How I Created the API and Javadoc in Eclipse

Step 1: Creating My Library (the API)

I created a new Java Project called 'people-lib' in Eclipse. Inside the src folder, I created a package called week4 and added three classes: Person, Student, and Staff.

I marked the classes and key methods as public so they can be accessed from other projects.

Step 2: Exporting the Compiled Library

I exported the project as a JAR file by right-clicking on it and selecting Export → Java → JAR File. I named it people-lib.jar and saved it in my workspace or a shared libs folder.

Step 3: Generating Javadoc

I generated documentation by going to Export → Java → Javadoc. I selected all three classes in the week4 package, chose the visibility level as Public, and selected a destination folder, such as C:\Users\Colette.Kirwan\eclipse-workspace\people-lib\doc.

Eclipse ran the javadoc tool and created HTML documentation including an index.html file that lists all classes and methods.

Step 4: Attaching the Javadoc to My Library

In my people-app project, I added people-lib.jar to the build path (Properties → Java Build Path → Libraries → Add External JARs). Then I expanded the entry for people-lib.jar, selected Javadoc location → Edit, and chose 'Javadoc in folder'. I browsed to the doc folder that contained the generated HTML documentation.

After validation, Eclipse confirmed that the Javadoc location was successfully linked. Now, when I hover over any class or method from week4 in my main project, the documentation appears.

Step 5: Verifying the Link

I opened my Main.java file and hovered over Person or Student. Eclipse showed the tooltip from the attached Javadoc. Pressing F2 while the tooltip was visible opened a scrollable Javadoc window. I also opened the Javadoc View from Window → Show View → Java → Javadoc to see it update automatically.

Step 6: Understanding the Result

By completing these steps, I created a reusable Java API (people-lib.jar) and documented it using Javadoc. The result is a clean, shareable library that others can import into their projects and understand through the Javadoc tooltips and HTML pages.