**VS Code Labs**

**Lab covers:**

* Use Case 1 – **Code Explanation**
* Use Case 2 – **Upgrade Java**

**Environment Setup**

1. Download the latest WCA4EJ Code VS Code Extension and install by following these instructions: [Wca4ej Extension](https://ibm.box.com/s/o26ggaar57eh61m2t4ndvx84s31zho3v)

* If using Windows OS, the Windows plugin requires the [Microsoft Edge WebView2 runtime](https://developer.microsoft.com/en-us/microsoft-edge/webview2/?form=MA13LH).
* Scroll down and choose Evergreen Standalone Installer.
* In newer versions of Windows this might already be installed. In that case, you will receive this message while installing.

1. To install the extension in your VS code, go to extensions, and click on 3 dots to select the vsix file downloaded in previous step

A screenshot of a computer

Description automatically generated

1. Next, we will install liberty tools from vs code marketplace. In the VS code marketplace, search liberty, and install the following extension (Liberty Tools)

A screenshot of a computer

Description automatically generated

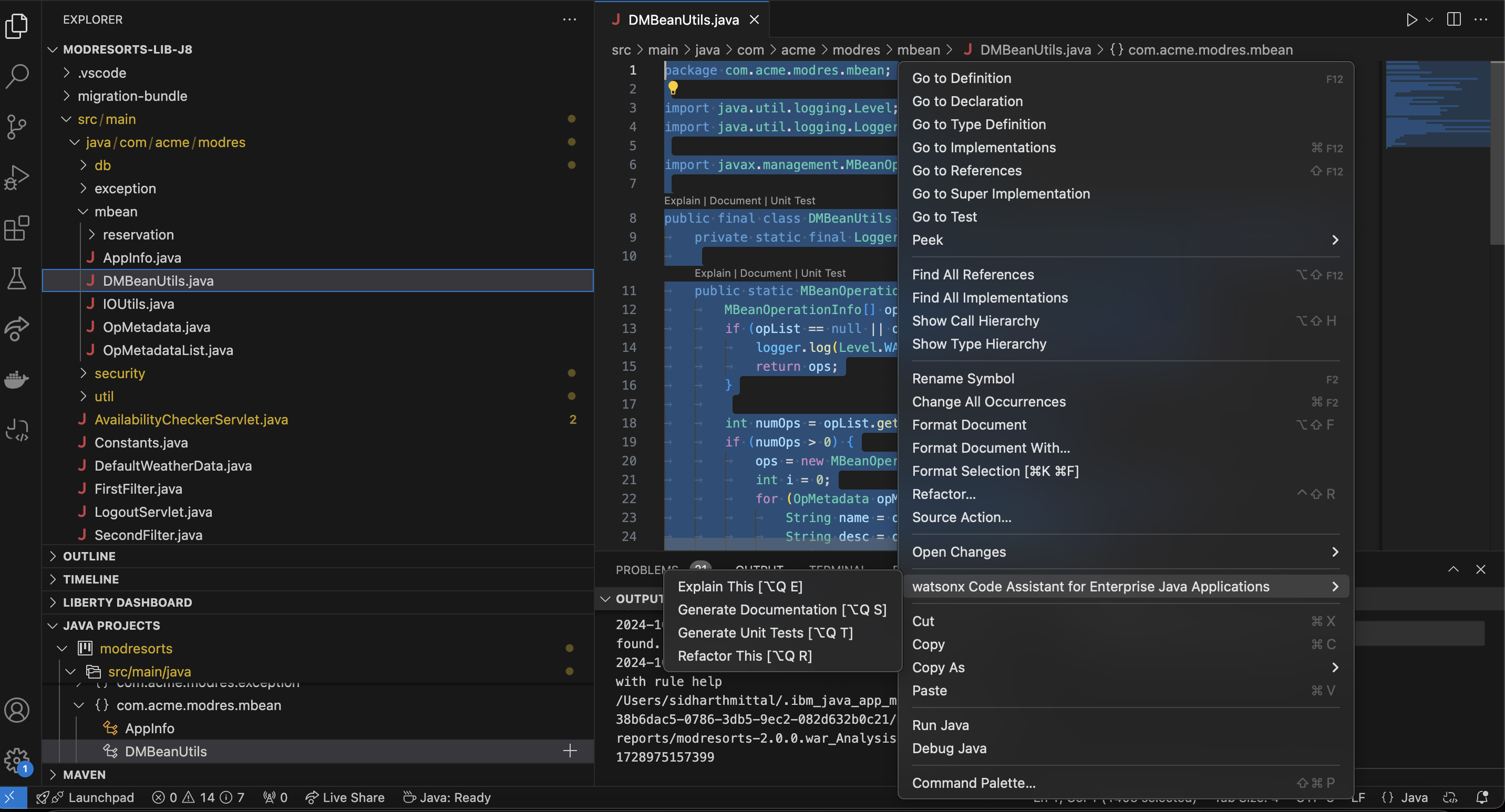
**Build Project**

Git clone the project from the [GitHub repository](https://github.com/sidharthmittal25/wca4ej-workshop/tree/main)

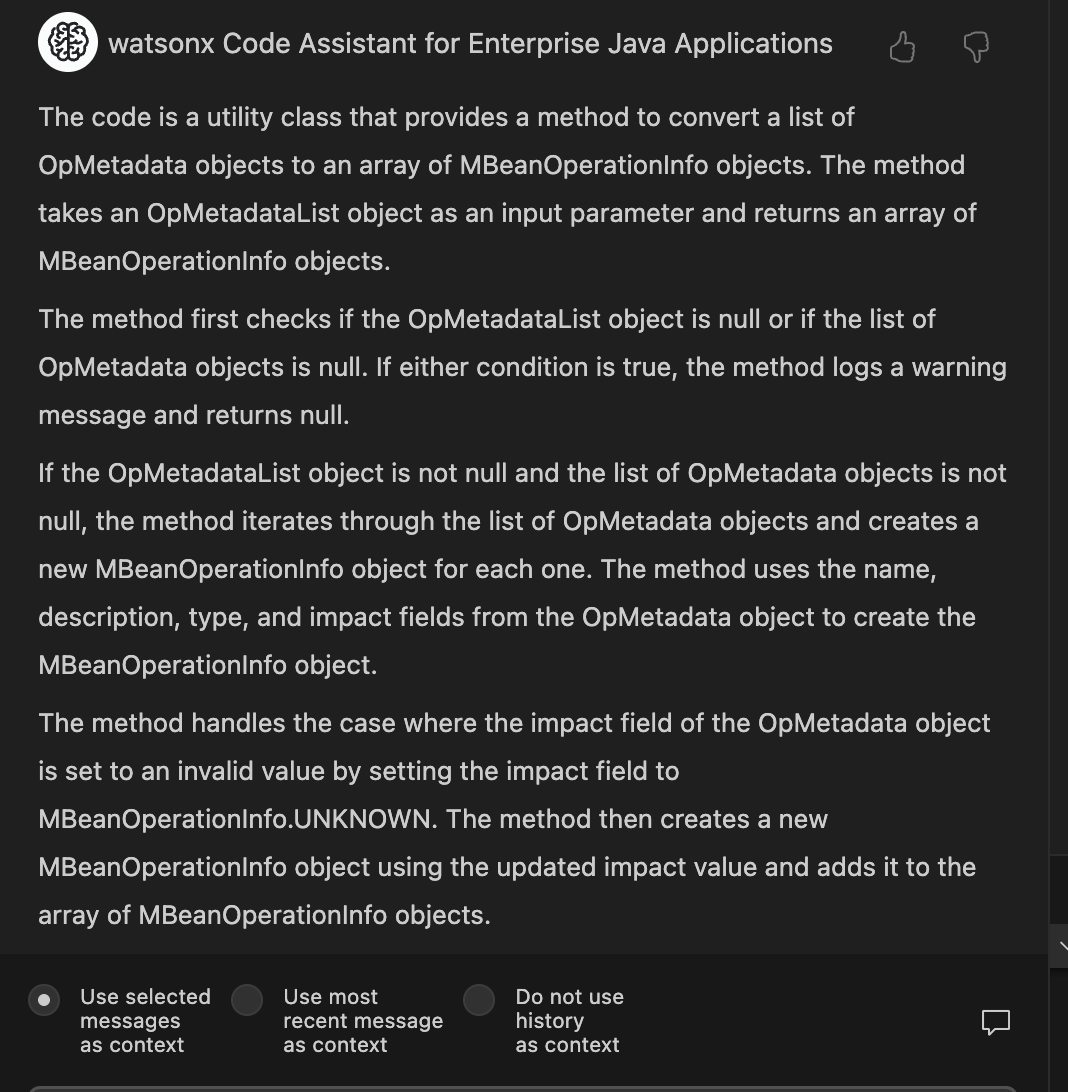
1. Open a terminal, and go to the project folder, and navigate to the folder: vs-code-labs/modresorts-liberty-j8 (in this case we are using a java 8 liberty application)
2. Build using: ***maven clean install***

**Use Case 1** **Code Explanation**

1. Open the project (vs-code-labs/modresorts-liberty-j8) into VS Code:
2. Once the project is open, we can look at the different java files, and try to select certain part of the java to get explanation. After selecting the code, right click, and ***select watsonx Code Assistant for Enterprise Java Applications***, and select ***explain this***



You will see the following code explanation, make sure you select the option = ‘***use selected messages as context***’ in the chat interface.



Feel free to select any other part of the code to get some explanations.

**Use Case 2** **Upgrade Java [From Java 8 to Java 21]**

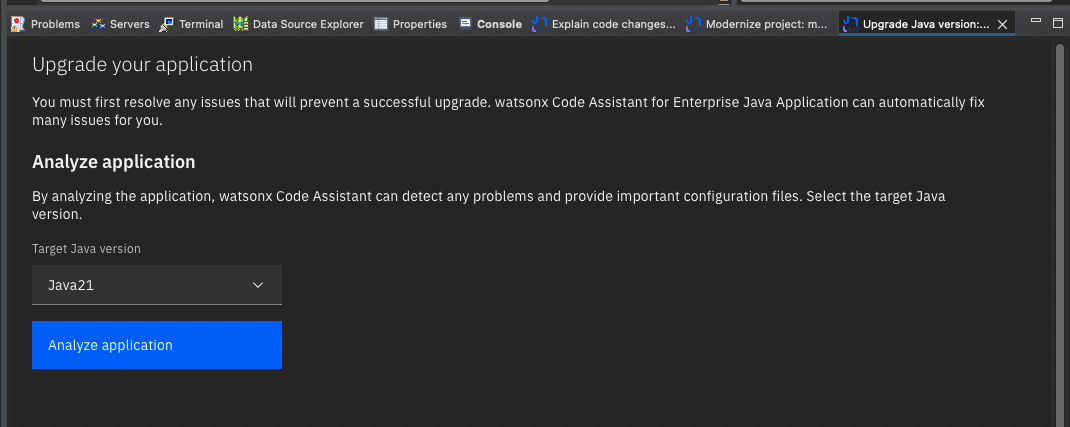
After getting the scenario, we will upgrade the java version for the liberty application.

1. Right click on src/main, select **watsonx Code Assistant for Enterprise Java Applications,** then select **Upgrade Java version**.

A screenshot of a computer

Description automatically generated

You should see the following panel appear:  
(Note: Click Start over in the newly opened tab to see this panel)



1. Before you Analyze the application, the application needs to be built. If it’s not built at this point build it now (command line: **mvn clean package**, or via the IDE UI if you have that configured). See also the previous note on Java versions in the building section. *(Note: Only perform this step if the application is not already ‘build and refresh’ in the previous lab)*
2. Ensure **Java21** is selected and Click the **Analyze application** button. After a few moments you should see the issues. There is one issue with an auto fix, and one issue with a LLM assisted fix:

A screenshot of a computer

Description automatically generated

1. At this point we can run ModResorts in a Java21 environment to observe how the application is not functioning correctly **BEFORE** we fix the Java upgrade issues.
2. Start the application using Liberty tools by navigating to the LIBERTY DASHBOARD view in the side panel.

A screenshot of a computer

Description automatically generated

* When the application is running, go to:   
  <http://localhost:9080/resorts>  
  and click the Where to drop-down. Select any value. You should observer ERRORs in the UI:

A screenshot of a computer screen

Description automatically generated

* These errors are ultimately the result of the Java upgrade issue with the MBeanOperatorInfo constructor which causes a server error when fetching data for the UI.

1. Back in VSCode, in Upgrade Java tab. Click **Build and Refresh** then the **Run auto-fixes** button. When you click auto-fixes, the change in java version is detected by VS Code, and you will see following pop-up. **Click Yes.**



1. Now we will resolve the other issue (**Behavior change in the** **javax.management.MBeanOperationInfo constructor)** with the help of the watsonx code assistant.
2. Due to limitation in private preview release, it is not easy to know which piece of code the issue relates to. To see this, you need to click on the **View analysis report** link. In the report, find the issue and click on **Show result**s to reveal the file, method and line for the issue.

A screenshot of a web page

Description automatically generated

1. Open the **DMBeanUtils** class in the IDE, as shown in the above path.
2. Select the **getOps** method beginning with the “public static” and finishing with the enclosing “}” and click **Help me** button:  
   *(code snippet as shown below)*



1. The code snippet should consist of the full **getOps** method, with the offending lines fixed. Replace your **getOps** method with the code returned from the LLM.  
   *(new code snippet as shown below)*

A computer screen shot of a program

Description automatically generated

1. After saving your work, go back to the **Upgrade Java** panel, and click **Build and refresh**. After a couple of moments, you should observe all the issues are fixed.

A screenshot of a computer

Description automatically generated

1. Return to <http://localhost:9080/resorts>

and click the **Where to** down. Select any value. You should observe that the errors have disappeared.

**ModResorts** is now a **Liberty application** and is upgraded to **Java21**!