

Lab3: Upgrade Java Version

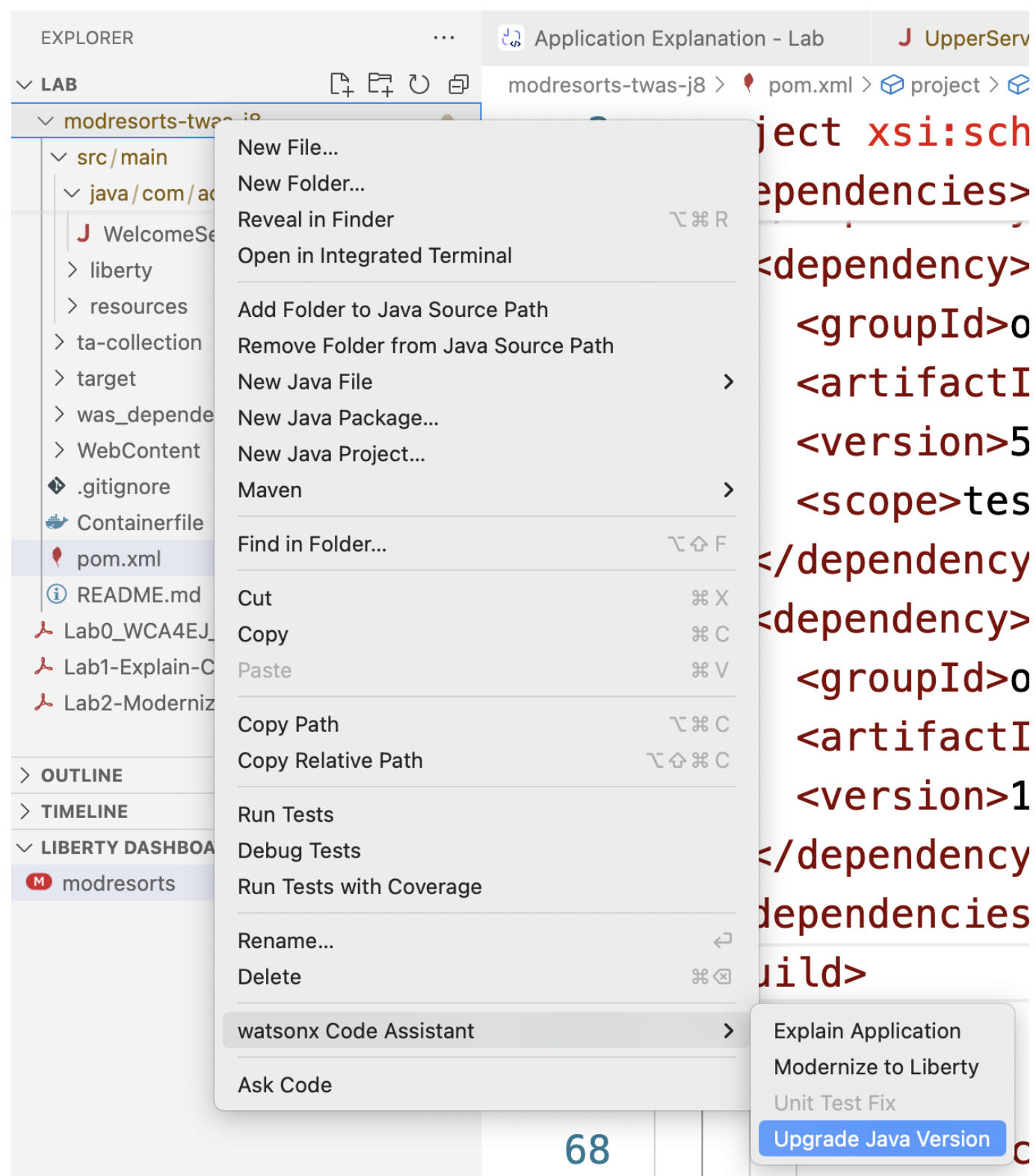
This document gives step-by-step guide to finish Lab3. You will find the steps very similar to Lab2.

Lab3 covers:

- Upgrade from Java8 to Java21

1. Application analysis

Right click on the `modresorts-twas-j8` folder and select `watsonx Code Assistant - Upgrade Java Version`.



A **Upgrade** tab will show up. In the **Target** dropdown, select **Java 21 (LTS)**, and then click **Build and analyze**.

Modernize

Upgrade X

☐ ...

Upgrade ModResorts

You must first resolve any issues that prevent a successful upgrade. Watsonx Code Assistant can automatically fix many issues for you.

Analyze application

Select the application analysis parameters. The analysis parameters define the rules that the application is evaluated against. Select the source runtime environment and target runtime environment for upgrading the Java version.

Source ⓘ
IBM Java 8 (Recommended) ▼
Detected in build configuration.

Target
Java 21 (LTS) ▼

❗ *The application must be built before analyzing, as the application analysis is performed on the application binary.*
*Choosing **Build and analyze** will automatically perform a build before the analysis assuming a default build command that also skips tests. For more advanced build configurations, you may wish to build the application manually, and then click **Analyze**.*

Analyze

Build and analyze

After analysis, the report will show you issues with this application to be fixed. 3 of them can be automated, 2 of them needs assistance. For the automated fixes, we click **Run automated fixes** (this process might take long).

Upgrade ModResorts

[↩ Return to analyze](#)

Application analysis parameters

Java version
IBM Java 8 to Java 21 (LTS)

[View technology report](#) | [View inventory report](#) | [View analysis report](#)

Issues 5

Generated on 13 November 2024 at 12:51

Issues are categorized by type of fixes available. Click the **Rebuild and refresh** button to update the list.

[Show more](#)

Rebuild and refresh

▼

Automated fixes	Assisted fixes	Self-directed fixes	Additional information
3	2	0	2

Instructions ▼

i

Java SE 21 general information and potential issues

▼

i

Java SE 17 general information and potential issues

▼

i

Java SE 11 general information and potential issues

▼

Run automated fixes

2. Apply Fixes

For the assisted fixes, you can click the **Instructions** dropdown to see the step which we will guide you. Now we expand the dropdowns of the first issue as shown. There are clear instructions on how to fix this issue at the bottom.

Modernize

Upgrade

Automated fixes3

Assisted fixes2

Self-directed fixes0

Additional information2

Instructions

Behavior change in the javax.management.MBeanOperationInfo constructor

In Java SE 11 the javax.management.MBeanOperationInfo constructor now throws an IllegalArgumentException when the impact provided is not one of INFO, ACTION, ACTION_INFO, or UNKNOWN.
For more information, see Java 10 Removed Features and Options.
[Read more](#)

src/main/java/com/acme/modres/mbean/DMBeanUtils.java

There is one occurrence of this issue in this file. You will have to resolve it sequentially.

Detected line number might differ from actual.

1. Method getOps

Match Criteria - javax.management.MBeanOperationInfo(java.lang.String, java.lang.String, javax.management.MBeanParameterInfo[], java.lang.String, int, javax.management.Descriptor)

Detected line number - 28

Step 1: Get a code suggestion

Click "Help Me" after you select the method that calls the MBeanOperationInfo constructor

Step 2: Check the impact field value

Confirm that MBeanOperationInfo.UNKNOWN is the appropriate impact value to be used in place of an invalid one

Back to top

Now we click the name of the file that has issue and needs to be fixed. In the editor window, we select the entire class defined in here that that calls the `MBeanOperationInfo` constructor by highlighting. Then we click `Help me` to ask help from LLM.

Application Explanation - Lab

UpperServlet.java 1

pom.xml

DMBeanUtils.java

modresorts-twas-j8 > src > main > java > com > acme > modres > mbean > DMBeanUtils.java > DMBeanUtils

5

6 import javax.management.MBeanOperationInfo;

7

8 public final class DMBeanUtils {

9 private static final Logger logger = Logger.getLogger(DMBeanUtils.class);

10

11 public static MBeanOperationInfo[] getOps(OpMetadataList opList) {

12 MBeanOperationInfo[] ops = null;

13 if (opList == null || opList.getOpMetadataList().isEmpty()) {

14 logger.log(Level.WARNING, "No operations found");

15 return ops;

16 }

17

18 int numOps = opList.getOpMetadataList().size();

19 if (numOps > 0) {

20 ops = new MBeanOperationInfo[numOps];

21 int i = 0;

22 for (OpMetadata opMetadata : opList.getOpMetadataList()) {

23 String name = opMetadata.getName();

24 String desc = opMetadata.getDescription();

25 String type = opMetadata.getType();

Automated fixes 3

Assisted fixes 2

Self-directed fixes 0

Additional info 2

Instructions

Behavior change in the javax.management.MBeanOperationInfo constructor

In Java SE 11 the `javax.management.MBeanOperationInfo` constructor now throws an `IllegalArgumentException` when the `impact` provided is not one of `INFO`, `ACTION`, `ACTION_INFO`, or `UNKNOWN`. For more information, see [Java 10 Removed Features and Options](#).

Read more →

src/main/java/com/acme/modres/mbean/DMBeanUtils.java

There is one occurrence of this issue in this file. You will have to resolve it sequentially.

Detected line number might differ from actual.

1. Method getOps

Match Criteria - `javax.management.MBeanOperationInfo(java.lang.String, java.lang.String, java.lang.String, javax.management.MBeanParameterInfo[], java.lang.String, int, javax.management.Descriptor)`

Detected line number - 28

Step 1: Get a code suggestion

Click "Help Me" after you select the method that calls the `MBeanOperationInfo` constructor

Step 2: Check the impact field value

Confirm that `MBeanOperationInfo.UNKNOWN` is the appropriate impact value to be used in place of an invalid one

Help me

Back to top

Avoid using `Thread.stop()`, `Thread.resume()`, and `Thread.suspend()`

The chat with model window will show up and ask model for help. The model gives suggestions for fixing the issue.

You

This code suggestion uses MBeanOperationInfo.UNKNOWN as the value for the impact field if an IllegalArgumentException is caught when instantiating an MBeanOperationInfo object

⋮



watsonx

```
public final class DMBeanUtils {
    private static final Logger logger = Logger.getLogger(DMBeanUtils.class.getName());

    public static MBeanOperationInfo[] getOps(OpMetadataList opList) {
        MBeanOperationInfo[] ops = null;
        if (opList == null || opList.getOpMetadataList() == null) {
            logger.log(Level.WARNING, "No operation is configured");
            return ops;
        }

        int numOps = opList.getOpMetadataList().size();
        if (numOps > 0) {
            ops = new MBeanOperationInfo[numOps];
            int i = 0;
            for (OpMetadata opMetadata : opList.getOpMetadataList()) {
                String name = opMetadata.getName();
                String desc = opMetadata.getDescription();
                String type = opMetadata.getType();
                int impact = opMetadata.getImpact();

                try {
                    MBeanOperationInfo opInfo = new MBeanOperationInfo(name, desc, /* signature */ null, type, impact);
                    ops[i++] = opInfo;
                } catch (IllegalArgumentException e) {
                    logger.log(Level.WARNING, "MBeanOperationInfo: Unknown impact. Operation: " + name + ". Using MBeanOperationInfo opInfo = new MBeanOperationInfo(name, desc, /* signature */ null, type, MBeanOperationInfo.UNKNOWN);
                    ops[i++] = opInfo;
                }
            }
        }
    }
}
```

Here we change the code in the `DMbeanUtils.java` file using the copy code from chat.

We can ignore the second issue for now as it is a deprecation warning. Or if the rebuild still have error, you can go through the same process and fix the warning as well.

3. Rebuild Application

Now that we have all the fixes applied, we can **Rebuild and refresh** to see that there are no more critical issues with the application.

Modernize

Upgrade

Upgrade ModResorts

Return to analyze

Application analysis parameters

Java version

IBM Java 8 to Java 21 (LTS)

View technology report

View inventory report

View analysis report

Issues

1

Generated on 13 November 2024 at 13:26

Issues are categorized by type of fixes available. Click the **Rebuild and refresh** button to update the list.

Show more

Rebuild and refresh

Automated fixes

Assisted fixes

Self-directed fixes

Additional information

0

1

0

2

Instructions

Avoid using Thread.stop(), Thread.resume(), and Thread.suspend() methods

This rule flags the use of the Thread.stop(), Thread.suspend(), and Thread.resume() methods. These methods have been terminally deprecated in Java SE 18 and have been updated to always throw an UnsupportedOperationException. The method will eventually be degraded and removed in a future release as it is inherently unsafe and has been deprecated since Java 1.2.

For more information see JDK-8277861.

Read more →

We can go to browser to view the new application. This time you will find that the location module is fixed.

