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

[1 Add the validator script to your app project 3](#_Toc449020578)

[2 Validate your exercise code 4](#_Toc449020579)

[3 Load an exercise solution 6](#_Toc449020580)

[4 Validate your bonus exercise 8](#_Toc449020581)

## Preview

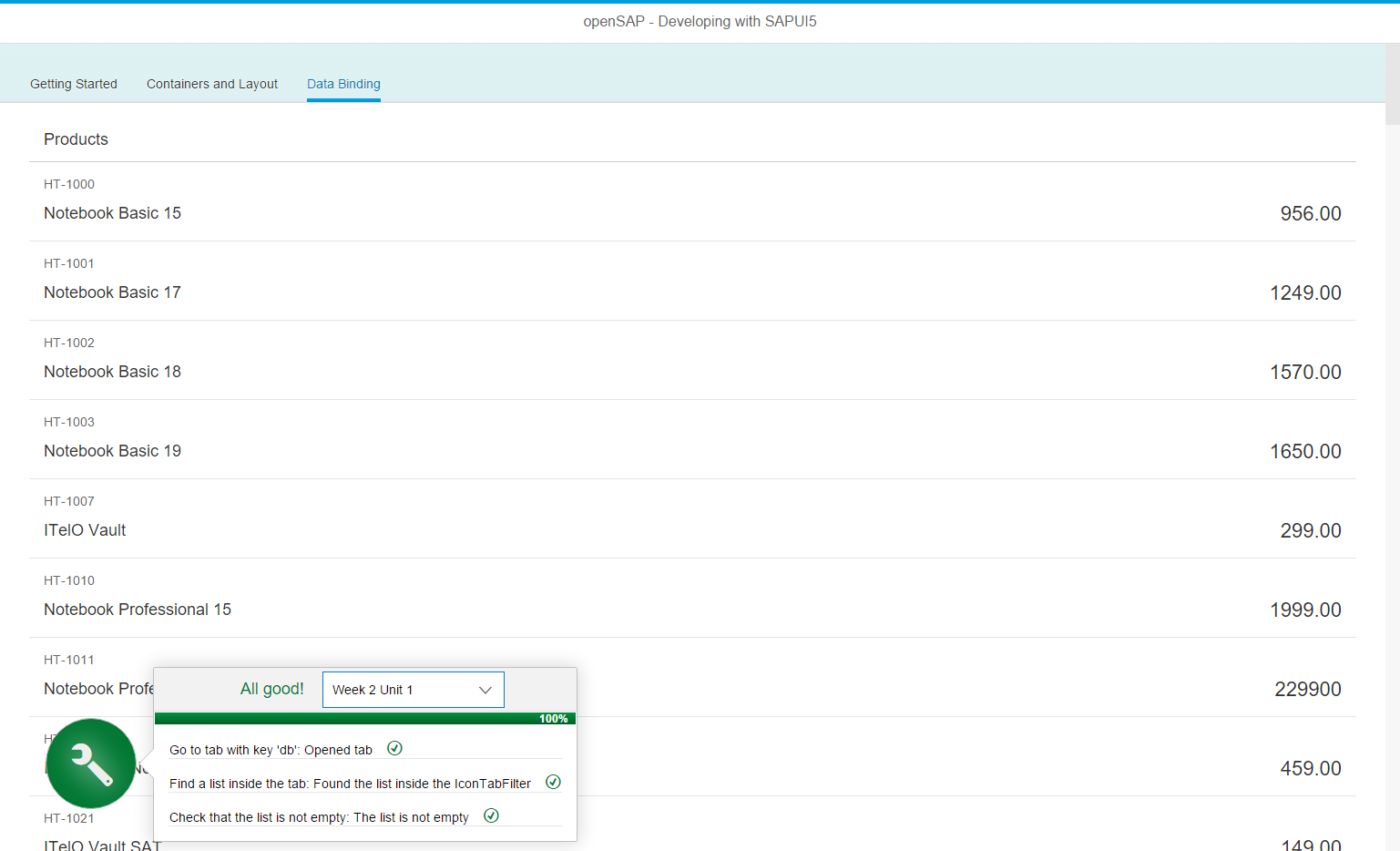
[](file:///C:\Users\D050079\AppData\Roaming\Microsoft\Word\The%20code%20validator%20running%20in%20your%20app)

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 |  |
| 1. Code changes follow the following pattern:     * File to be changed    * Changes marked in yellow    * Explanation of the changes below |  |
| 1. 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 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.

# 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. |  |
| 1. Click the “Validate” button or press F9 to open the validator menu. 2. 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. |  |
| 1. Choose the unit that you are currently working on and you want to validate. 2. 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. |  |
| 1. If there is a problem with your exercise coding, you will get an error message in the result window of the validator. 2. Check the error message, fix the code that causes the issue, and run the validator again. 3. 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. |  |
| 1. Once all tests have run successfully, the validator will display a success message.   **You have successfully validated your exercises.** |  |

# 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**](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. |  |
| 1. If you encounter this dialog press the “Do it later” button |  |
| 1. The Git project with the solution code and the sample mock data is added to your workspace. |  |
| 1. 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.** |  |
| 1. 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. |  |
| 1. Click the “Fetch” button, wait for the success message. Then click the “Rebase” button to update your project. 2. If there are new changes in the Git repository, new files and folders will now be added to your workspace. |  |

# 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. |  |
| 1. If there is a problem with your exercise coding, you will get an error message in the result window of the validator. 2. 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. |  |
| 1. Once all tests have run successfully, the validator will display a result code. 2. You have successfully validated your exercises. **Copy the code into your clipboard!** |  |
| 1. Start the quiz for the bonus exercise. There will only be one question with an input field for your answer. 2. Paste the code in the corresponding field of the openSAP exercise platform and press submit to score the extra points. |  |

**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 productive 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 damage caused by use of the Code, except where such damage was caused by SAP with intent or through gross negligence.