

openSAP Evolved Web Apps with SAPUI5

Week 1 Unit 6: Getting Ready for the Course Exercises

Exercises

PUBLIC



TABLE OF CONTENTS

- GETTING READY FOR THE COURSE EXERCISES 3
- Using the coding exercise handouts 4
- Integrate Rufus to your app..... 5
- Working with Rufus to validate your exercise code..... 6
- Clone the course repository from GitHub to your SAP Web IDE..... 8
- Update the exercise solutions from GitHub in your SAP Web IDE.....10
- Reset your app project with the exercise solution from GitHub.....11
- RELATED MATERIAL.....12

GETTING READY FOR THE COURSE EXERCISES

Summary

In this unit, you will learn how to perform the exercises in this course. You will get an overview on how the coding exercise handouts are structured. In addition, you are shown how to integrate Rufus – your personal assistant – so he can support you in your exercises and validate your code. In case you get stuck with any exercise, you will learn where to get the solution and how you can repair your app with it.

Preview

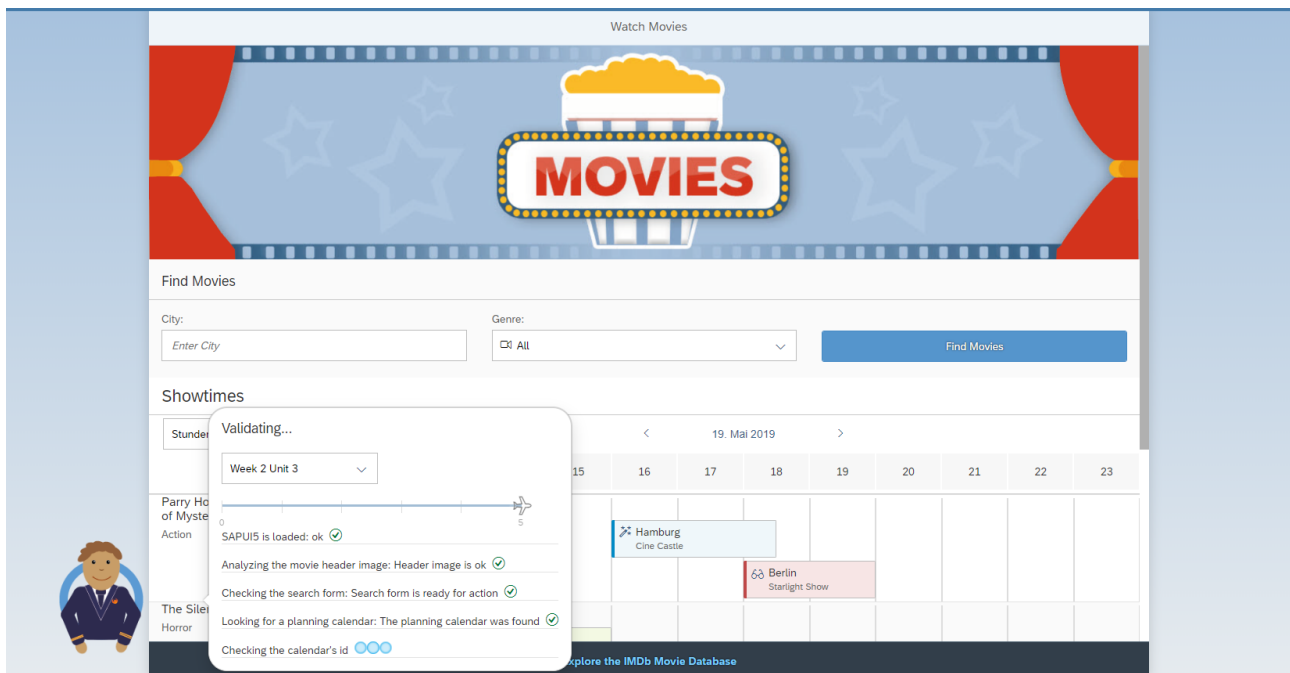
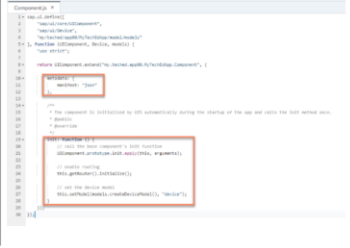
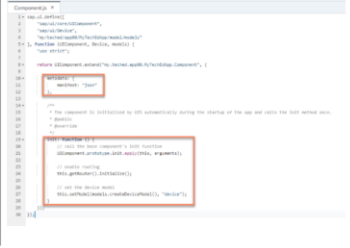
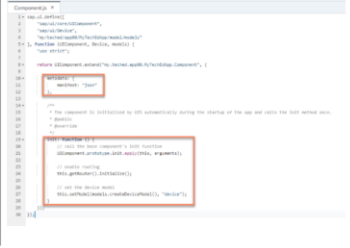
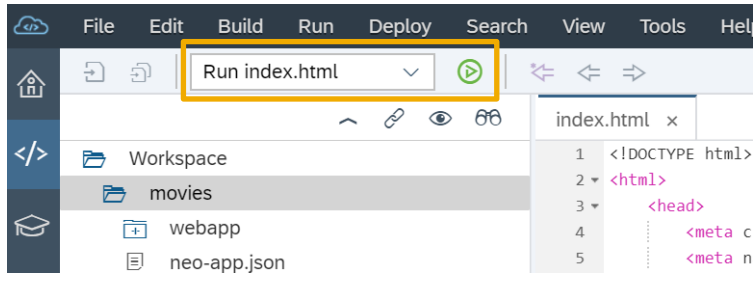


Figure 1 – Rufus checking the code in your app

Using the Coding Exercise Handouts

This section explains how to use the handouts for the exercises in this course.

Explanation	Screenshot				
<p>1. Open the exercise document after you have watched the unit video.</p> <p>Follow the steps described in the document by applying the changes marked in yellow to your app project in SAP Web IDE.</p> <p>Note: Exercise Types The exercises for a unit can be either configuration steps, code changes, or a mixture of both.</p>	<p>Add code to trigger navigation</p> <p>The last step to make your navigation work is the coding of the event handler. But let's take a step back for a moment and learn about a piece of configuration that came with the SAPUI5 Application template: the router initialization in the <code>Component.js</code> file. The <code>Component.js</code> is loaded first, and in it the <code>manifest.json</code> is called: at this point, your routing settings are loaded and are ready to be used.</p> <table border="1"> <thead> <tr> <th>Explanation</th><th>Screenshot</th></tr> </thead> <tbody> <tr> <td> <p>All UI assets are encapsulated in a component that is instantiated from your <code>index.html</code> page.</p> <p>Components are independent and reusable parts used in SAPUI5 applications.</p> <ol style="list-style-type: none"> 1. Open your <code>Component.js</code> file and look at the implementation of the component. Note that the metadata is loaded from the "manifest". 2. The <code>init</code> function configures additional models that are not defined in the manifest. It also initializes the router. </td><td>  </td></tr> </tbody> </table> <p>webapp/view/App.view.xml</p> <pre> <f:SimpleForm </f:SimpleForm> <PlanningCalendar id="calendar" startDate="(path: 'movies'/initDate, formatter: '.formatter.formatDate')" rows="(movies>/movies)" appointments@realization="filled" appointmentSelect="onAppointmentSelect(\$(\$parameters/appointment))" <toolbarContent> ... </pre> <p>Add the <code>planningCalendar</code> event <code>appointmentSelect</code> and the name of the function it triggers to your xml code.</p>	Explanation	Screenshot	<p>All UI assets are encapsulated in a component that is instantiated from your <code>index.html</code> page.</p> <p>Components are independent and reusable parts used in SAPUI5 applications.</p> <ol style="list-style-type: none"> 1. Open your <code>Component.js</code> file and look at the implementation of the component. Note that the metadata is loaded from the "manifest". 2. The <code>init</code> function configures additional models that are not defined in the manifest. It also initializes the router. 	
Explanation	Screenshot				
<p>All UI assets are encapsulated in a component that is instantiated from your <code>index.html</code> page.</p> <p>Components are independent and reusable parts used in SAPUI5 applications.</p> <ol style="list-style-type: none"> 1. Open your <code>Component.js</code> file and look at the implementation of the component. Note that the metadata is loaded from the "manifest". 2. The <code>init</code> function configures additional models that are not defined in the manifest. It also initializes the router. 					
<p>2. Code changes follow the following pattern:</p> <ol style="list-style-type: none"> a. The header indicates the file to be changed. In case the file needs to be created <i>New</i> is put at the end in brackets b. Lines marked in yellow need to be added or exchanged. c. Lines marked in red need to be deleted. d. The changes are explained below the code block. 	<p>webapp/controller/App.controller.js a</p> <pre> ... oAppointmentsBinding.filter(oFilterCity); ... onAppointmentSelect: function (oAppointment) { var oContext = oAppointment.getBindingContext("movies"); sPath = oContext.getPath(); var aParameters=sPath.split("/"); UIComponent.getRouterFor(this).navTo("Detail", { movieId: aParameters[2], appointmentId: aParameters[4] }); } ... </pre> <p>b</p> <p>d</p> <p>Add the function <code>onAppointmentSelect</code> to your App controller and pass <code>oAppointment</code> as a parameter into it. Then you define two variables, <code>oContext</code> for capturing the binding context, and <code>sPath</code> to capture the data binding information.</p> <p>Now you need to extract the position of the selected movie and appointment from <code>sPath</code> – so you split it at each slash and save the result in an array. Then you call the route you defined previously in the app descriptor and provide the two mandatory parameters giving their respective position in the array <code>aParameters</code>.</p> <p>In the <code>Detail</code> controller, you now have to receive those two parameters to display the correct information.</p>				
<p>3. Check the result of your changes frequently by running the app.</p> <p>Note: When You're Working with SAP Web IDE): Make sure that your changed files are saved. Unsavd changes are marked with an asterisk (*) behind the filename.</p>					

Integrate Rufus to Your App

Rufus – your personal flight assistant – will check the course exercises and only needs to be set up once for each new app. He will help you to stay on course with the exercises.

webapp/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>movies</title>
    <script id="sap-ui-bootstrap"
      src="../../resources/sap-ui-core.js"
      data-sap-ui-theme="sap_belize"
      data-sap-ui-resourceroots='{ "opensap.movies": "." }'
      data-sap-ui-compatVersion="edge"
      data-sap-ui-oninit="module:sap/ui/core/ComponentSupport"
      data-sap-ui-async="true"
      data-sap-ui-frameOptions="trusted">
    </script>
    <script src="https://sap.github.io/openSAP-ui5-
course/Rufus.js"></script>
  </head>
  <body class="sapUiBody">
    <div data-sap-ui-component data-name="opensap.movies" data-
id="container" data-settings='{ "id" : "movies" }'></div>
  </body>
</html>
```

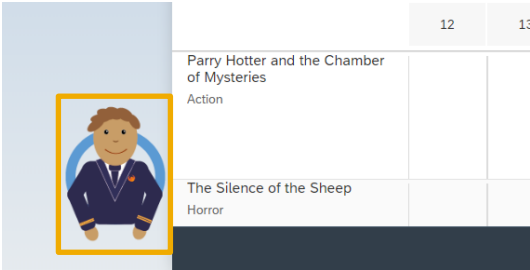
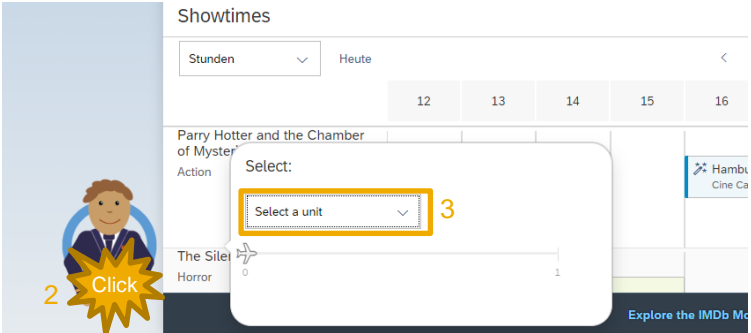
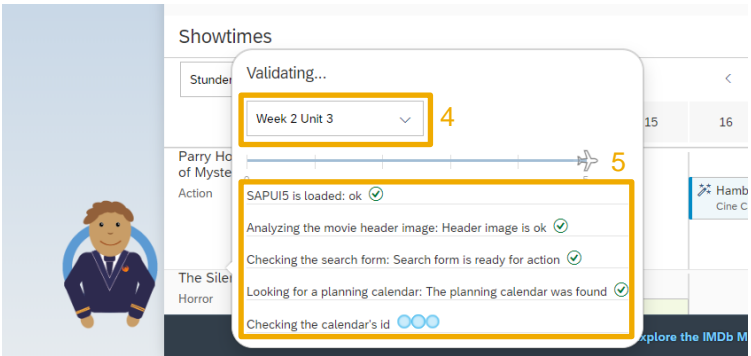
Rufus is a validator script for this course and is hosted on a public GitHub repository. Just add the script tag after the UI5 bootstrap tag. It injects a button that can run tests against your application coding to verify the functionality that you have added.

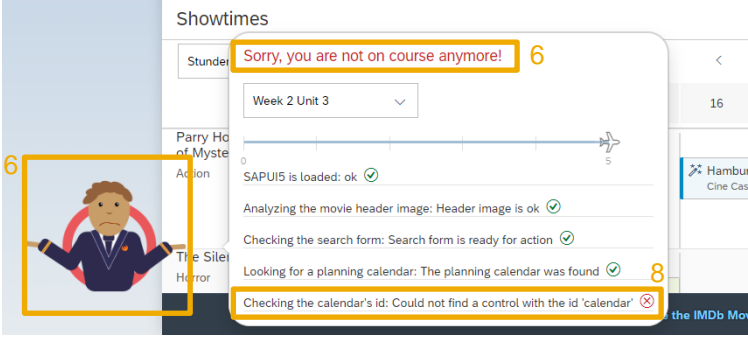
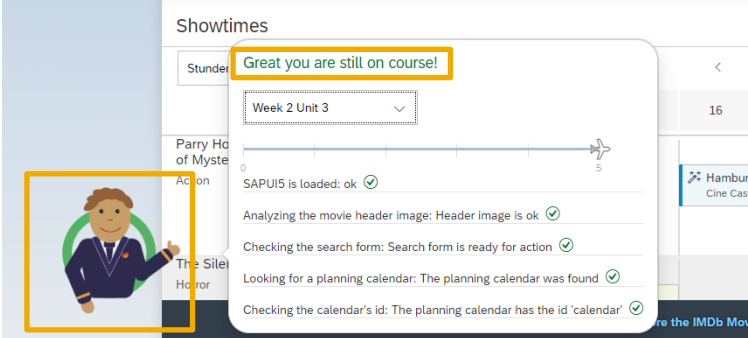
Note: Validator Issues

As the URL and functionality of the validator may change at short notice during the course, be sure to check the forum announcements for additional details. Running the validator is optional, but we recommend you check your exercise coding with it.

Working with Rufus to Validate Your Exercise Code

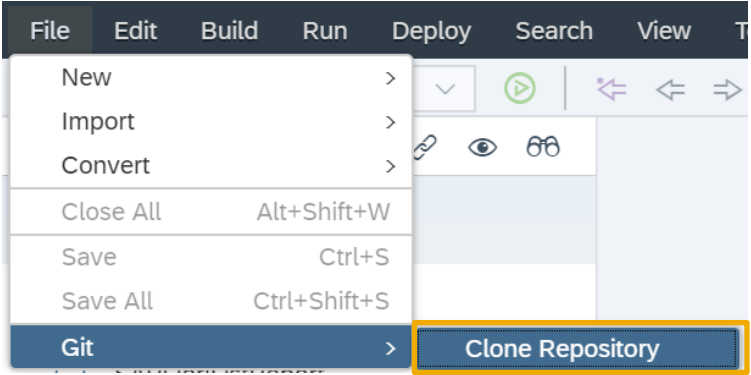
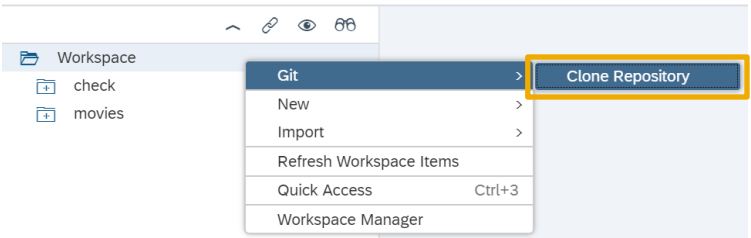
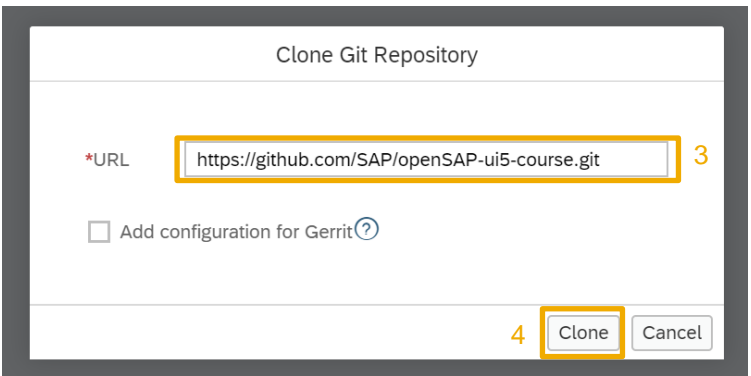
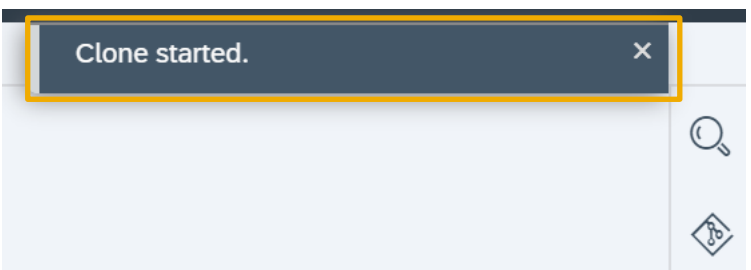
The validation script will check the exercises starting from Week 2 Unit 1 to Week 4 Unit 2 as well as the challenges plus the units of the optional weeks. Validating your code is optional, but it may be helpful to check if your exercise code is correct.

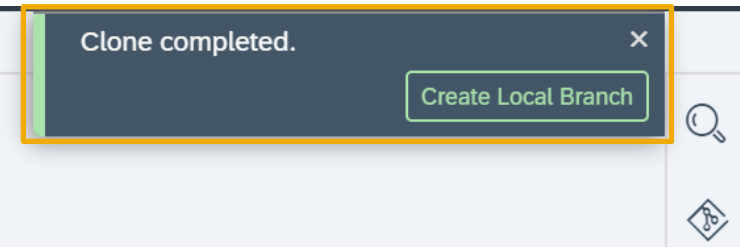
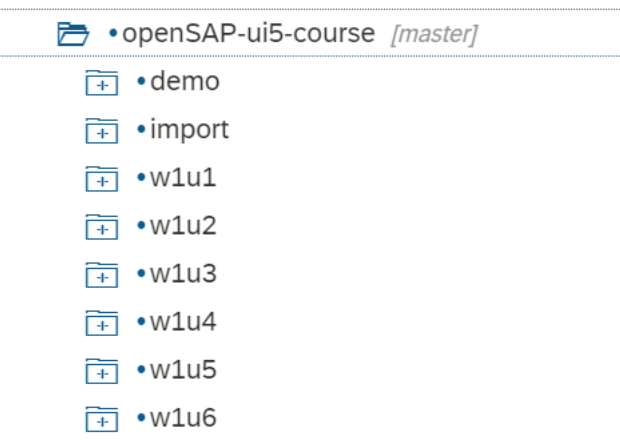
Explanation	Screenshot
<ol style="list-style-type: none"> After adding Rufus to your app, you should see him in the lower left area of your screen whenever you run the app. 	
<ol style="list-style-type: none"> Click on the picture, or press F9 to open the validator menu. Click the <i>Select</i> menu to open the unit selection. <p>Note: Cookies The first time you call the validator, or when you clear your browser cookies, the selection is empty. Afterwards, the last executed unit will be saved as a cookie for your convenience.</p>	
<ol style="list-style-type: none"> Choose the unit that you are currently working on and that you want to validate. The tests will automatically start to run, and the progress is updated after each test is executed. <p>Note: Previous Tests As you build up your app during the course, you will refactor some parts. Therefore, not all tests from the previous units will run successfully.</p>	

Explanation	Screenshot
<p>6. If there is a problem with your exercise coding, the report status of the tests changes to red, and Rufus will be unhappy.</p> <p>7. Check the error message, fix the code that causes the issue, and run Rufus again.</p> <p>Note: Test Logic The tests will check that certain conditions are fulfilled in your app. They rely on</p> <ul style="list-style-type: none"> – specific controls being used, – nesting orders, – aggregations being filled, – IDs set to a value that is specified in the exercise documents. <p>You need to stick closely to the solution exercises, otherwise the validation will probably not work.</p>	 <p>The screenshot shows the 'Showtimes' application interface. On the left, a list of movies is visible, including 'Stunder', 'Parry Ho of Myste', 'Action', 'The Sile', and 'Horror'. On the right, a test log is displayed. The log shows several successful tests with green checkmarks: 'SAPUI5 is loaded: ok', 'Analyzing the movie header image: Header image is ok', 'Checking the search form: Search form is ready for action', and 'Looking for a planning calendar: The planning calendar was found'. However, the final test, 'Checking the calendar's id: Could not find a control with the id 'calendar'', has failed, indicated by a red X. A message box at the top right of the log area says 'Sorry, you are not on course anymore!' with a red border. Rufus, the character on the left, is shown with a sad expression, indicated by a red circle around his head. A yellow box highlights Rufus and the failed test message.</p>
<p>8. Once all tests have run successfully, the report status changes to green. Rufus looks happy.</p>	 <p>The screenshot shows the 'Showtimes' application interface. On the left, the same list of movies is visible. On the right, the test log shows all tests passing with green checkmarks, including the final test: 'Checking the calendar's id: The planning calendar has the id 'calendar''. A message box at the top right of the log area says 'Great you are still on course!' with a green border. Rufus, the character on the left, is shown with a happy expression, indicated by a green circle around his head. A yellow box highlights Rufus and the successful test message.</p>

Clone the Course Repository from GitHub to Your SAP Web IDE

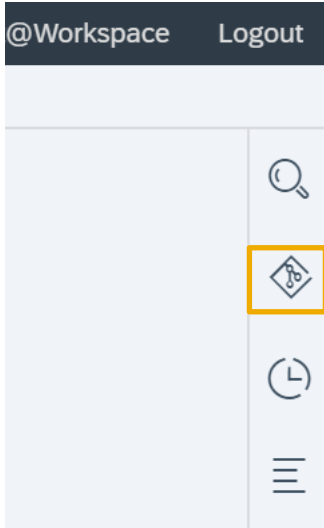
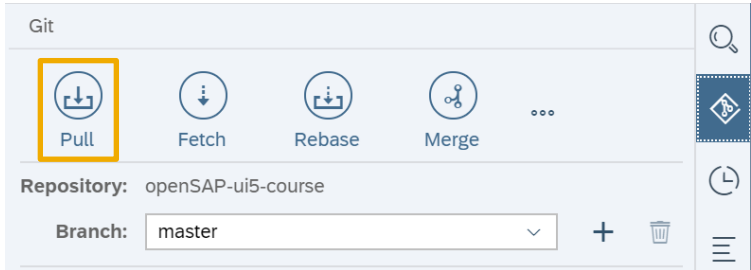
This section explains how to clone the course repository from the public GitHub to your SAP Web IDE workspace to get the solution for each unit exercise.

Explanation	Screenshot
1. In SAP Web IDE, select <i>Workspace</i> , and choose <i>File</i> → <i>Git</i> → <i>Clone Repository</i> .	 A screenshot of the SAP Web IDE interface. The top menu bar includes 'File', 'Edit', 'Build', 'Run', 'Deploy', 'Search', and 'View'. The 'Git' menu is open, showing options like 'New', 'Import', 'Convert', 'Close All', 'Save', 'Save All', and 'Clone Repository'. The 'Clone Repository' option is highlighted with a yellow box.
2. Alternatively, you can also right-click on <i>Workspace</i> and choose <i>Git</i> → <i>Clone Repository</i> .	 A screenshot of the SAP Web IDE workspace. The 'Workspace' folder is selected, and a right-click context menu is open. The 'Git' menu is open, and 'Clone Repository' is highlighted with a yellow box.
3. Enter the following URL in the configuration dialog: https://github.com/SAP/openSAP-ui5-course.git 4. Choose <i>Clone</i> .	 A screenshot of the 'Clone Git Repository' dialog box. The '*URL' field contains the text 'https://github.com/SAP/openSAP-ui5-course.git' and is highlighted with a yellow box. The 'Clone' button is highlighted with a yellow box. A yellow box with the number '3' is next to the URL field, and a yellow box with the number '4' is next to the 'Clone' button.
5. SAP Web IDE will inform you about the clone process in the message bar.	 A screenshot of the SAP Web IDE message bar. A message box is displayed with the text 'Clone started.' and a close button (X) in the top right corner. The message box is highlighted with a yellow box.

Explanation	Screenshot
<p>6. As soon as the clone process is completed, you will get a success message.</p> <p>Note: Ignore Untracked Files If a dialog pops up asking you about untracked files immediately after cloning the repository, choose <i>ignore untracked files</i>.</p>	 <p>The screenshot shows a dark blue dialog box with the text 'Clone completed.' and a close button (X) in the top right corner. Below the text is a button labeled 'Create Local Branch'. The dialog is overlaid on a light blue background that appears to be a software interface.</p>
<p>7. The Git project with the solution is added to your workspace. You have successfully cloned the course repository to your SAP Web IDE workspace.</p>	 <p>The screenshot shows a file explorer view of a Git repository. The root directory is 'openSAP-ui5-course' with the current branch set to '[master]'. Below the root, there is a list of subdirectories: 'demo', 'import', 'w1u1', 'w1u2', 'w1u3', 'w1u4', 'w1u5', and 'w1u6'. Each directory is preceded by a small folder icon with a plus sign.</p>

Update Exercise Solutions from GitHub in Your SAP Web IDE

The course repository on GitHub might be updated from time to time in case issues are reported, therefore it is recommended to check at the beginning of a new unit if any changes have been made on the master repository and, if that be the case, to update your repository accordingly. This section explains how to keep your clone up to date with the course repository.

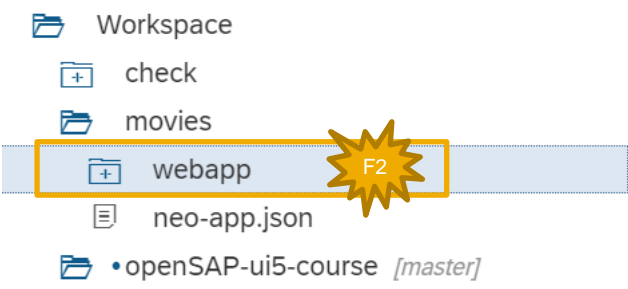
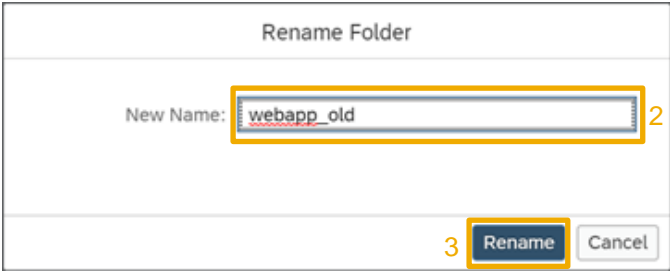
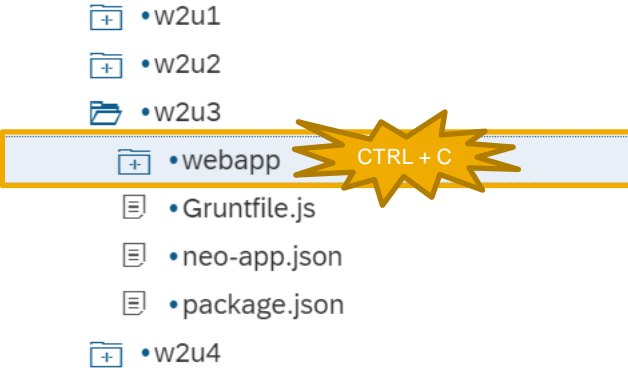
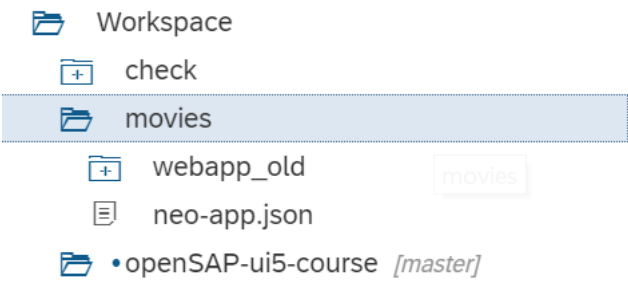
Explanation	Screenshot
1. In SAP Web IDE, select the folder <code>openSAP-ui5-course</code> , and open the Git pane on the right side.	
2. Choose <i>Pull</i> . Note: Automatic Updates Any changes in the course repository will be fetched and downloaded from the course repository and immediately updated in your workspace to match the changed content.	

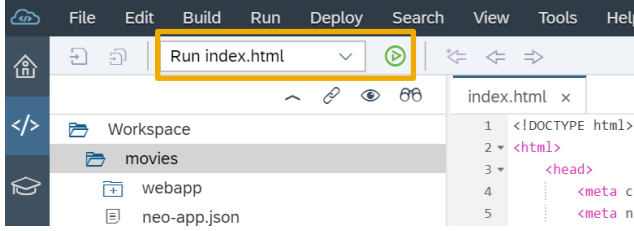
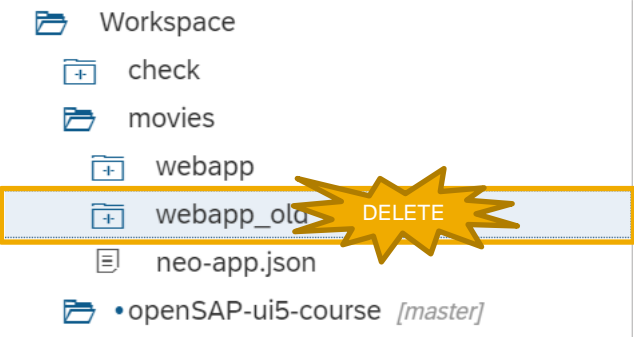
Note: Merge Conflicts

In case you made any manual changes to this project in your SAP Web IDE, this might lead to conflicts making it difficult or impossible to incorporate the changes from the course repository to your local clone. Therefore, we recommend using a local copy of the respective unit folder in case you want to experiment with it.

Reset Your App Project with the Exercise Solution from GitHub

In this section, you will learn how you can reset your app project with the exercise solution from the course repository in case you get stuck.

Explanation	Screenshot
<p>1. Open your app project, and select the file or folder you want to reset. Press F2 to rename the folder.</p> <p>Note: Override Folders For technical reasons, it is not possible to override folders in SAP Web IDE. You must delete or rename the old content in the project folder first.</p>	
<p>2. Assign a new name for the folder (i.e. by adding _old to the end of the origin name)</p> <p>3. Choose <i>Rename</i>.</p>	
<p>4. Open the cloned project, and navigate to the respective unit folder (e.g. <i>w2u3</i> is week 2 unit 4) and copy (CTRL+C).</p>	
<p>5. Go back to your app project, and select the parent folder of the folder you renamed. Paste the copied content to the folder (CTRL+V).</p>	

Explanation	Screenshot
6. Test your app to verify that everything is on track again.	
7. (optional) Select the renamed folder, and delete it by pressing the Delete key. Your app project is now reset with the GitHub content you copied.	

RELATED MATERIAL

- [Course Solution GitHub Repository](#)
- openSAP course [SAP Cloud Platform Version Control with Git](#)

Coding Samples

Any software coding or code lines/strings ("Code") provided in this documentation are only examples and are not intended for use in a production system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules for certain SAP coding. SAP does not warrant the correctness or completeness of the Code provided herein and SAP shall not be liable for errors or damages caused by use of the Code, except where such damages were caused by SAP with intent or with gross negligence.

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.