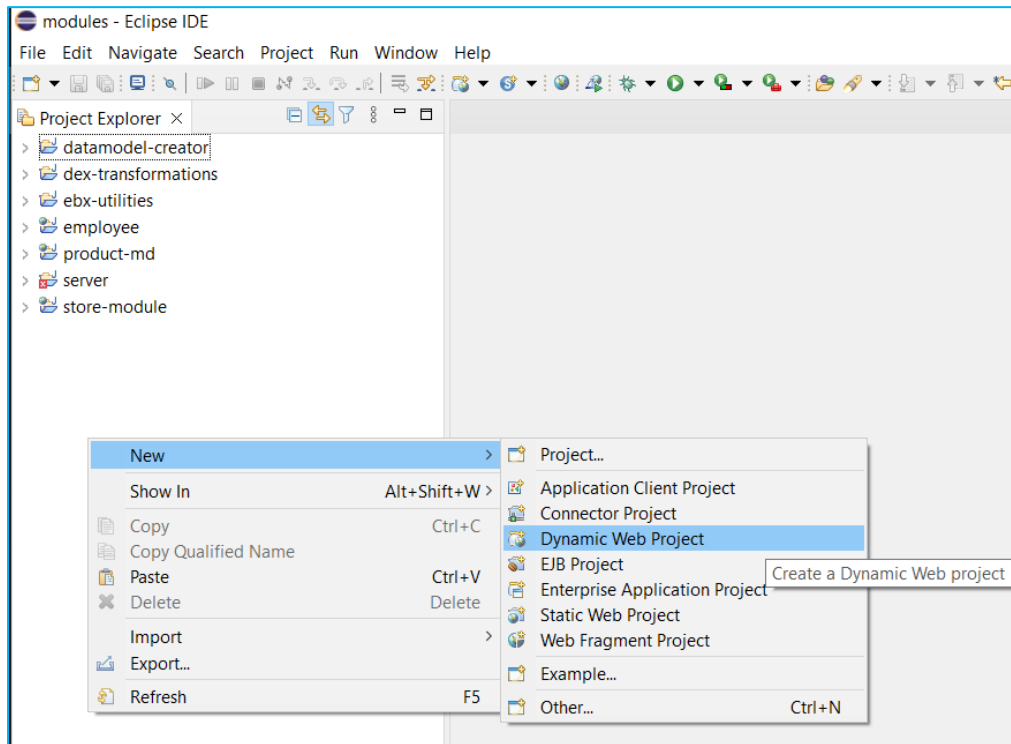


Module Creation in Eclipse

1. Right click in Project Explorer window.

New --> Dynamic Web Project



In New Dynamic Web Project enter the module name.

New Dynamic Web Project

Dynamic Web Project

Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name: data-import

Project location

☒ Use default location

Location: E:\EBX_Canvas\modules\data-import Browse...

Target runtime

<None> New Runtime...

Dynamic web module version

4.0

Configuration

Default Configuration Modify...

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR New Project...

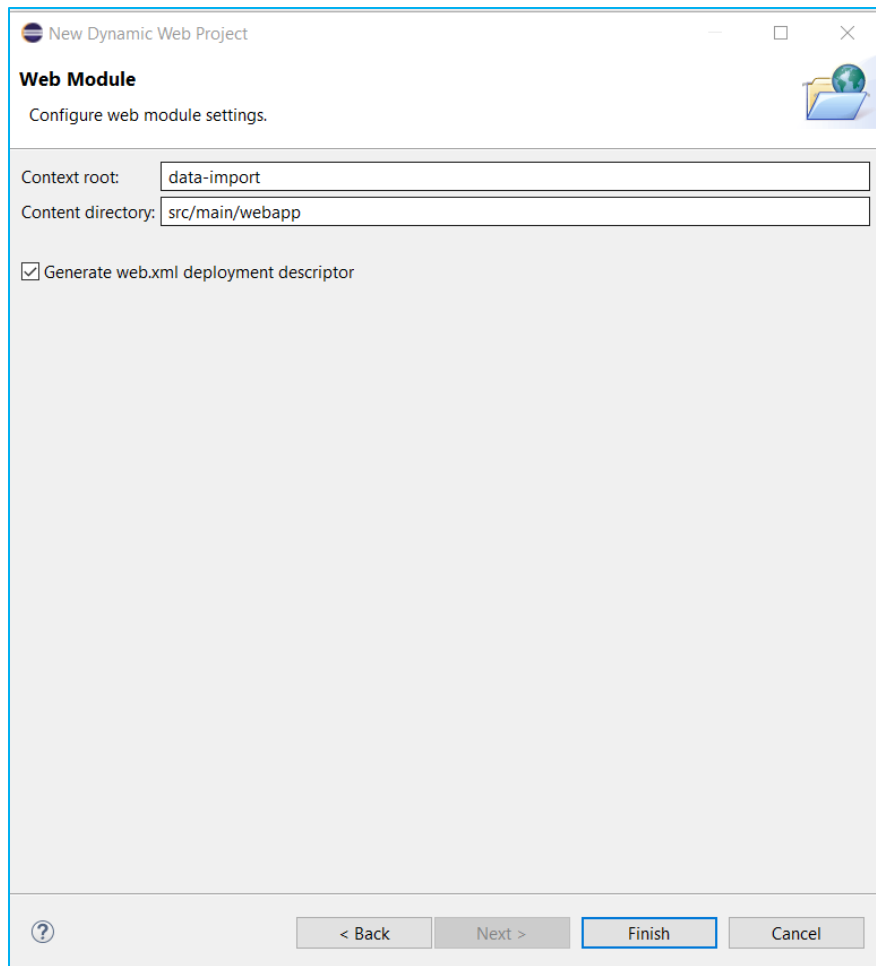
Working sets

☐ Add project to working sets New...

Working sets: Select...

? < Back Next > Finish Cancel

Click Next ----> Next on Web Module screen select the “Generate web.xml deployment description” checkbox.



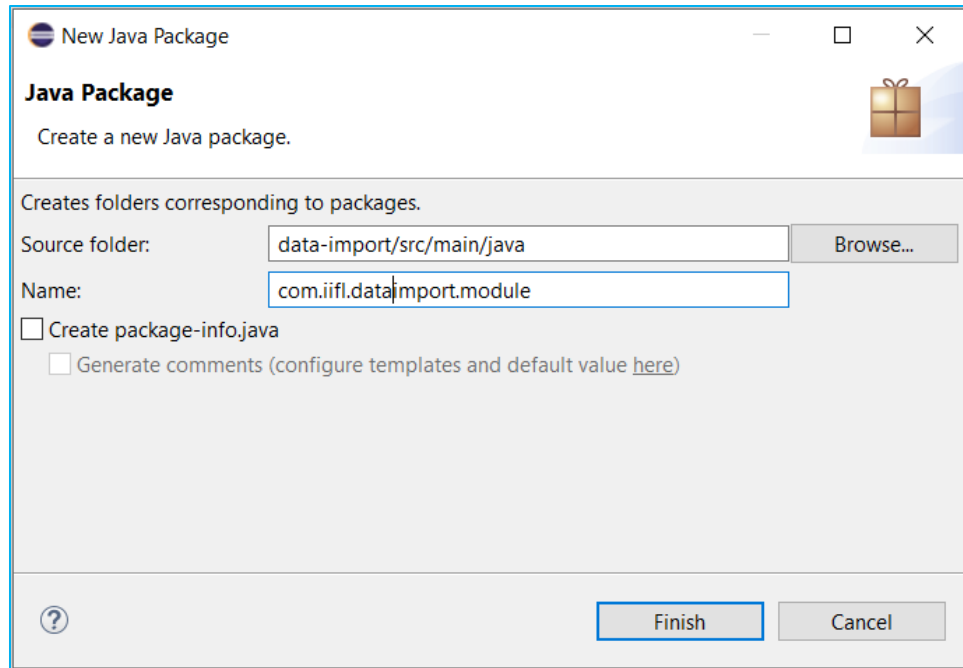
Click Finish. This will generate the module folder in the eclipse workspace.

Module Registration with EBX

Create appropriate packages in the newly created module.

1. Right click on the src/main/java. Then New --> Package.

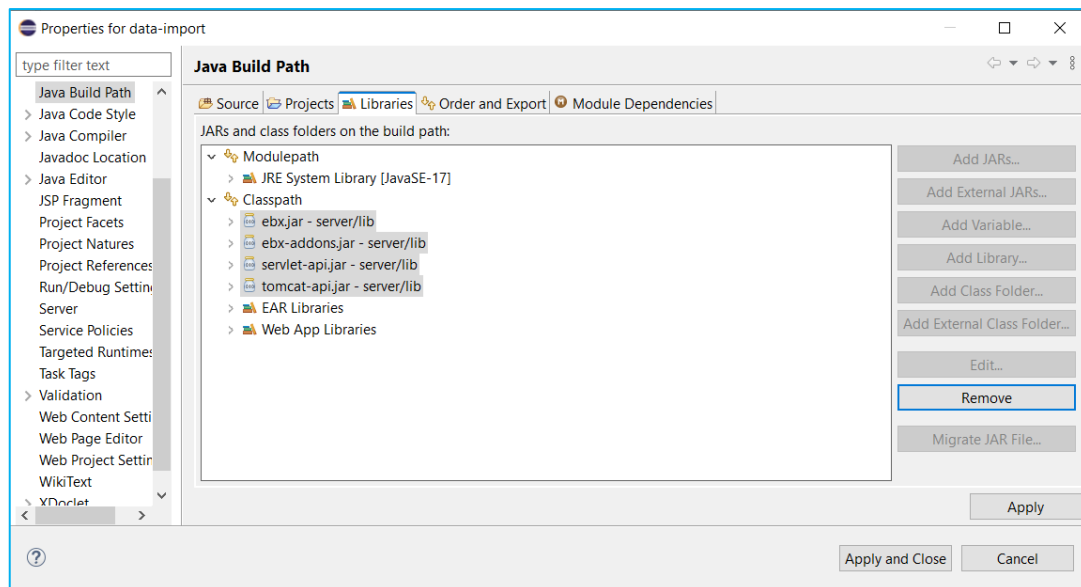
Provide the name of the module and click Finish.



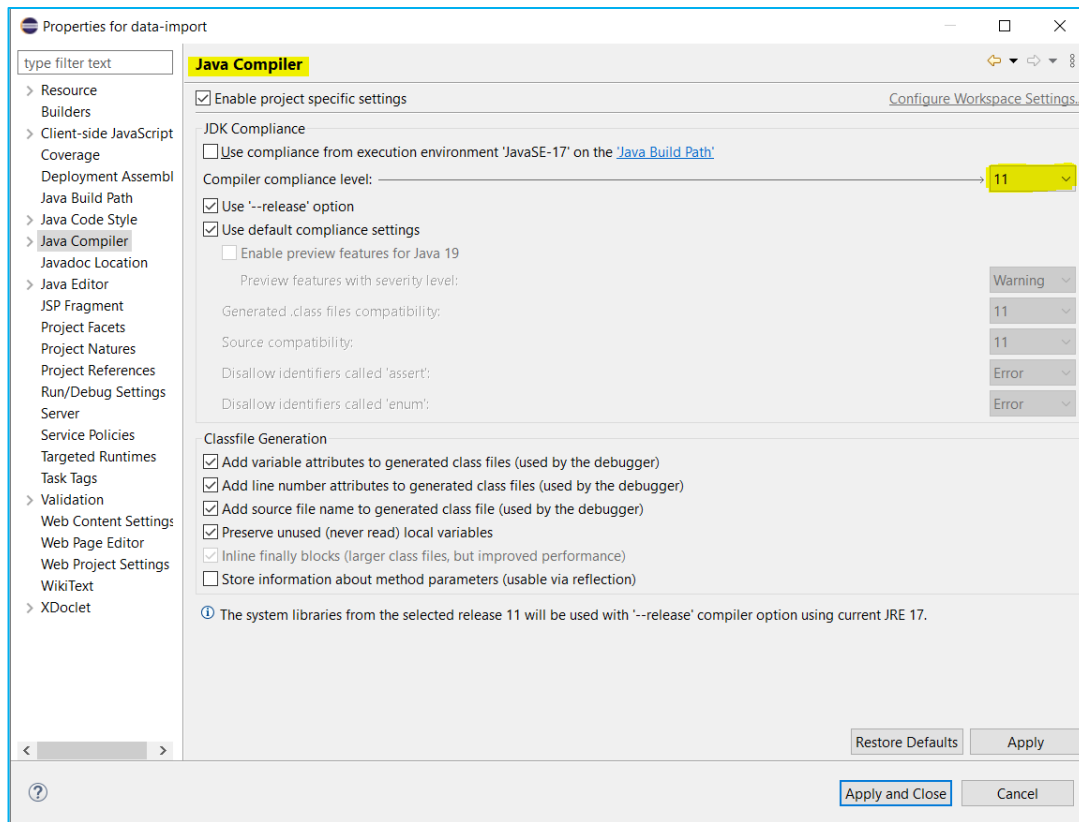
2. On the newly created package add the dependency jar in the module configuration.

Right click on the module folder. Build Path --> Configure Build Path.

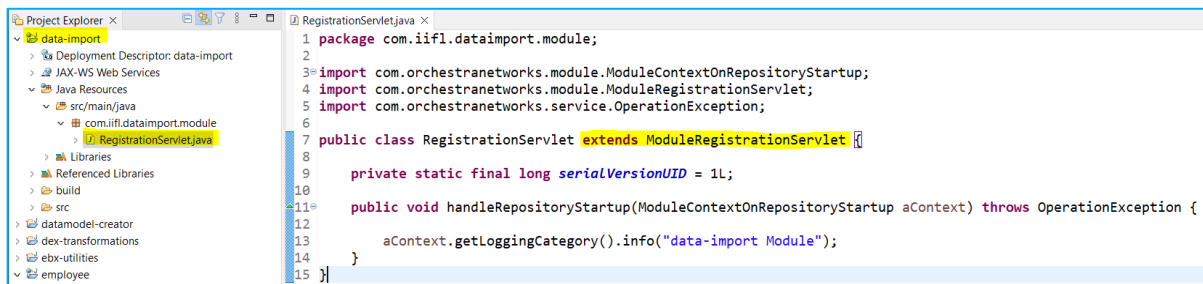
In the "Java Build Path" --> Libraries add the EBX jar required for using the EBX API.



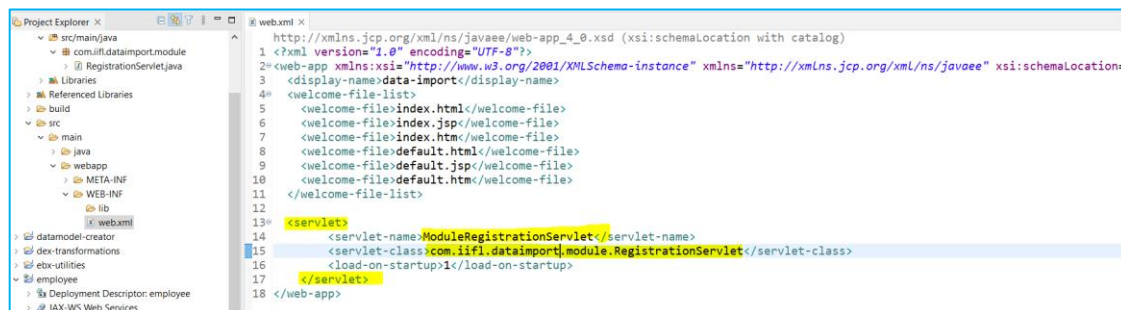
3. Also set the java compiler version to same as that of on which the EBX is running. (In this document it is Java 11)



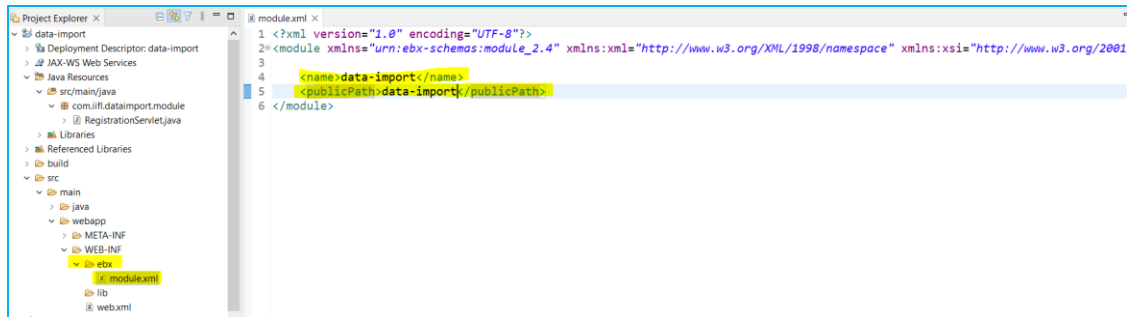
- Module can be registered with the EBX by defining the class extending the EBX extending the EBX class Module Registration Servlet. Create the class inside the package created above.



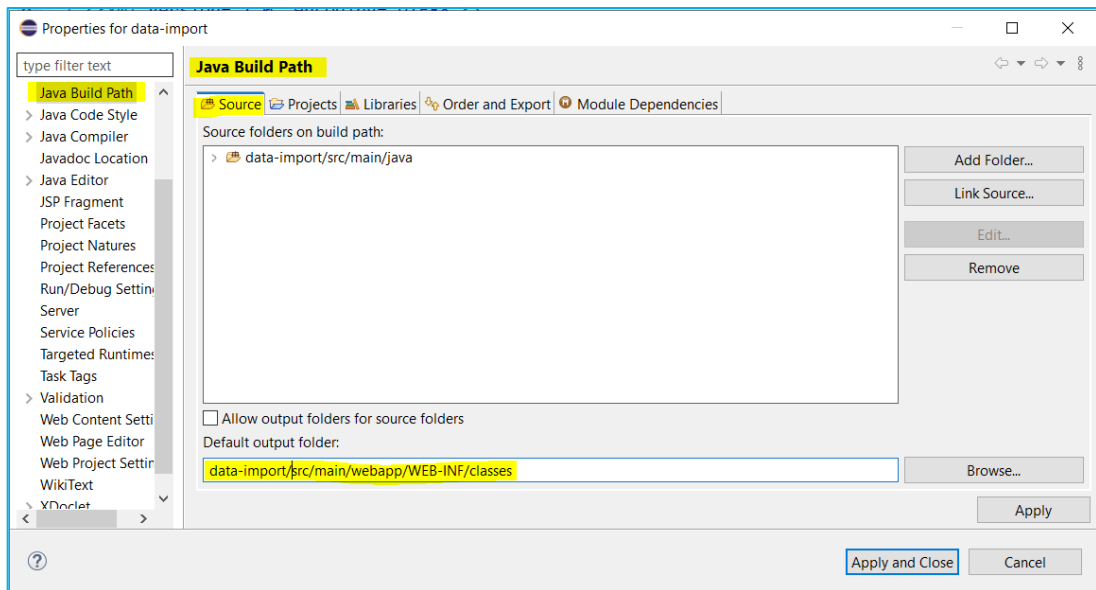
- Once the class is created make the servlet entry inside the web.xml file.



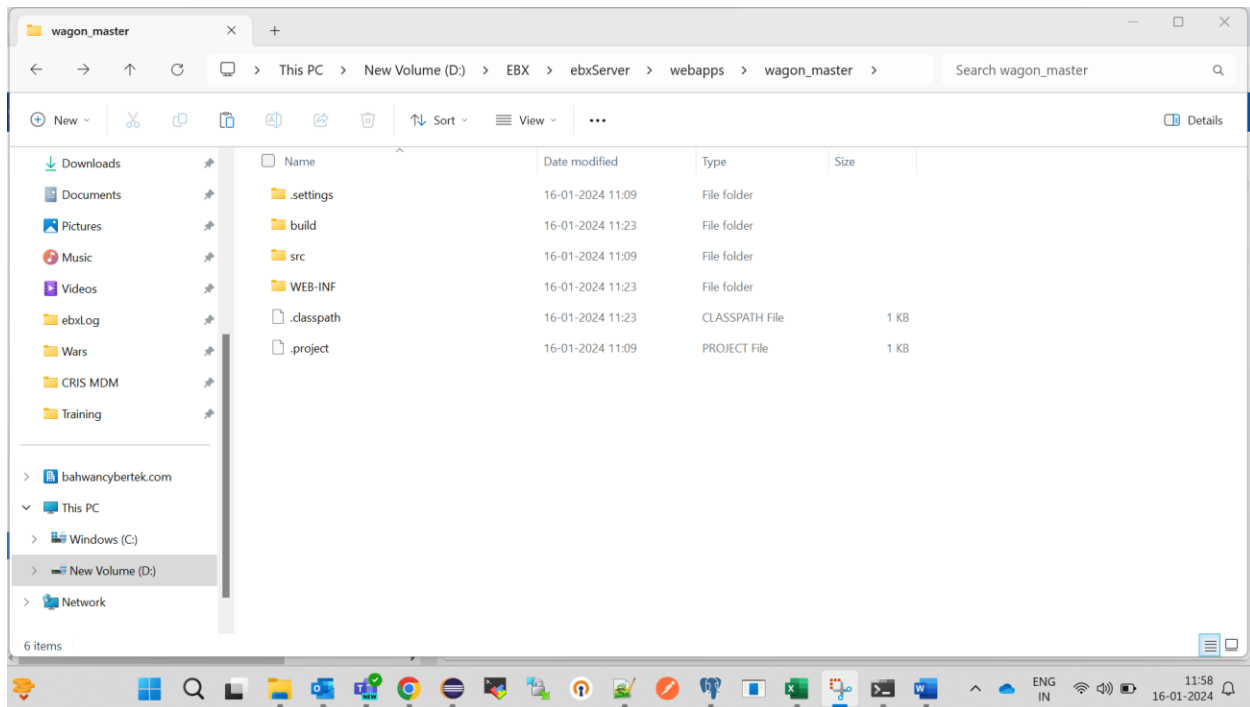
- Create ebx folder inside the WEB-INF folder. And also create the module.xml inside the WEB-INF folder.



7. Ensure that you set the class file generation location to classes folder inside the WEB-INF folder. This can be done through the Configure build path window.

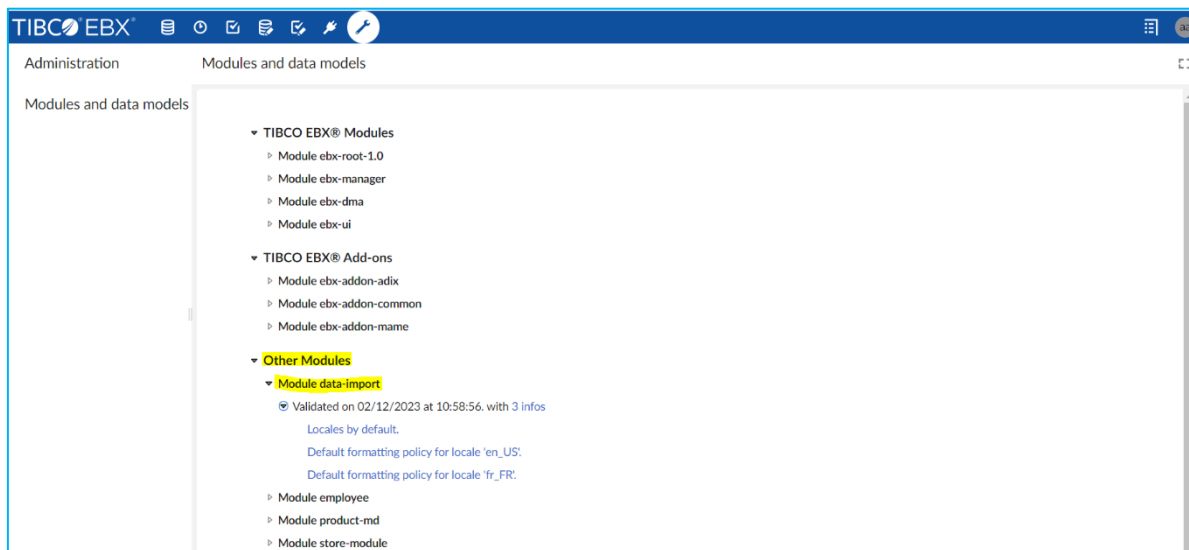


8. Copy paste the WEB-INF folder from src/main/webapp to parent folder

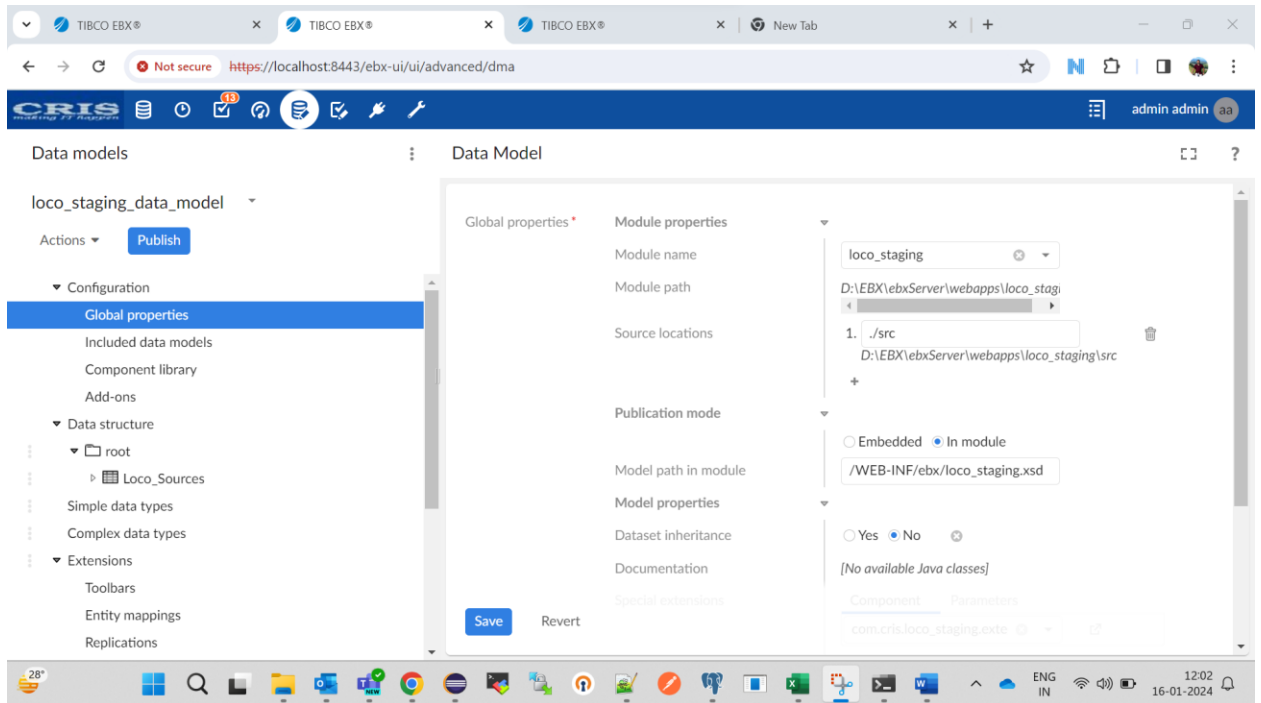


9. Restart the EBX server.

10. Check in the Modules and data model screen of the TIBCO EBX if the module is registered.

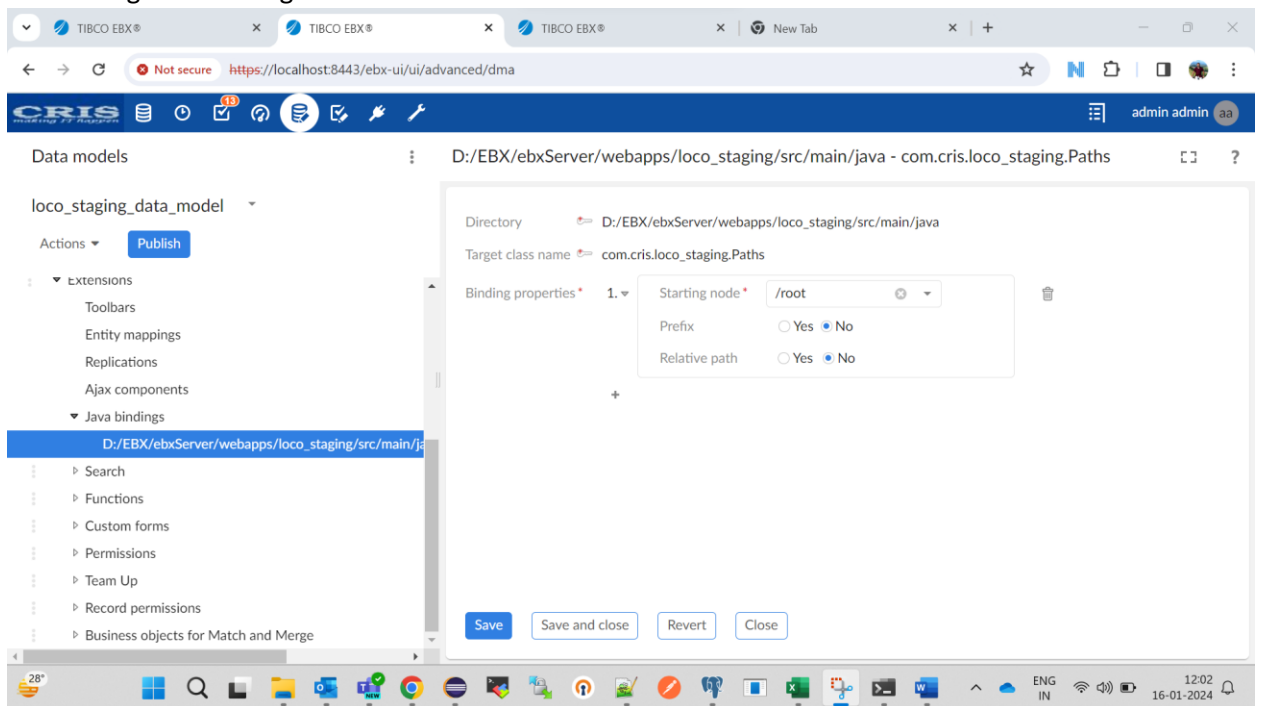


11. Data model publishing configuration



12.

13. Generating Java Bindings



14.

15. Publish the data model.