 'OK' button is visible at the bottom of the dialog. At the bottom right, there are buttons for 'Add & View', 'Save as Draft & New', and 'Cancel'.</p>



Explanation	Screenshot
<p>Note the Sales Order amount, it is reduced by 20%</p>	 <p>The screenshot shows the SAP Business One Sales Order screen. The total amount displayed is 254.40 \$, which has been reduced from its original value. The screen includes tabs for General, Contents, Logistics, Accounting, Attachments, and User-Defined Fields. A product table lists items with their descriptions, quantities, and unit prices.</p>
<p>Click "Add"</p>	
<p>Click "OK"</p>	 <p>The screenshot shows the SAP Business One Sales Order screen after an action has been completed. A message box appears stating "Operation completed successfully". The "OK" button in the message box is highlighted with a red box. The background shows customer details and other sales order fields.</p>

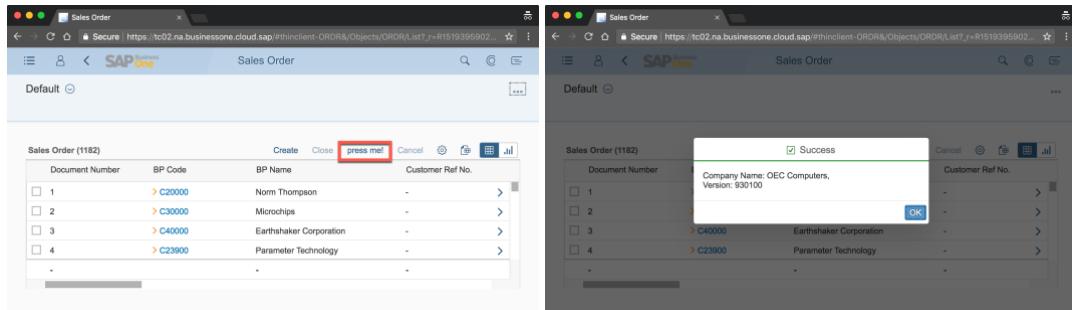
Congratulations! You have finished the exercises on the extensibility of SAP Business One Web Client. Please, give us your feedback.



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.

```
1 define(function() {
2     // List page
3     const BTN_GET_COMPANY_LIST = "uuid_get_company_info";
4
5     // Event names
6     const onAfterButtonClickedGetCompanyInfo = `on${BTN_GET_COMPANY_LIST}AfterButtonClick`;
7
8     return {
9         [onAfterButtonClickedGetCompanyInfo]: async function(oInst) {
10             let resp = await fetch("/app-web/s1/CompanyService_GetCompanyInfo", {
11                 method: "POST",
12                 credentials: "include"
13             });
14             let json = await resp.json();
15             oInst.showMessageBox("Success", `Company Name: ${json.CompanyName}, \r\nVersion: ${json.Version}`, [
16                 {label: "OK", key: "OK"}
17             ]);
18         }
19     };
20 });
21 });
22});
```

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

```
1 {
2     "views": [
3         {
4             "uuid": "fdbe3da4-26bc-4d65-a114-80960cbcefc2",
5             "file": "ORDR_List.js",
6             "layout": "ORDR_List.xml",
7             "description": "ORDR list view"
8         }
9     ]
10 }
```

www.sap.com/contactsap

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

