



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

PARTNER



SAP



TABLE OF CONTENTS

PREREQUISITES.....	4
i. SAP Business One Web Client DevTools.....	4
EXERCISE 1: AUTOMATICALLY FILL THE DELIVERY DATE.....	8
EXERCISE 2: MAKE FIELD “CUSTOMER REFERENCE NUMBER” MANDATORY.....	24
EXERCISE 3: ADD A BUTTON TO THE SALES ORDER TO CHECK BUSINESS PARTNER’S DEBTS..	30



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



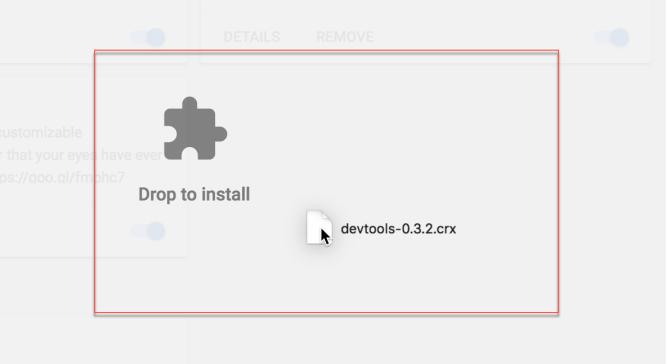
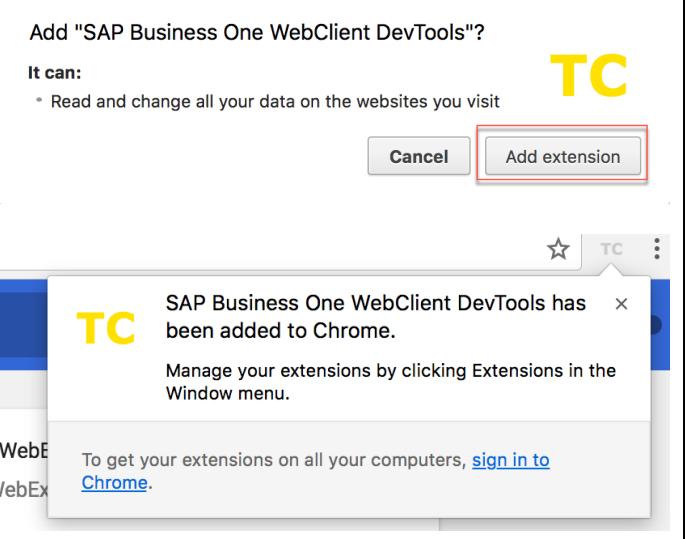
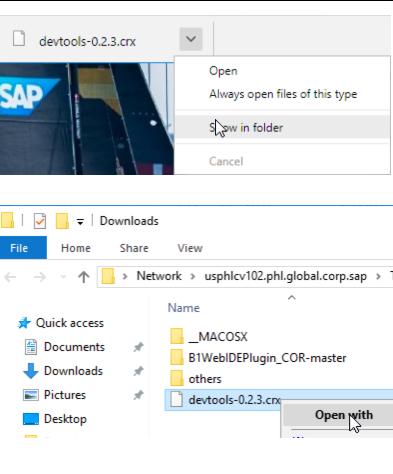
PREREQUISITES

i. SAP Business One Web Client DevTools

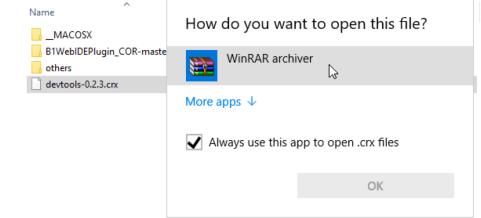
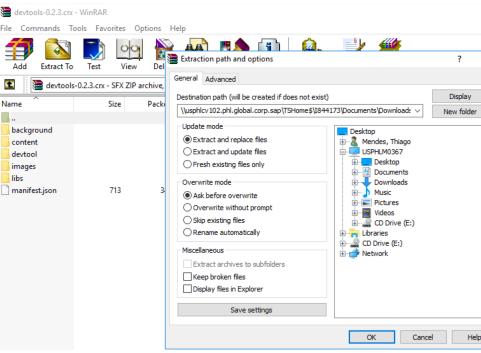
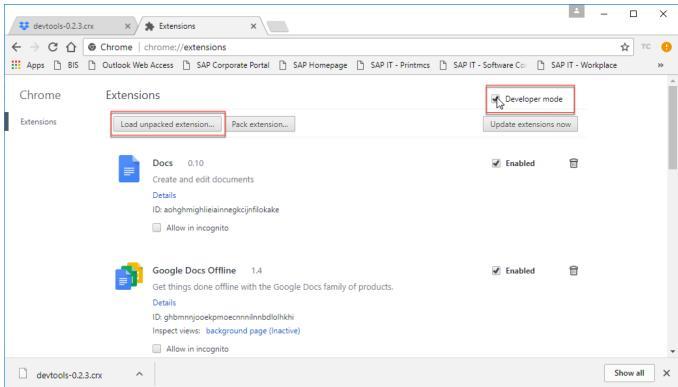
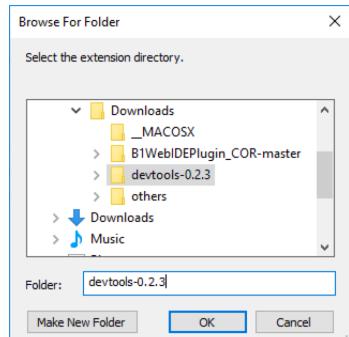
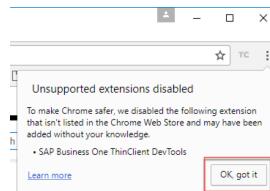
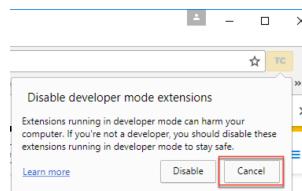
This is the SAP Business One Web Client DevTools, the Chrome Extension which enables you to get detailed technical data of the SAP Business One Web Client UI components.

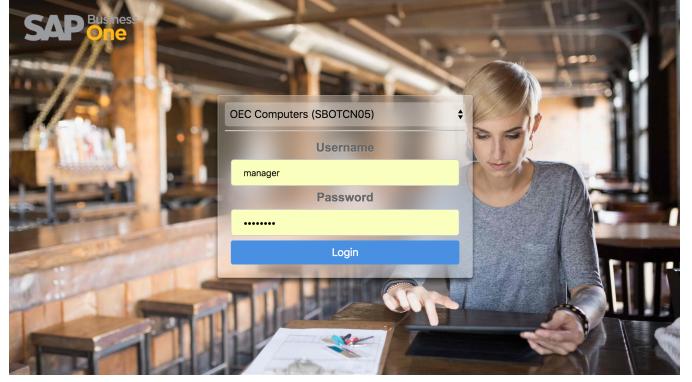
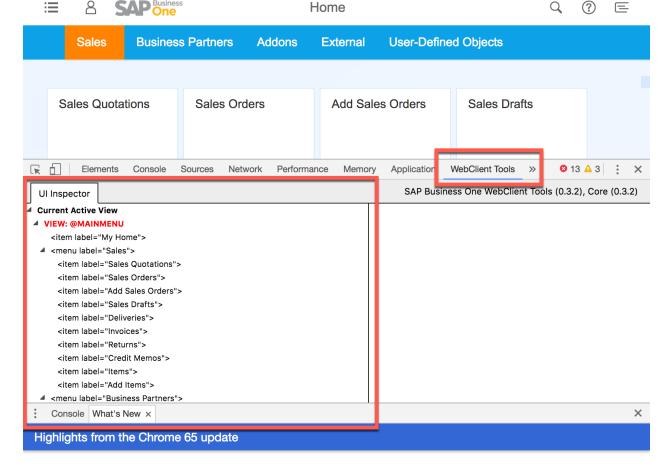
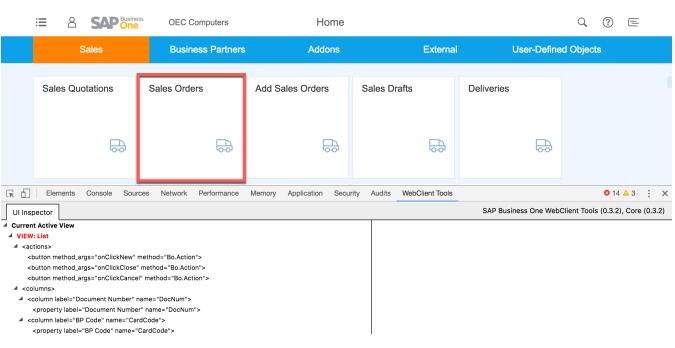
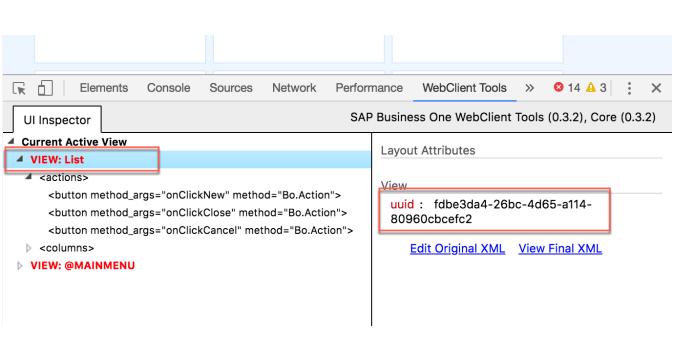
Explanation	Screenshot
Open Google Chrome	
Download and save the latest version of the Chrome extension here: https://goo.gl/qTX1ge	
In your chrome, open URL "chrome://extensions/".	



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, otherwise go straight to page 7</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>Your instructor has assigned an specific URL to you, check with him.</p> <p>Login using below credentials:</p> <p>Company: Check with your instructor Username: manager Password: Check with your instructor 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>	
<p>In the "UI Inspector" tab, select the "VIEW: List". Look at the right-frame. The attributes of the selected component are shown there.</p> <p>In this case, the Universally Unique Identifier for the Sales Order List View is shown here.</p> <p>Select other components on the "UI Inspector" tab and see how they get highlighted in the browser.</p>	

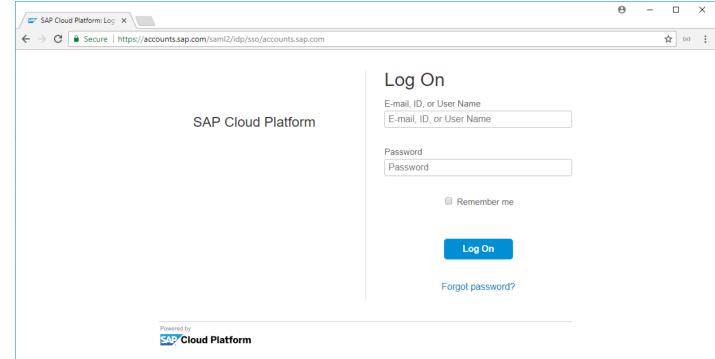
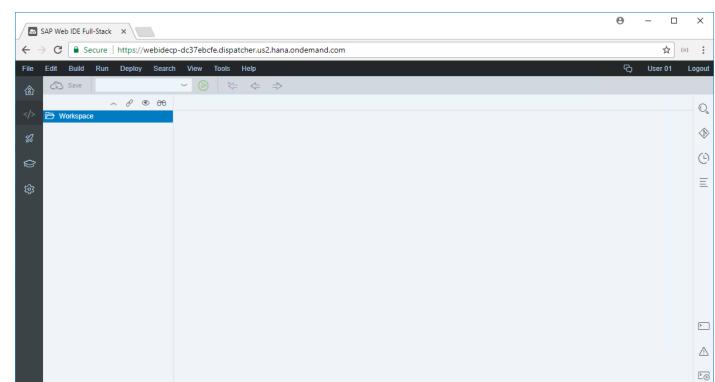
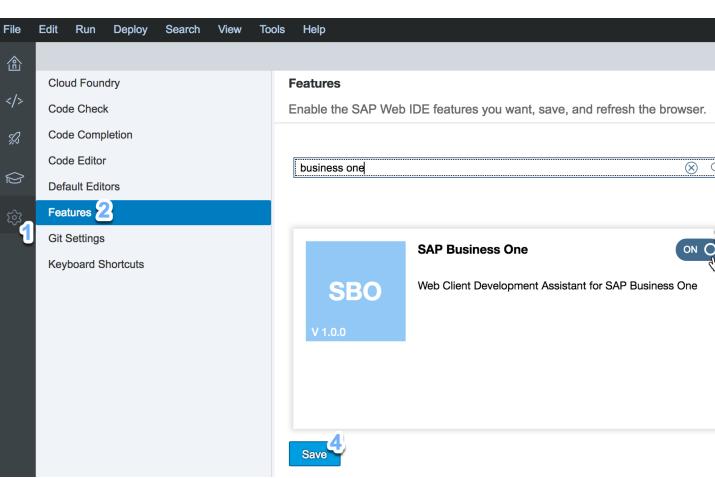


EXERCISE 1: AUTOMATICALLY FILL THE DELIVERY DATE

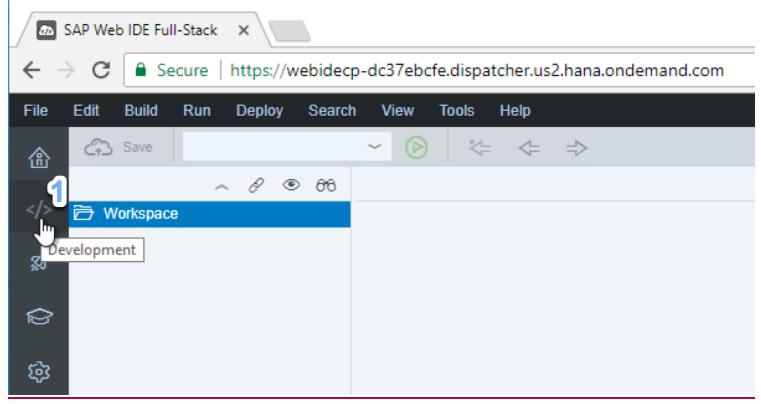
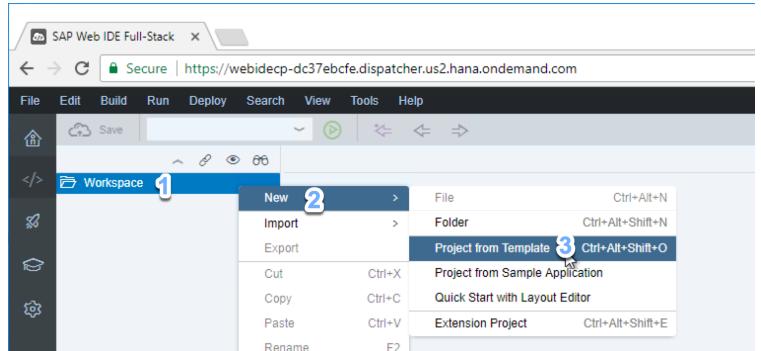
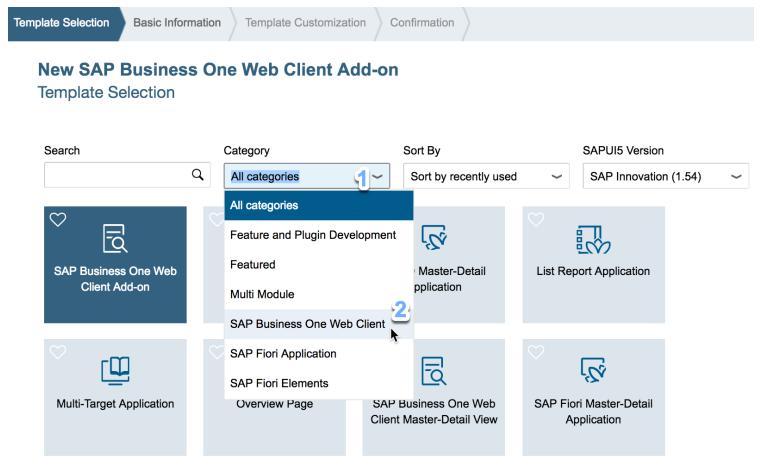
Let's get started, coding your first loosely-coupled application on top of SAP Business One Web Client.

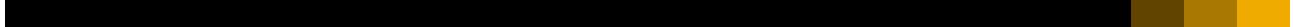
For this first exercise, we want to do something very simple: during the creation of a new Sales Order, automatically fill in the delivery date (in the case it's not yet filled) before saving it.

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.</p> <p>https://goo.gl/PZqa1Y</p>	 The screenshot shows the SAP Cloud Platform log-on page. It has fields for 'E-mail, ID, or User Name' and 'Password', a 'Remember me' checkbox, a 'Log On' button, and a 'Forgot password?' link. The page is powered by SAP Cloud Platform.
	 The screenshot shows the SAP Web IDE Full-Stack interface. The title bar says 'SAP Web IDE Full-Stack'. The main area is titled 'Workspace' and contains several icons for different projects or components.
<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>	 The screenshot shows the SAP Web IDE Preferences screen under the 'Features' tab. A search bar at the top contains 'business one'. Below it, there is a list of features: Cloud Foundry, Code Check, Code Completion, Code Editor, Default Editors, Features (selected), Git Settings, and Keyboard Shortcuts. A blue button labeled 'Save' with a number '4' is at the bottom right. A callout bubble with number '3' points to the 'ON' switch for the SAP Business One feature.

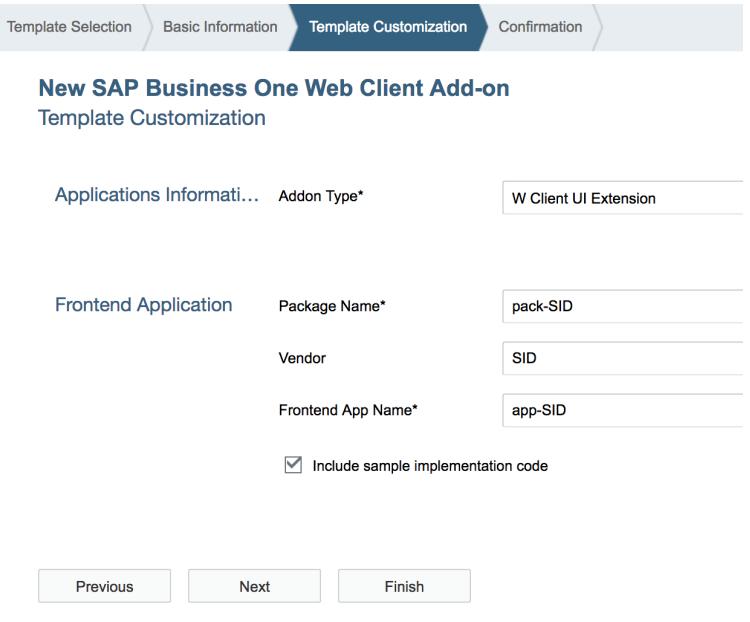


Explanation	Screenshot
<p>Select the Development perspective</p>	
<p>Right click the Workspace folder, select New > Project from Template</p>	
<p>Open the Category dropdown field > select SAP Business One Web Client</p>	



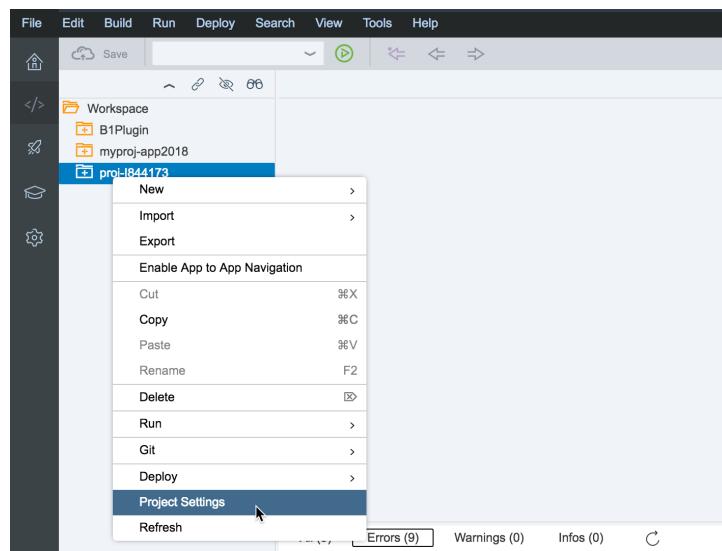
Explanation	Screenshot
<p>Select the SAP Business One Web Client Add-on and hit “Next”</p>	<p>New SAP Business One Web Client Add-on Template Selection</p> <p>Search Category Sort By SAPUI5 Version</p> <p>SAP Business One Web C Sort by recently used SAP Innovation (1.54)</p> <p>SAP Business One Web Client Add-on 3 SAP Business One Web Client Master-Detail View</p> <p>SAP Business One Web Client Add-on</p> <p>Previous Next 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* proj-SID</p> <p>Previous Next</p>



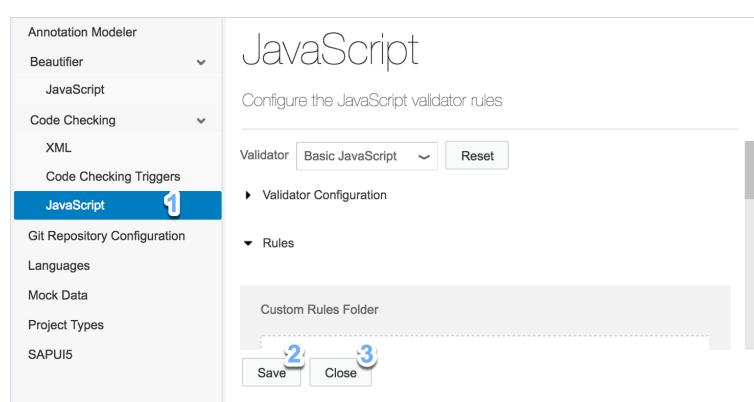
Explanation	Screenshot
Fill in the Template Customization as bellow:	
<p>Add-on Type: W Client UI Extension Package Name: pack-<snumber> Vendor: <snumber> Frontend App Name: app-<snumber></p>	
<p>(Note: DO NOT FORGET THE DASH for the Frontend App Name)</p>	
<p>Check the “Include sample implementation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



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

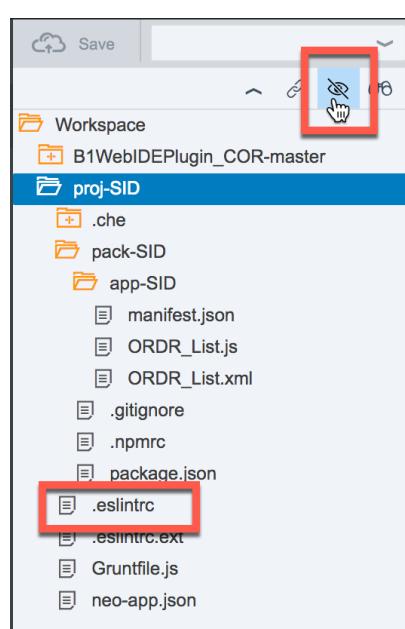


Select Code Checking > JavaScript > Reset > Save and Close.



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 at the end of the file, as indicated in the image in the right side, to activate the correct version for the code checker:

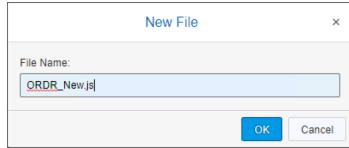
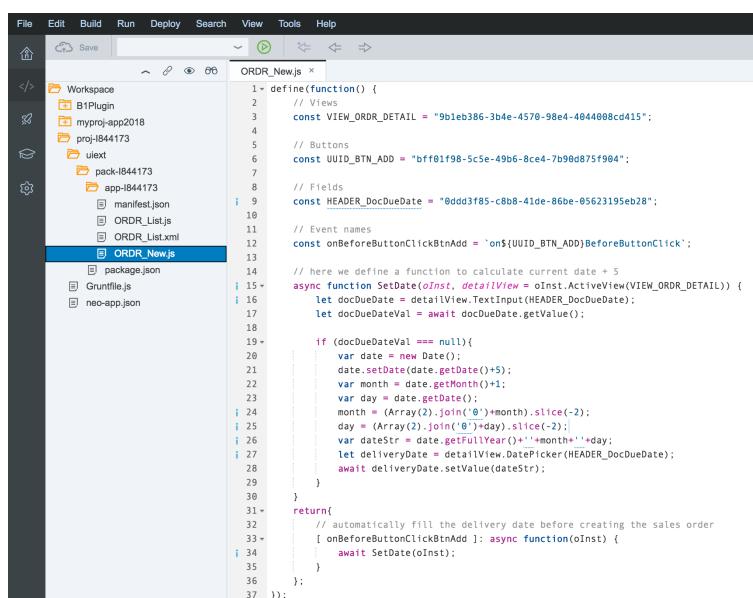
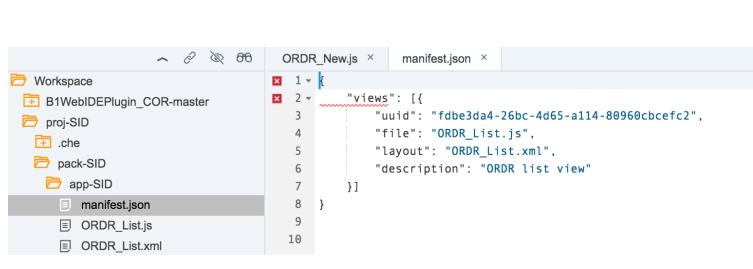
```
"parserOptions": {  
    "ecmaVersion": 2017  
},
```

SAVE IT!

```
File Edit Build Run Deploy Search View Tools Help  
Save 86  
Workspace B1Plugin myproj-app2018 proj-l844173 .che uiext .eslintrc .eslintrc.ext Gruntfile.js neo-app.json  
*.eslintrc x  
449     "crtypes": 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**"

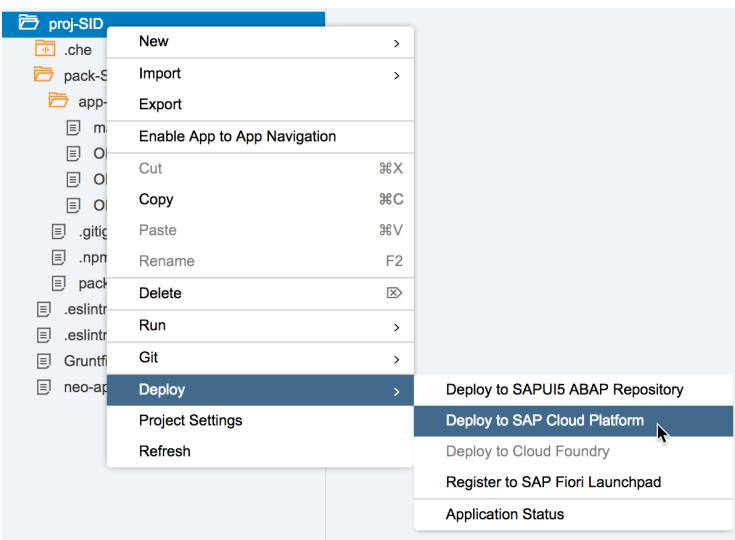
Explanation	Screenshot
<p>Right click the folder <code>proj-<snumber>/pack-<snumber>/app-<snumber></code></p> <p>Select "New" > "File"</p>	
<p>Name it "ORDR_New.js" and hit "OK".</p> <p>This file will handle the Sales Orders Detail View</p>	
<p>Time to write the JavaScript with your custom logic.</p> <p>Open the "ORDR_New.js" file that you've recently created.</p> <p>Copy and paste the code from the link below:</p> <p>https://goo.gl/iQD82Q</p> <p>Save your changes.</p>	 <pre> 1 define(function() { 2 // Views 3 const VIEW_ORDR_DETAIL = "9b1eb386-3b4e-4570-98e4-4044008cd415"; 4 // Buttons 5 const UUID_BTN_ADD = "bff01f98-5c5e-49b6-8ce4-7b90d875f904"; 6 // Fields 7 const HEADER_DocDueDate = "0dd3f85-c8b8-41de-86be-05623195eb28"; 8 // Event names 9 const onBeforeButtonClickBtnAdd = 'on\$({UUID_BTN_ADD})BeforeButtonClick'; 10 // here we define a function to calculate current date + 5 11 async function SetDate(oInst, detailView = oInst.ActiveView(VIEW_ORDR_DETAIL)) { 12 let docDueDate = detailView.TextInput(HEADER_DocDueDate); 13 let docDueDateVal = await docDueDate.getValue(); 14 15 if (docDueDateVal === null) { 16 var date = new Date(); 17 date.setDate(date.getDate() + 5); 18 var month = date.getMonth() + 1; 19 var day = date.getDate(); 20 month = (Array(2).join('0') + month).slice(-2); 21 day = (Array(2).join('0') + day).slice(-2); 22 var dateStr = date.getFullYear() + '-' + month + '-' + day; 23 let deliveryDate = detailView.DatePicker(HEADER_DocDueDate); 24 await deliveryDate.setValue(dateStr); 25 } 26 } 27 return [28 // automatically fill the delivery date before creating the sales order 29 { onBeforeButtonClickBtnAdd: async function(oInst) { 30 await SetDate(oInst); 31 }; 32 }; 33]; 34} 35 36}; 37}); </pre>
<p>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".</p> <p>Note: just ignore the errors displayed for the manifest.json – this is an unreleased version of the web client.</p>	 <pre> 1 [2 { 3 "views": [4 { 5 "uid": "fdbe3da4-26bc-4d65-a114-80960cbcef2", 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 additional code (highlighted in blue in the picture) to the manifest.json file "views" array:</p> <p>https://goo.gl/7dMHB3</p>	<p>The screenshot shows the SAP Business One WebClient Tools interface. At the top, there are two tabs: 'ORDR_New.js' and 'manifest.json'. The 'manifest.json' tab is active, displaying the following JSON code:</p> <pre> 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 "uuid": "9b1eb386-3b4e-4570-98e4-4044008cd415", 11 "file": "ORDR_New.js", 12 "description": "ORDR detail view new" 13 }] } </pre> <p>Below the tabs, a 'Sales Order' view is displayed. The UI Inspector tool is open, showing the current active view ('VIEW: Detail') and its XML structure. A red box highlights the 'uuid' attribute in the XML, which corresponds to the highlighted code in the manifest.json file.</p>

Save the file.

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

Deploy Application to SAP Cloud Platform

Application Details

Deploy a new application Update an existing application

*Account: se996cf12 Get Accounts

Project Name: proj-SID

*Application Name: projsid

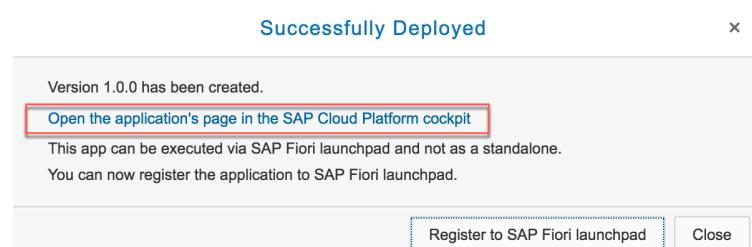
Version Management

*Version: 1.0.0 Activate

Deploy Cancel

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.

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

Copy the application URL.

Actions

Started



Active Version

Active Application Version: 1.0.0

Application URL: <https://projsid-se996cf12.dispatcher.eu3.hana.ondemand.com>

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.

```
C ⓘ Secure | https://projsid-se996cf12.dispatcher.eu3.hana.ondemand.com/pack-SID/package.json
// 20180319130339
// https://projsid-se996cf12.dispatcher.eu3.hana.ondemand.com/pack-SID/package.json

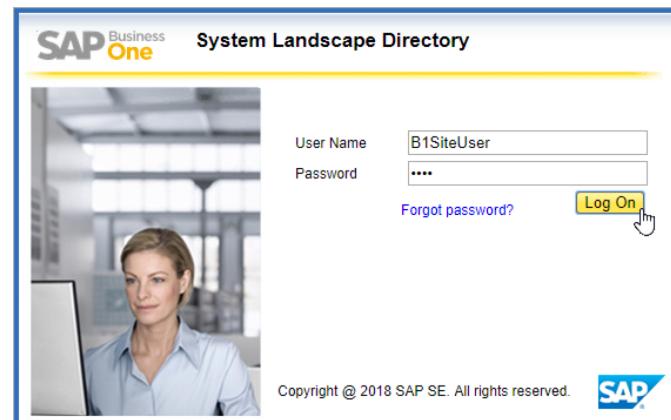
{
  "name": "pack-SID",
  "description": "Thin Client UI Extension addons and Fiori Application addons",
  "version": "0.0.1",
  "vendor": "SID",
  "ui-ext": [
    {
      "folder": "app-SID"
    }
  ]
}
```

Open the Extension Manager:

<http://goo.gl/9UtVW6>

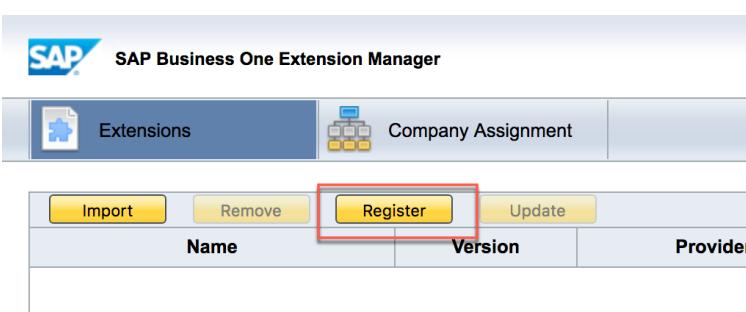
Use the following credentials:

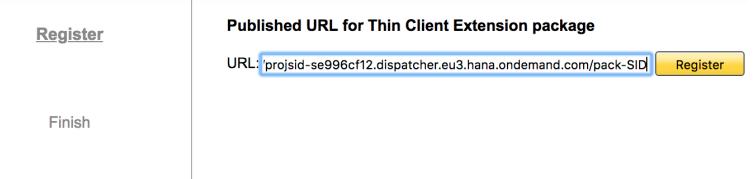
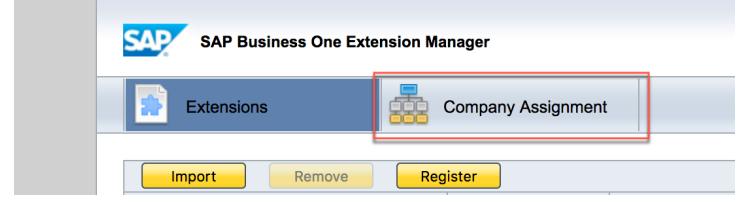
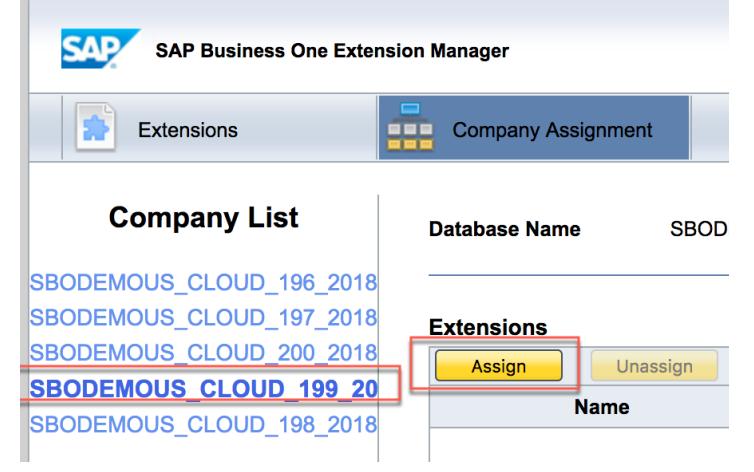
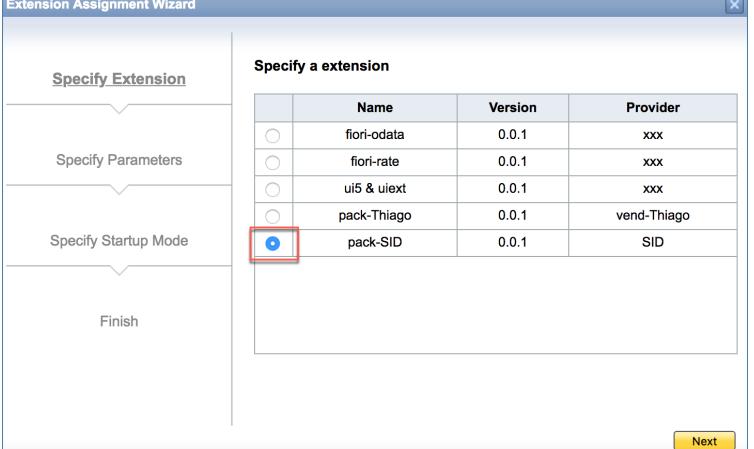
User Name: B1SiteUser
Password: **Check with your instructor**



You will see this form.

Click "Register"



<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 (the tenant is the one your instructor assigned to you at the beginning of the exercise)</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. At the top, there's a navigation bar with 'Extensions' and 'Company Assignment'. Below it is a 'Company List' section containing several database names: SBOTCN05, SBOTCN13, SBOTCN20, SBOTCN08, SBOTCN16, and SBOTCN01. To the right, a 'Database Name' is set to SBOTCN05. Under the 'Extensions' section, there are 'Assign' and 'UnAssign' buttons, followed by a table with columns 'Name' and 'Version'. One entry in the table is 'pack-SID' with version '0.0.1'.

Browse to the B1 Web Client.

Use the specific URL that your instructor has assigned to you.

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

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

Login using below credentials:

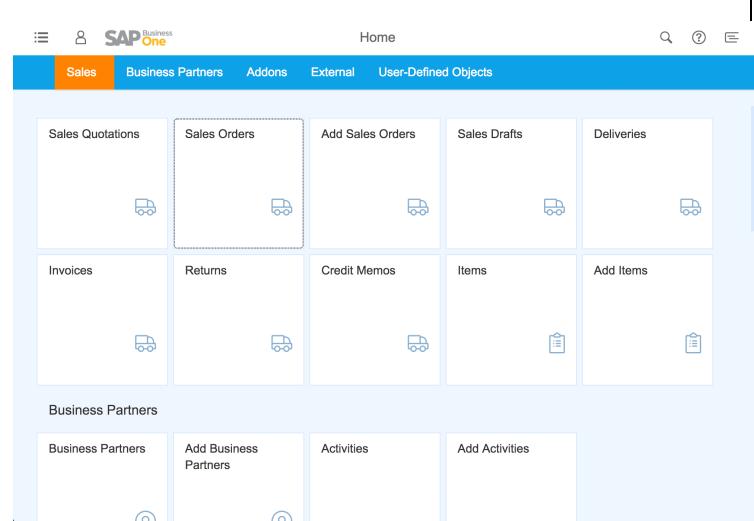
Username: manager
Password: check with your instructor

The screenshot shows the SAP Business One Web Client interface. It features a header with the SAP Business One logo and a user profile for 'Jayson Butler' (Sales). Below the header are two main menu items: 'Sales Quotations' and 'Sales Orders'. A secondary navigation bar shows another user profile for 'OEC Computers'. At the bottom, there's a 'Sign Out' button.

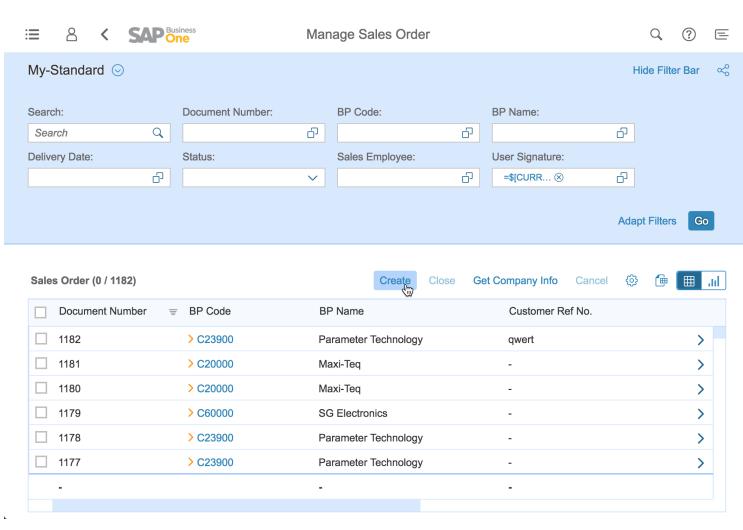
The screenshot shows a login dialog box. The title bar says 'OEC Computers (SBOTCN05)'. It contains fields for 'Username' (filled with 'manager') and 'Password' (filled with '.....'). A large blue 'Login' button is at the bottom.



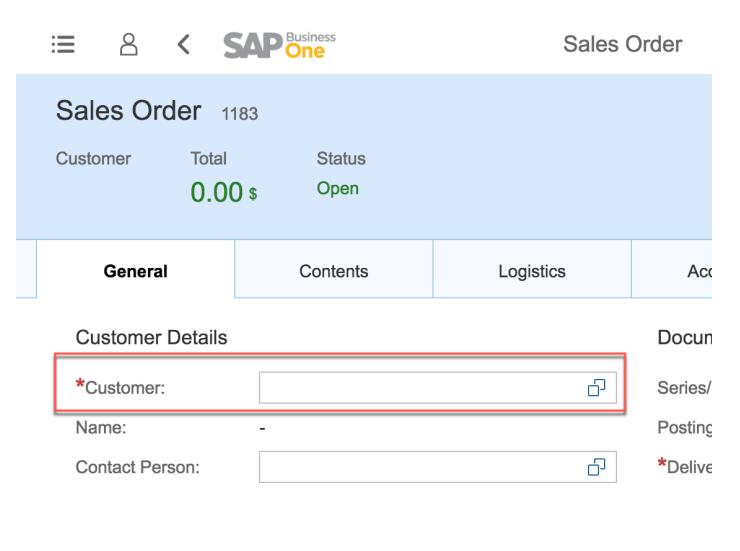
Click the “Sales Orders” tile



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”

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 Fie...
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

Maxi-Teq

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
*Delivery Date: MMM d, y
Document Date: Mar 19, 2018
Sales Employee: Sales Manager
Owner:

Add & View Save as Draft & New Cancel

Contents	Logistics	Accounting	User-Defined Fie...
----------	-----------	------------	---------------------

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!



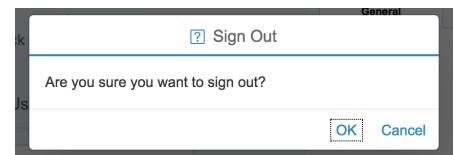
Before continuing, please log out from SAP Business One Web Client, and close the window:

Click the user icon

Select "Sign Out"

Confirm the message

Close the browser





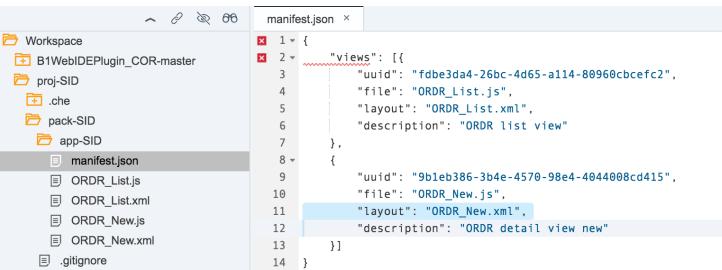
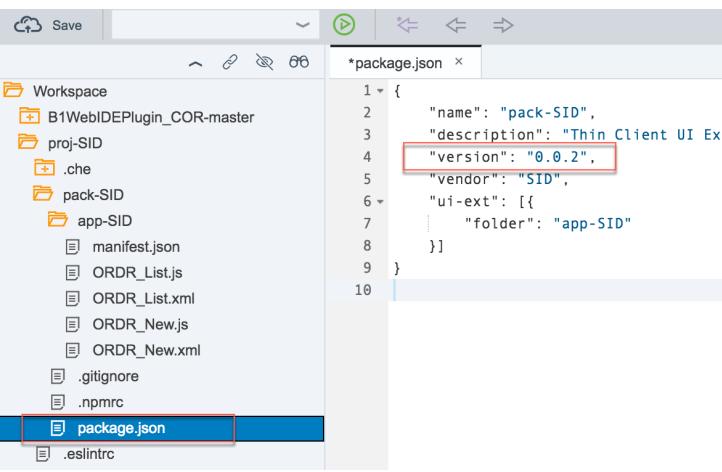
EXERCISE 2: MAKE FIELD “CUSTOMER REFERENCE NUMBER” MANDATORY

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

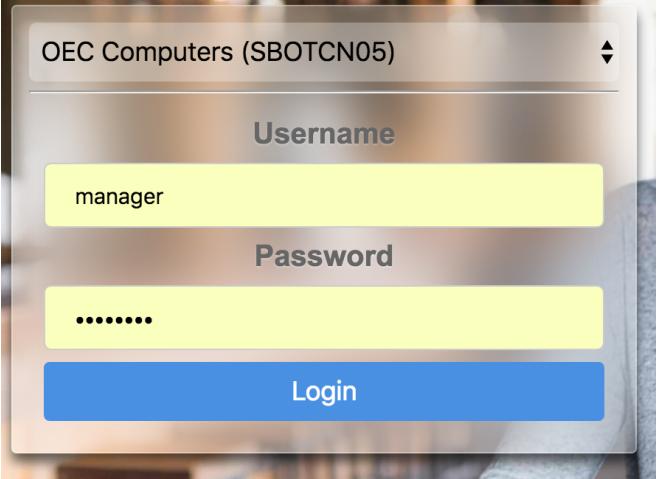
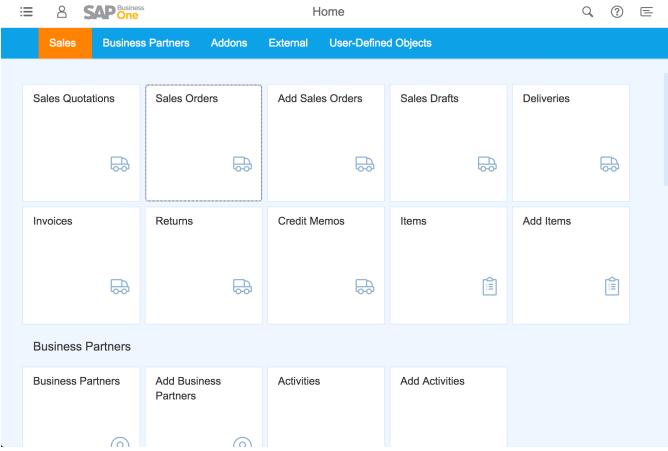
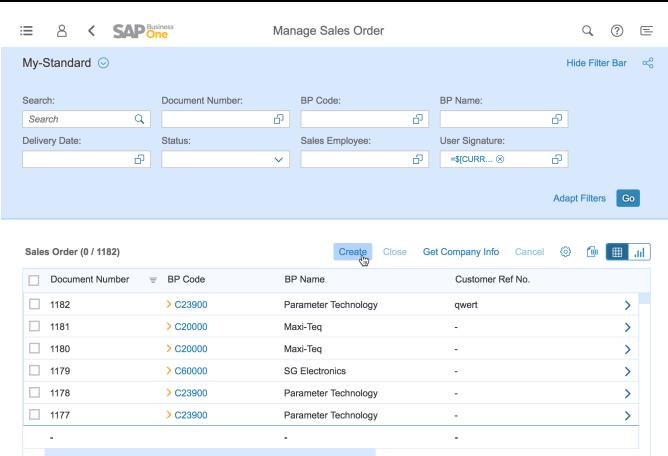
To accomplish changes in the UI components, we need an XML file with the corresponding components and properties.

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</p> <p>Save it.</p>	 <pre> 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</p> <p>Save it.</p>	 <pre> 1 { 2 "views": [3 { 4 "uuid": "fbe3da4-26bc-4d65-a114-80960cbcef2", 5 "file": "ORDR_List.js", 6 "layout": "ORDR_List.xml", 7 "description": "ORDR list view" 8 }, 9 { 10 "uuid": "9b1eb386-3b4e-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> 1 { 2 "name": "pack-SID", 3 "description": "Thin Client UI Extension", 4 "version": "0.0.2", 5 "vendor": "SID", 6 "ui-ext": [7 { 8 "folder": "app-SID" 9 } 10 </pre>

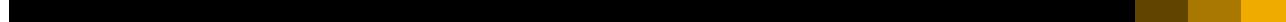
Explanation	Screenshot
<p>Your changes are ready to be deployed and tested.</p> <p>Right click the project main folder > Deploy > Deploy to SAP Cloud Platform</p> <p>It will update the existing application.</p> <p>Click the “Deploy” button.</p>	
<p>Now, before testing the changes, you need to tell SAP 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 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 >></p> <p>Your instructor has assigned an specific URL to you, check with him.</p> <p>Login using below credentials:</p> <p>Company: Check with your instructor Username: manager Password: Check with your instructor 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" and the number "1184" are displayed. Underneath, there are fields for "Customer" (with value "0.00 \$"), "Total" (with value "0.00 \$"), and "Status" (with value "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" sections for "Series/No.:", "Posting Date:", "Delivery Date:", and "Document Date:". Below the customer details, there are fields for "Remarks:" and "Sales Employee:", both with red borders. The entire screenshot is framed by a thick gray 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 standard Windows-style error dialog box. It has a dark gray border and a white interior. At the top center, there is a red exclamation mark icon followed by the word "Error". Below this, a message in black text reads "Missing value: Customer Ref No.". In the bottom right corner of the dialog box, there is a blue "OK" button.</p>

Congratulations!



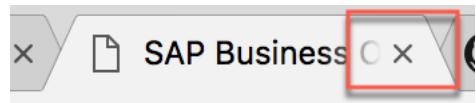
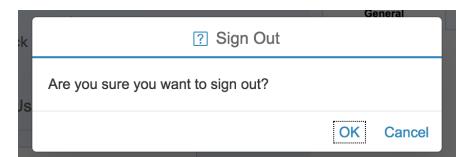
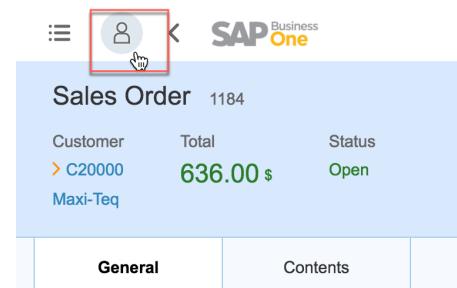
Before continuing, please sign out SAP Business One Web Client, and close the window:

Click the user icon

Select "Sign Out"

Confirm the message

Close the browser

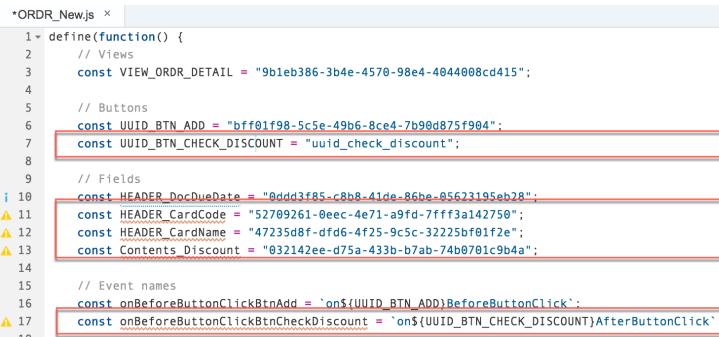


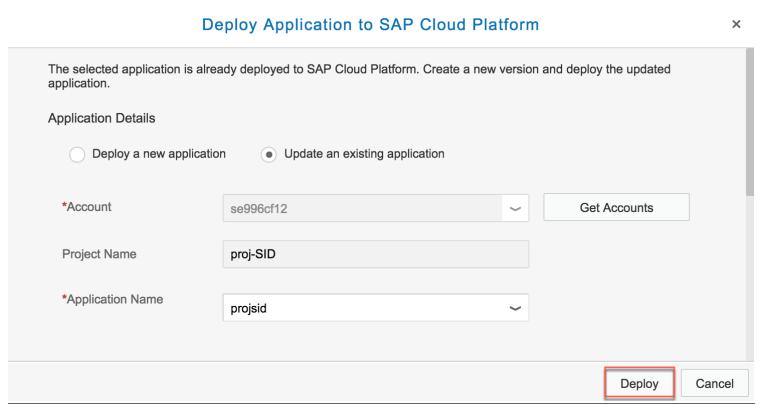
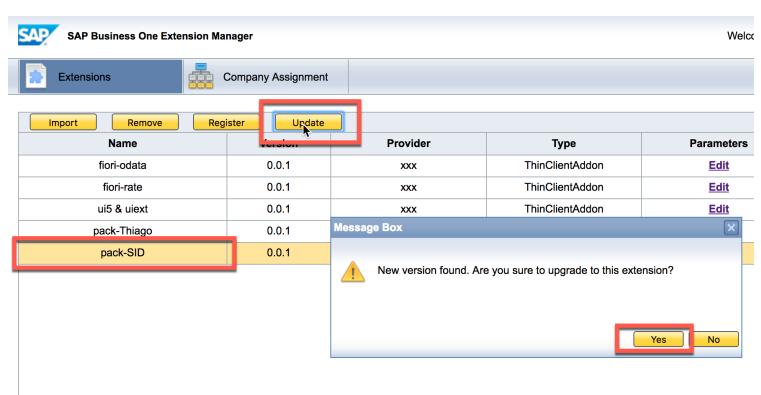


EXERCISE 3: ADD A BUTTON TO THE SALES ORDER TO CHECK BUSINESS PARTNER'S DEBTS

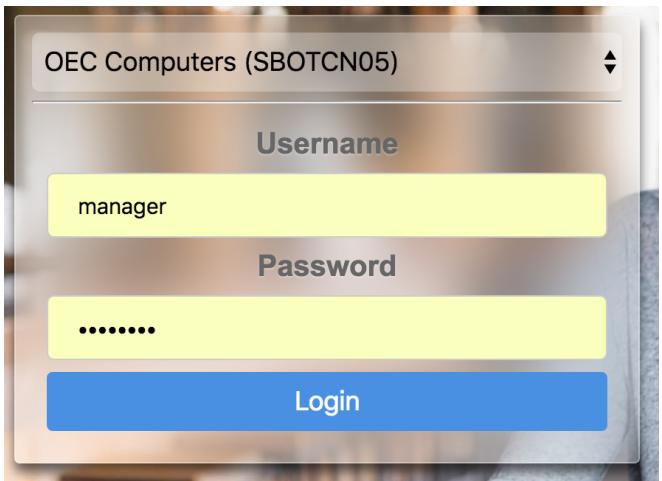
As a last exercise, you will create a new button to the existing Sales Order UI. You will handle the button click to check if the customer has pending debts. If there's no pending debts, alert the user and apply 20% discount to the current Sales Order.

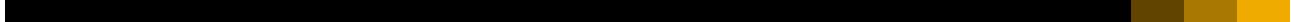
Explanation	Screenshot
Open Google Chrome.	
Access the SCP Web IDE using your S-number credentials. https://goo.gl/PZqa1Y	
Let's add a new button to the existing Sales Order UI. It should be called "Check Discount" and placed after the "Last Record" navigation button.	 UI Inspector code snippet: <pre><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 text="Check Discount" uid="uuid_check_discount"></button></pre>
Open the ORDR_New.xml and add the new button to the form. You can copy and paste the new code lines from here: https://goo.gl/h9MBC3	<pre>1 <overlay> 2 <on uuid="31ffb45d-e42c-4eea-94dc-21d1dbaaa7c7" mandatory="true"/> 3 <after uuid="9366573f-9be8-484a-a37d-9a8b36f6ee1a"> 4 ... 5 <button text="Check Discount" uid="uuid_check_discount"></button> 6 </after> 7 </overlay></pre>

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 x 1- define(function() { 2 // Views 3 const VIEW_ORDER_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 = "0ddd3f85-c8b8-41de-86be-05623195eb28"; 11 const HEADER_CardCode = "52709261-0ec-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 onBeforeButtonClickListenerAdd = `on\${UUID_BTN_ADD}BeforeButtonClick`; 17 const onBeforeButtonClickListenerCheckDiscount = `on\${UUID_BTN_CHECK_DISCOUNT}AfterButtonClick`; 18</pre>
<p>Let's start coding the decision logic for the discount.</p> <p>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 x 35 } 36 37 async function SetDiscount(oInst, detailView = oInst.ActiveView(VIEW_ORDER_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-web/s1/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}`, [63 {label: "OK", key: "OK"} 64]); 65 66 // If no debts apply the discount 67 if (ab >= 0) { 68 let discount = detailView.TextInput(Contents_Discount); 69 await discount.setValue("20"); 70 } 71 } 72</pre>
<p>Now append the event handler of the new button:</p> <p>https://goo.gl/RxPNMc</p> <p>Save it!</p>	 <pre>74 return; 75 // automatically fill the delivery date before creating the sales order 76 [onBeforeButtonClickListenerAdd]: async function(oInst) { 77 await SetDate(oInst); 78 }, 79 80 [onBeforeButtonClickListenerCheckDiscount]: async function(oInst) { 81 await SetDiscount(oInst); 82 } 83};</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>	 <pre>package.json x ORDR_New.js x ORDR_New. 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>

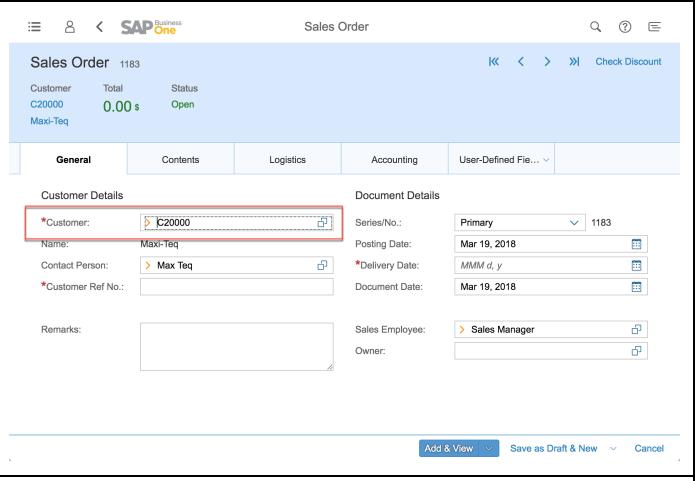
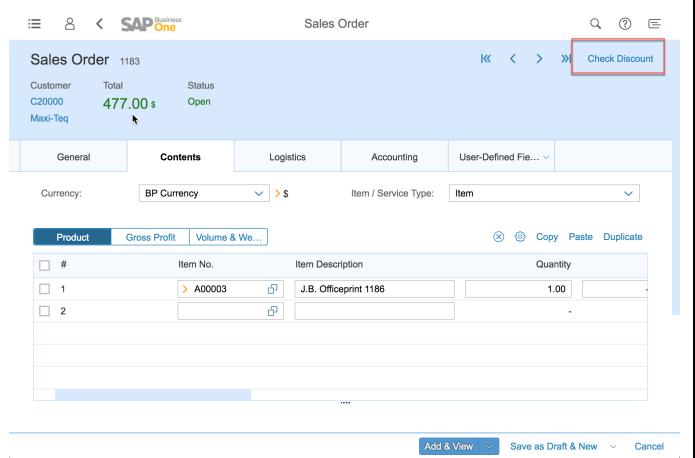
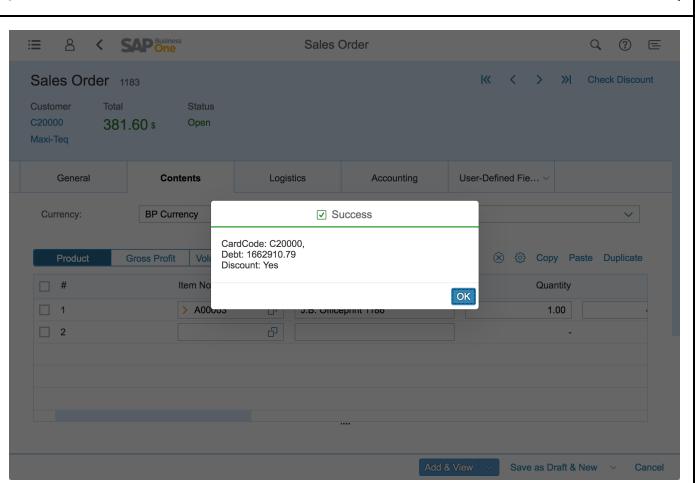
Explanation	Screenshot
<p>Your changes are ready to be deployed and tested. Right click the project main folder > Deploy > Deploy to SAP Cloud Platform.</p> <p>Click the “Deploy” button.</p>	
<p>Now, before testing the changes, you need to tell SAP Business One, that your app is updated.</p> <p>Open 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>	

Now, test the changes.

Explanation	Screenshot
<p>Open Google Chrome</p>	
<p>Open the SAP Business One Web Client >></p> <p>Your instructor has assigned an specific URL to you, check with him.</p> <p>Login using below credentials:</p> <p>Company: Check with your instructor Username: manager Password: Check with your instructor Click Login</p>	



Explanation	Screenshot
Click the Sales Orders tile.	The screenshot shows the SAP Business One Home screen. The top navigation bar has tabs for Sales, Business Partners, Addons, External, and User-Defined Objects. Below the tabs is a grid of tiles. The 'Sales Orders' tile is highlighted with a dashed border and a blue icon. Other tiles include Sales Quotations, Add Sales Orders, Sales Drafts, Deliveries, Invoices, Returns, Credit Memos, Items, and Add Items. A section for Business Partners is also visible.
Click "Create"	The screenshot shows the 'Manage Sales Order' screen. At the top, there are search fields for Document Number, BP Code, and BP Name, and dropdowns for Delivery Date, Status, Sales Employee, and User Signature. Below this is a table titled 'Sales Order (0 / 1182)'. A 'Create' button is located at the top right of the table area. The table lists several sales orders with columns for Document Number, BP Code, BP Name, and Customer Ref No.
Validate that the new button "Check Discount" is displayed, don't click it.	The screenshot shows the 'Sales Order' screen for order 1183. The top header includes a 'Check Discount' button. The main area displays customer details (Customer: 0.00 s, Total: 0.00 s, Status: Open) and document details (Series/No.: Primary, Posting Date: Mar 19, 2018, Delivery Date: MMM d, y, Document Date: Mar 19, 2018). A remarks field and sales employee/owner fields are also present. Buttons at the bottom include Add & View, Save as Draft & New, and Cancel.

Explanation	Screenshot																														
<p>Select a customer</p>	 <p>Sales Order 1183</p> <p>Customer Total Status C20000 0.00 \$ Open</p> <p>Maxi-Teq</p> <p>General Contents Logistics Accounting User-Defined File...</p> <p>Customer Details</p> <p>*Customer: C20000</p> <p>Name: Maxi-Teq</p> <p>Contact Person: Max Teq</p> <p>*Customer Ref No.: </p> <p>Document Details</p> <p>Series/No.: Primary 1183</p> <p>Posting Date: Mar 19, 2018</p> <p>*Delivery Date: MMM d, y</p> <p>Document Date: Mar 19, 2018</p> <p>Remarks: </p> <p>Sales Employee: Sales Manager</p> <p>Owner: </p> <p>Add & View Save as Draft & New 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>Sales Order 1183</p> <p>Customer Total Status C20000 477.00 \$ Open</p> <p>Maxi-Teq</p> <p>General Contents Logistics Accounting User-Defined File...</p> <p>Currency: BP Currency \$ Item / Service Type: Item</p> <table border="1"> <thead> <tr> <th>Product</th> <th>Gross Profit</th> <th>Volume & We...</th> <th>Copy</th> <th>Paste</th> <th>Duplicate</th> </tr> </thead> <tbody> <tr> <td>#</td> <td>Item No.</td> <td>Item Description</td> <td colspan="3">Quantity</td> </tr> <tr> <td>1</td> <td>A00003</td> <td>J.B. Officeprint 1186</td> <td colspan="3">1.00</td> </tr> <tr> <td>2</td> <td></td> <td></td> <td colspan="3">-</td> </tr> <tr> <td colspan="6">....</td> </tr> </tbody> </table> <p>Add & View Save as Draft & New Cancel</p>	Product	Gross Profit	Volume & We...	Copy	Paste	Duplicate	#	Item No.	Item Description	Quantity			1	A00003	J.B. Officeprint 1186	1.00			2			-							
Product	Gross Profit	Volume & We...	Copy	Paste	Duplicate																										
#	Item No.	Item Description	Quantity																												
1	A00003	J.B. Officeprint 1186	1.00																												
2			-																												
....																															
<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>Sales Order 1183</p> <p>Customer Total Status C20000 381.60 \$ Open</p> <p>Maxi-Teq</p> <p>General Contents Logistics Accounting User-Defined File...</p> <p>Currency: BP Currency</p> <p>Success</p> <p>CardCode: C20000, Debt: 1662910.79 Discount: Yes</p> <p>OK</p> <p>Add & View Save as Draft & New Cancel</p>																														



Explanation	Screenshot
Note the Sales Order amount, it is reduced by 20%	<p>The screenshot shows the SAP Business One Sales Order screen. The total amount at the top is 254.40 \$, which is highlighted with a red box. Below the table, there are buttons for 'Add', 'Add as Draft & New', and 'Cancel'.</p>
Click "Add"	<p>The screenshot shows the SAP Business One Sales Order screen after an item has been added. A confirmation message box is displayed in the center, stating 'Operation completed successfully'. The 'OK' button in this message box is highlighted with a red box.</p>
Click "OK"	

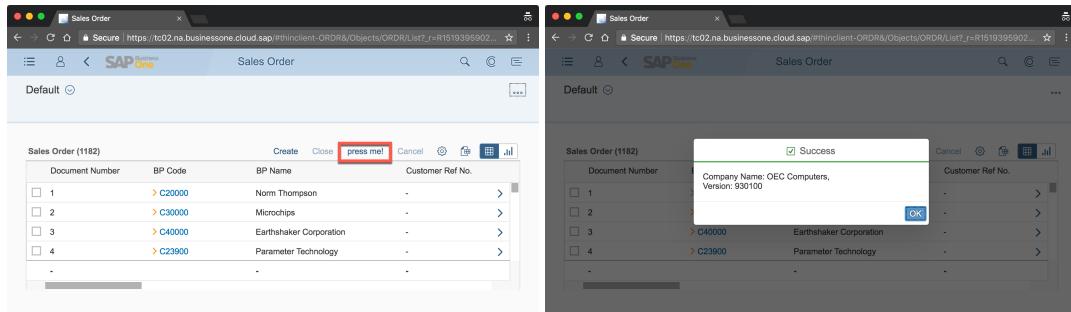
Congratulations! You have finished the exercises on the extensibility of SAP Business One Web Client. Please, give us your feedback.



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 onAfterButtonClickBtnGetCompanyInfo = `on${BTN_GET_COMPANY_LIST}AfterButtonClick`;
7
8     return {
9         [onAfterButtonClickBtnGetCompanyInfo]: 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",
17                 key: "OK"
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.