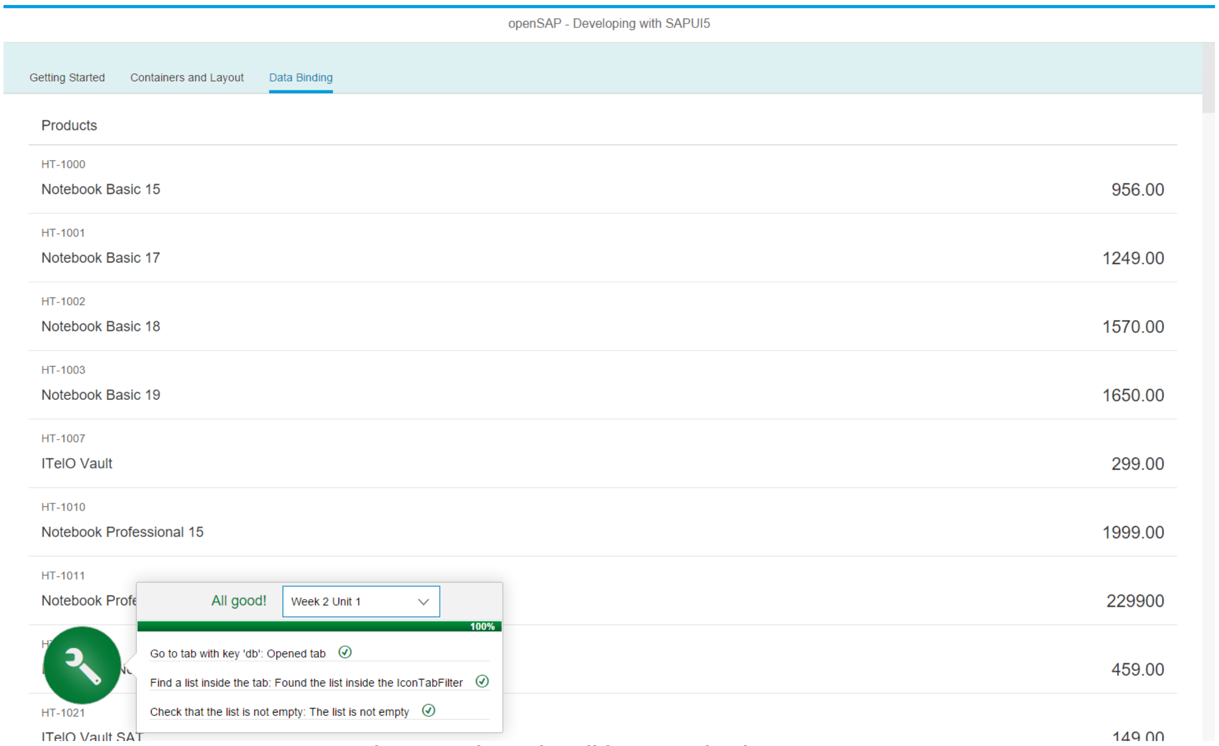
WEEK 0 UNIT 5
A GLANCE AT THE CODING EXERCISES

Please use these instructions to perform the exercises in the course as shown in the video.

Table of Contents

Preview


openSAP - Developing with SAPUI5

Products

HT-1000 Notebook Basic 15	956.00
HT-1001 Notebook Basic 17	1249.00
HT-1002 Notebook Basic 18	1570.00
HT-1003 Notebook Basic 19	1650.00
HT-1007 ITelO Vault	299.00
HT-1010 Notebook Professional 15	1999.00
HT-1011 Notebook Profe	229900
	459.00
HT-1021 ITelO Vault SAT	149.00

All good! Week 2 Unit 1

100%

Go to tab with key 'db': Opened tab ✓

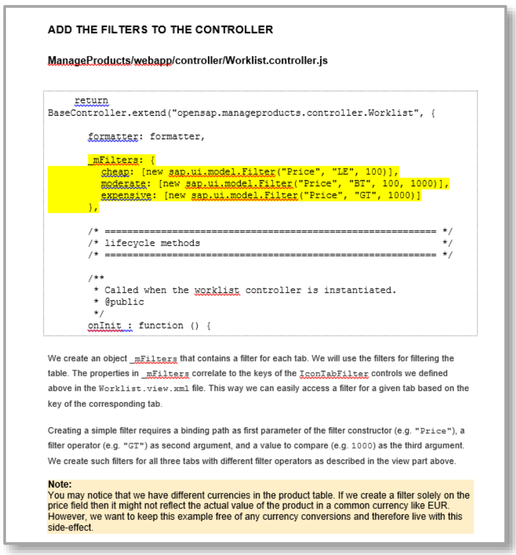
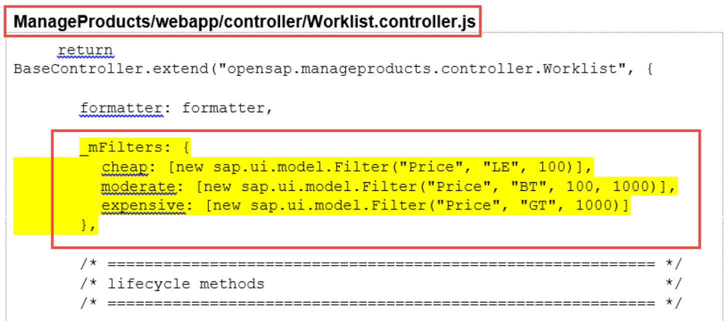
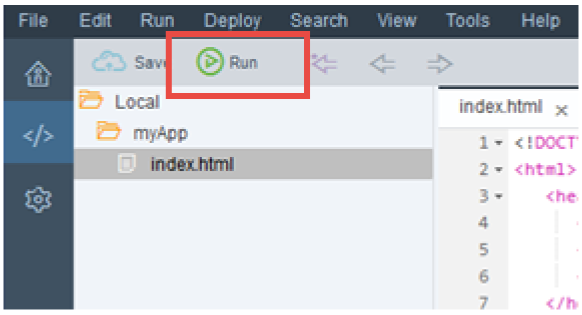
Find a list inside the tab: Found the list inside the IconTabFilter ✓

Check that the list is not empty: The list is not empty ✓

Figure 1 - The code validator running in your app

Using the coding exercise handouts

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

Explanation	Screenshot
1. Open the exercise document after you have watched the unit video. Follow the steps described in the document by applying the changes marked in yellow to your app project in SAP Web IDE. **Note: Exercise types** The exercises for a unit can be either configuration steps, code changes, or a mixture of both	
2. Code changes follow the following pattern: • File to be changed • Changes marked in yellow • Explanation of the changes below	
3. Check the result of your changes frequently by running the app. **Note:** Make sure that your changed files are saved. Unsaved changes are marked with an asterisk (*) behind the filename.	

ADD THE FILTERS TO THE CONTROLLER

ManageProducts/webapp/controller/Worklist.controller.js

```
    return
BaseController.extend("opensap.manageproducts.controller.Worklist", {

        formatter: formatter,

        _mFilters: {
            cheap: [new sap.ui.model.Filter("Price", "LE", 100)],
            moderate: [new sap.ui.model.Filter("Price", "BT", 100, 1000)],
            expensive: [new sap.ui.model.Filter("Price", "GT", 1000)]
        },

        /* =========================================================== */
        /* lifecycle methods                                           */
        /* =========================================================== */

        /**
         * Called when the worklist controller is instantiated.
         * @public
         */
        onInit : function () {
```

We create an object `_mFilters` that contains a filter for each tab. We will use the filters for filtering the table. The properties in `_mFilters` correlate to the keys of the `IconTabFilter` controls we defined above in the `Worklist.view.xml` file. This way we can easily access a filter for a given tab based on the key of the corresponding tab.

Creating a simple filter requires a binding path as first parameter of the filter constructor (e.g. `"Price"`), a filter operator (e.g. `"GT"`) as second argument, and a value to compare (e.g. `1000`) as the third argument. We create such filters for all three tabs with different filter operators as described in the view part above.

Note:
You may notice that we have different currencies in the product table. If we create a filter solely on the price field then it might not reflect the actual value of the product in a common currency like EUR. However, we want to keep this example free of any currency conversions and therefore live with this side-effect.

ManageProducts/webapp/controller/Worklist.controller.js

```
    return
BaseController.extend("opensap.manageproducts.controller.Worklist", {

        formatter: formatter,

        _mFilters: {
            cheap: [new sap.ui.model.Filter("Price", "LE", 100)],
            moderate: [new sap.ui.model.Filter("Price", "BT", 100, 1000)],
            expensive: [new sap.ui.model.Filter("Price", "GT", 1000)]
        },

        /* =========================================================== */
        /* lifecycle methods                                           */
        /* =========================================================== */
```

```
<!DOCT
<html>
  <he

  </h
```

1 ADD THE VALIDATOR SCRIPT TO YOUR APP PROJECT

This script will check the course exercises and only needs to be set up once for each app project.

webapp/index.html

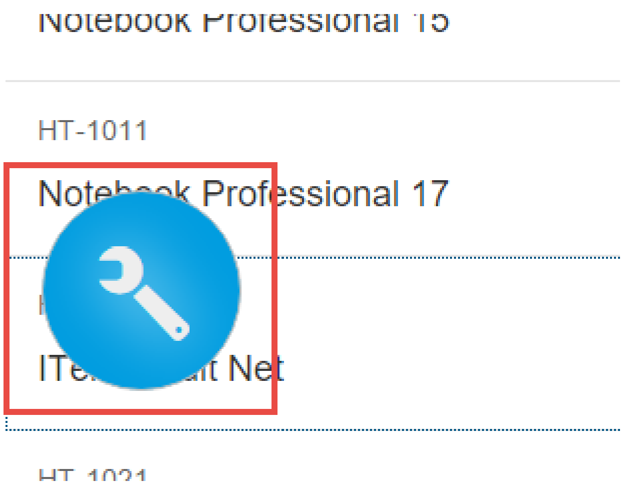
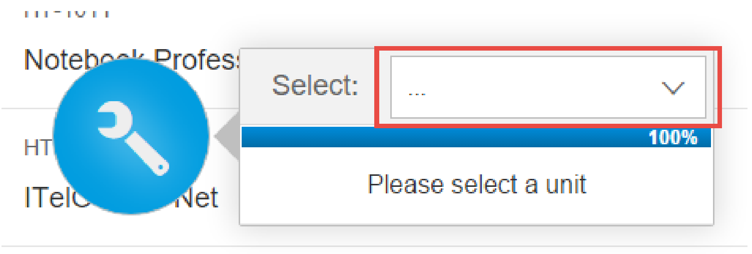
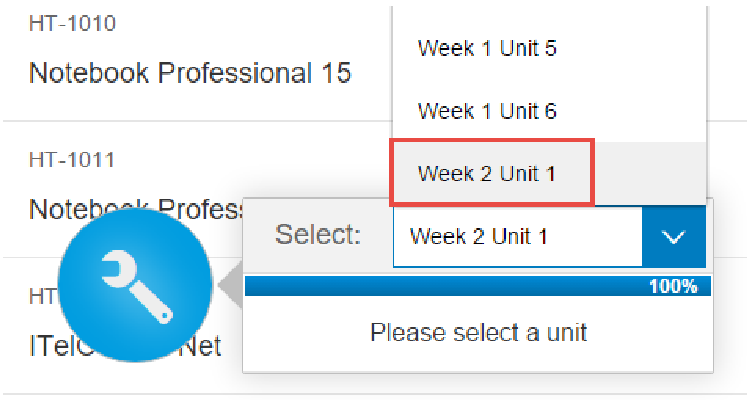
```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta charset="utf-8">
  <title>openSAP - Developing Web Apps with SAPUI5</title>
  <script id="sap-ui-bootstrap" src="…resources/sap-ui-core.js" …>
  </script>
  <script src="https://sap.github.io/openSAP-ui5-
course/Validator.js"></script>
  …
</head>
<body class="sapUiBody" id="content">
</body>
</html>
```

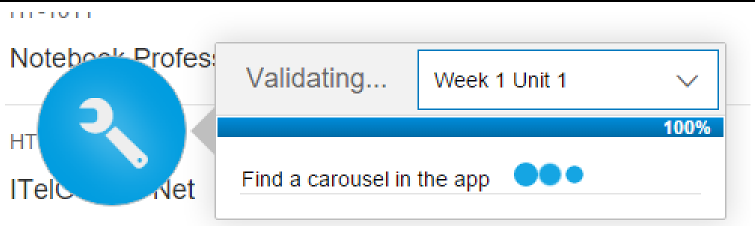
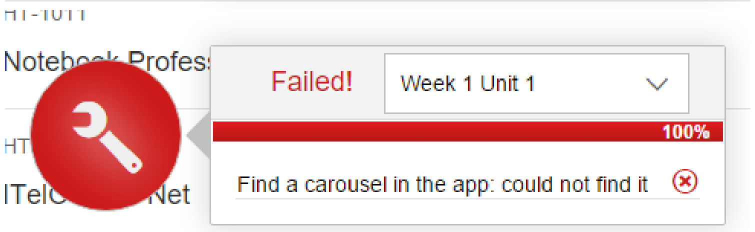
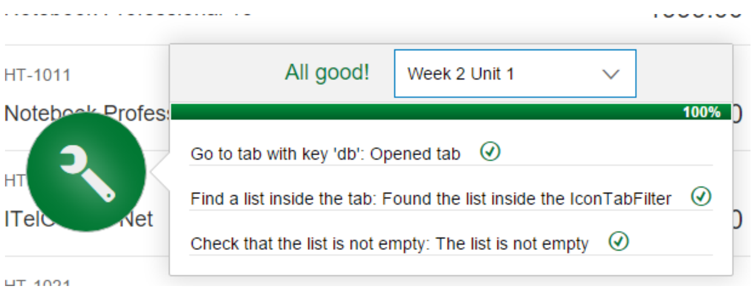
The validator script for this course is hosted on a public GitHub repository. Just add the script tag after the SAPUI5 boostrap tag. It injects a button that can run tests against your application coding to verify the functionality that you have added.

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.

2 VALIDATE YOUR EXERCISE CODE

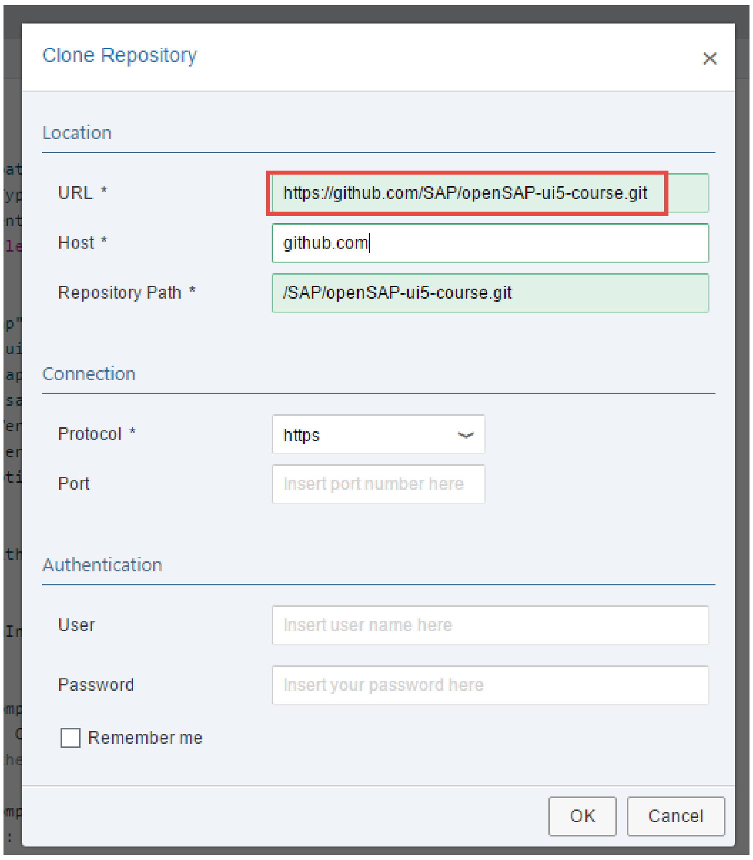
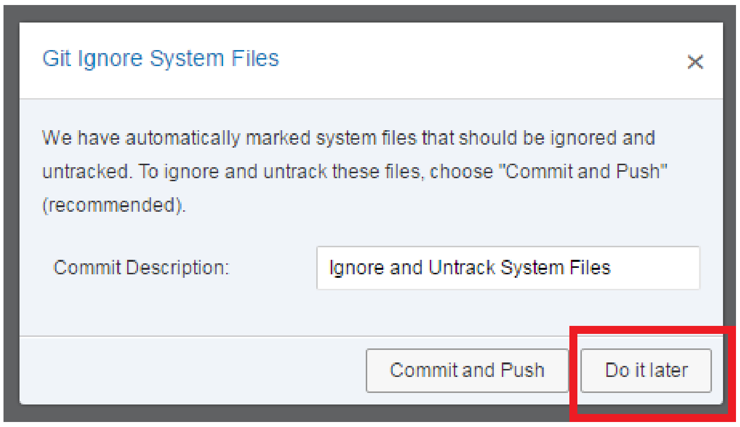
This script will check the exercises starting Week 1 Unit 1 to Week 4 Unit 5 and the bonus exercises. Validating your code is optional but may be helpful to check if your exercise code is correct.

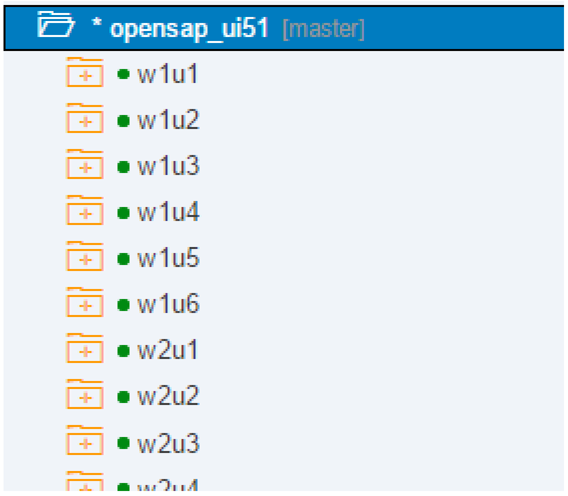
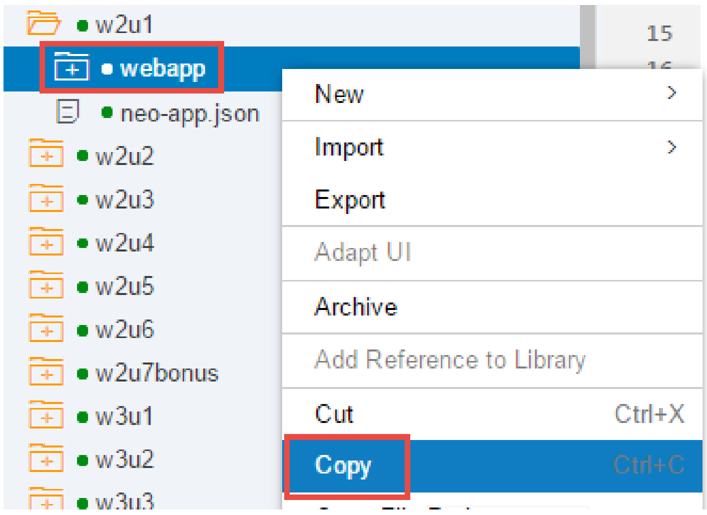
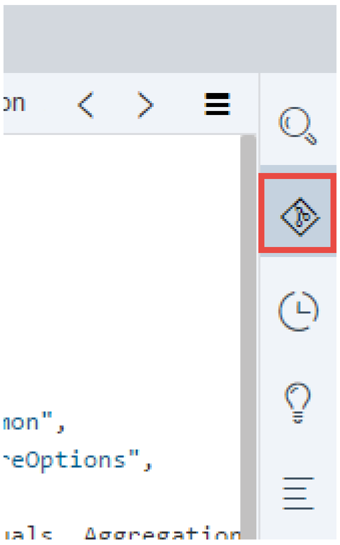
Explanation	Screenshot
1. After adding the script tag to your "index.html" file, run your app with one of the run configurations. **You should see a blue button with a wrench icon in the lower left area of the screen.** **Note: Validator issues** As the URL and functionality of the validator may change at short notice during the course, be sure to check the forum announcements for additional details.	 Notebook Professional 15 HT-1011 Notebook Professional 17 ITelO Vault Net HT-1021
2. Click the "Validate" button or press F9 to open the validator menu. 3. Click the Select menu to open the unit selection. **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.	 Notebook Profes Select: ... 100% Please select a unit ITelO Net
4. Choose the unit that you are currently working on and you want to validate. 5. The tests will automatically start to run and the progress is updated after each test is executed. **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.	 HT-1010 Notebook Professional 15 HT-1011 Notebook Profes Week 1 Unit 5 Week 1 Unit 6 Week 2 Unit 1 Select: Week 2 Unit 1 100% Please select a unit ITelO Net

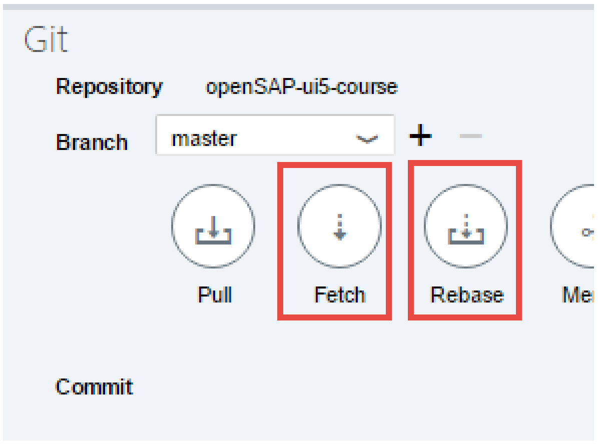
Explanation	Screenshot
6. If there is a problem with your exercise coding, you will get an error message in the result window of the validator. 7. Check the error message, fix the code that causes the issue, and run the validator again. 8. You can trigger another test run by clicking the Validate button again if you think that there was a problem with the test execution. **Note: Test logic** The tests will check that certain conditions are fulfilled in your app. They rely on specific controls being used, nesting orders, aggregations being filled, or IDs set to a value that is specified in the exercise documents. You need to stick closely to the solution exercises, otherwise the validation will probably not work.	 Validating... Week 1 Unit 1 100% Find a carousel in the app  Failed! Week 1 Unit 1 100% Find a carousel in the app: could not find it
9. Once all tests have run successfully, the validator will display a success message. **You have successfully validated your exercises.**	 All good! Week 2 Unit 1 100% Go to tab with key 'db': Opened tab ✓ Find a list inside the tab: Found the list inside the IconTabFilter ✓ Check that the list is not empty: The list is not empty ✓

3 LOAD AN EXERCISE SOLUTION

This section explains how to load an exercise solution for one of the units from the code repository

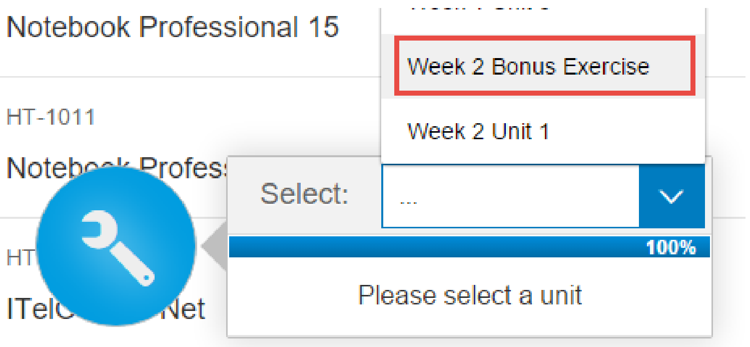
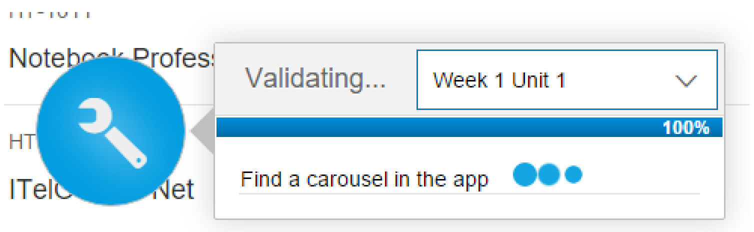
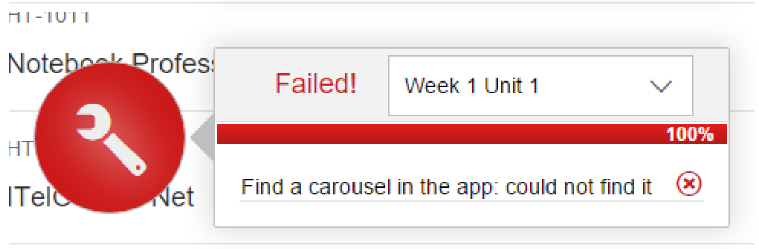
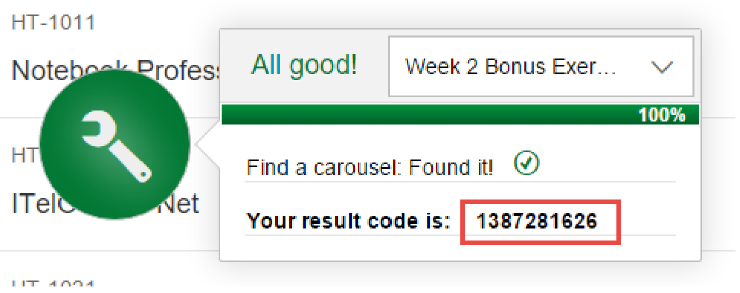
Explanation	Screenshot
1. In SAP Web IDE, go to **File > git > Clone Repository** and simply enter the following URL in the configuration dialog: https://github.com/SAP/openSAP-ui5-course.git **Note: Authentication** Leave all other fields of the dialog empty. No authentication is needed for this Git repository.	 Clone Repository Location Connection Authentication OK Cancel URL * https://github.com/SAP/openSAP-ui5-course.git Host * github.com Repository Path * /SAP/openSAP-ui5-course.git Protocol * https Port Insert port number here User Insert user name here Password Insert your password here Remember me
2. If you encounter this dialog press the “Do it later” button	 We have automatically marked system files that should be ignored and untracked. To ignore and untrack these files, choose "Commit and Push" (recommended). Commit Description: Ignore and Untrack System Files

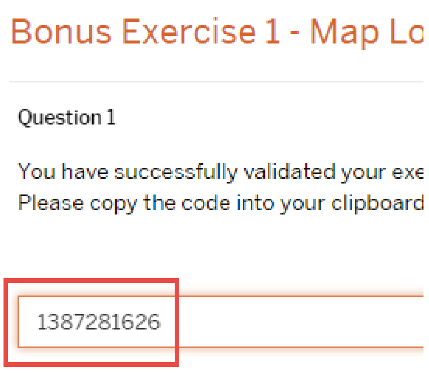
Explanation	Screenshot
3. The Git project with the solution code and the sample mock data is added to your workspace.	 * opensap_ui51 [master] w1u1 w1u2 w1u3 w1u4 w1u5 w1u6 w2u1 w2u2 w2u3
4. Copy the contents of one of the folders (e.g. w2u1 is week 2 unit 1) over to your app project to reset your workspace. **Note: Copying folders** Right-click and select “copy” on the solution folder. Then right-click your project root folder and select “paste”. **For technical reasons, it is not possible to override folders in SAP Web IDE; you have to delete or rename the old contents in the project folder first.**	 w2u1 webapp neo-app.json w2u2 w2u3 w2u4 w2u5 w2u6 w2u7bonus w3u1 w3u2 w3u3 15 New Import Export Adapt UI Archive Add Reference to Library Cut Ctrl+X Copy Ctrl+C
5. We will provide the solutions on a weekly basis. Therefore, you will have to fetch and rebase if the solution for your unit is not yet in your workspace. To do so, select the folder “opensap_ui51” in your workspace and switch to the Git pane on the right side.	

Explanation	Screenshot
6. Click the “Fetch” button, wait for the success message. Then click the “Rebase” button to update your project. 7. If there are new changes in the Git repository, new files and folders will now be added to your workspace.	 Git Repository openSAP-ui5-course Branch master Commit

4 VALIDATE YOUR BONUS EXERCISE

This script will check the bonus exercises that are offered in week 2 and 4 of the course. Validating your code is required to check your exercise code and to score the bonus points.

Explanation	Screenshot
1. Run the validator as described in the previous steps, but this time select the bonus exercise from the selection menu. **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.	 Notebook Professional 15 HT-1011 Week 2 Bonus Exercise Week 2 Unit 1 Select: ... 100% Please select a unit
2. If there is a problem with your exercise coding, you will get an error message in the result window of the validator. 3. Check the error message, fix the code that causes the issue, and run the validator again. **Note: Test logic** The tests will check that certain conditions are fulfilled in your app. They rely on specific controls being used, nesting orders, aggregations being filled, or IDs set to a value that is specified in the exercise documents. You need to stick closely to the task of the exercise, otherwise the validation will probably not work.	 Find a carousel in the app Find a carousel in the app: could not find it
4. Once all tests have run successfully, the validator will display a result code. 5. You have successfully validated your exercises. **Copy the code into your clipboard!**	 Find a carousel: Found it! **Your result code is:** 1387281626

Explanation	Screenshot
6. Start the quiz for the bonus exercise. There will only be one question with an input field for your answer. 7. Paste the code in the corresponding field of the openSAP exercise platform and press submit to score the extra points.	 Bonus Exercise 1 - Map Lo Question 1 You have successfully validated your exe Please copy the code into your clipboard 1387281626