What Is JavaDoc

JavaDoc is a special tool that is packaged with the JDK. It is used to generate the code documentation of Java source code in HTML format.

This tool uses “doc comments” format to document Java classes. IDEs like Eclipse, IntelliJIDEA, or NetBeans have an in-built JavaDoc tool to generate HTML documentation.

Apart from source code documentation this tool also provides API that creates “doclets” and “taglets” that we use to analyze the structure of a Java application.

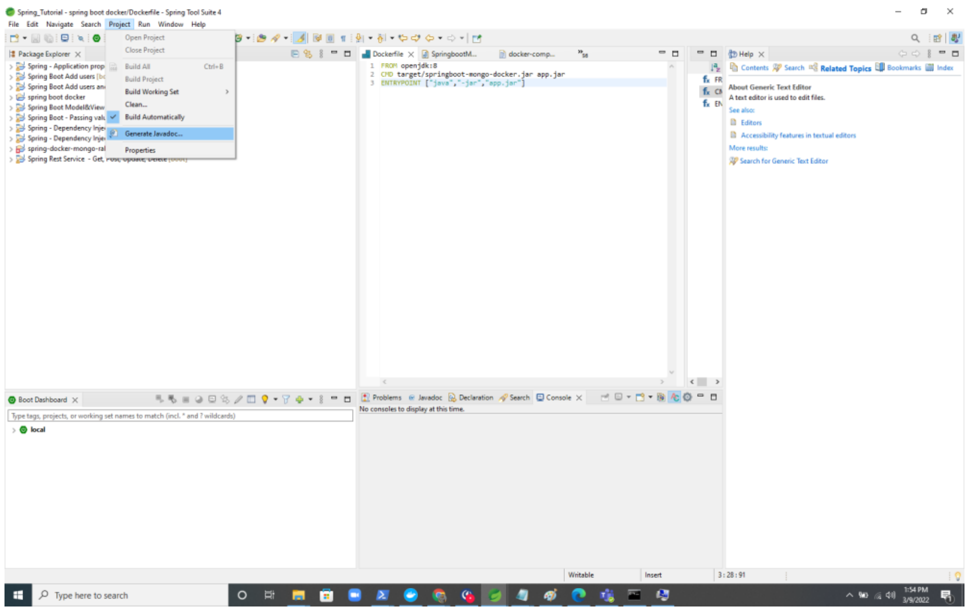
How to Write Doc Comments for the Javadoc Tool: Refer below website

https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html

Option 1: Eclipse

1. Goto Project ->Generate JavaDoc
2. Select the SRC folder [Under Select Types for which Javadoc will be generated]
3. Choose "Use Standard Docklet" and choose the destination path.
4. Choose Next and Click Finish.

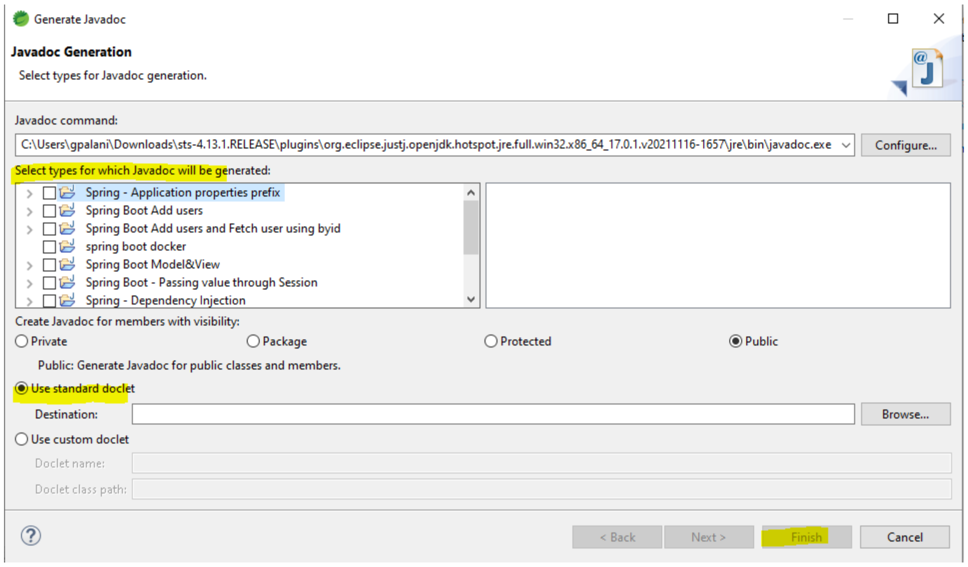
Note: For any project, import the project as a Gradle project into eclipse which is required for the dependencies to load for the Javadoc to get compiled.



```
FROM openjdk:8
CMD target/springboot-mongo-docker.jar app.jar
ENTRYPOINT ["java","-jar","app.jar"]
```

About Generic Text Editor
A text editor is used to edit files.

No consoles to display at this time.


Generate Javadoc

Javadoc Generation

Select types for Javadoc generation.

Javadoc command:

C:\Users\gpalani\Downloads\sts-4.13.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javadoc.exe

Select types for which Javadoc will be generated:

- Spring - Application properties prefix
- Spring Boot Add users
- Spring Boot Add users and Fetch user using byid
- spring boot docker
- Spring Boot Model&View
- Spring Boot - Passing value through Session
- Spring - Dependency Injection

Create Javadoc for members with visibility:

Private Package Protected [✓ Public]

Public: Generate Javadoc for public classes and members.

[✓ Use standard doclet]

Destination:

Use custom doclet

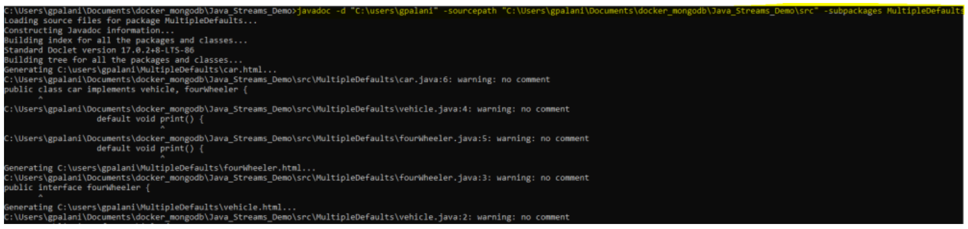
Doclet name:

Doclet class path:

Option 2: Through Command Prompt

Syntax: javadoc -d "Document destination path" -sourcepath "Source code main path" -subpackages "Sub Packages folder name"

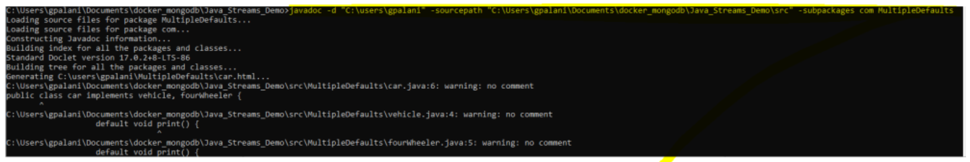
Sample 1: javadoc -d " C:\Users\gpalani\Downloads\spring-docker-mongo-rabbitmq-gradle-add-class-level-documentation\Java Doc " -sourcepath " C:\Users\gpalani\Downloads\spring-docker-mongo-rabbitmq-gradle-add-class-level-documentation\src\main\java\com\nexient\springmongodockerrabbitmqgradle" -subpackages "cotroller domain repository service"



```
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo>javadoc -d "C:\users\gpalani" -sourcepath "C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src" -subpackages MultipleDefaults
Loading source files for package MultipleDefaults...
Constructing Javadoc information...
Building index for all the packages and classes...
Standard Doclet version 17.0.2+8-LTS-86
Building tree for all the packages and classes...
Generating C:\users\gpalani\MultipleDefaults\car.html...
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\car.java:6: warning: no comment
public class car implements vehicle, fourWheeler {
       ^
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\vehicle.java:4: warning: no comment
                   default void print() {
                                ^
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\fourWheeler.java:5: warning: no comment
                   default void print() {
                                ^
Generating C:\users\gpalani\MultipleDefaults\fourWheeler.html...
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\fourWheeler.java:3: warning: no comment
public interface fourWheeler {
       ^
Generating C:\users\gpalani\MultipleDefaults\vehicle.html...
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\vehicle.java:2: warning: no comment
```

Sample 2 for multiple sub folders: javadoc -d "C:\users\gpalani\javadoc" -sourcepath "C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src" -subpackages com MultipleDefaults

In the sample 2, the project structure had 2 different sub packages:-com and MultipleDefaults.



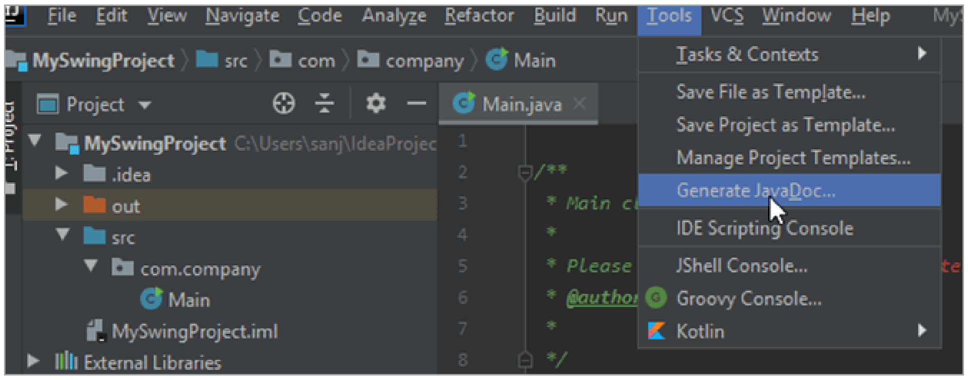
```
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo>javadoc -d "C:\users\gpalani" -sourcepath "C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src" -subpackages com MultipleDefaults
Loading source files for package MultipleDefaults...
Loading source files for package com...
Constructing Javadoc information...
Building index for all the packages and classes...
Standard Doclet version 17.0.2+8-LTS-86
Building tree for all the packages and classes...
Generating C:\users\gpalani\MultipleDefaults\car.html...
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\car.java:6: warning: no comment
public class car implements vehicle, fourWheeler {
       ^
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\vehicle.java:4: warning: no comment
                   default void print() {
                                ^
C:\Users\gpalani\Documents\docker_mongodb\Java_Streams_Demo\src\MultipleDefaults\fourWheeler.java:5: warning: no comment
                   default void print() {
```

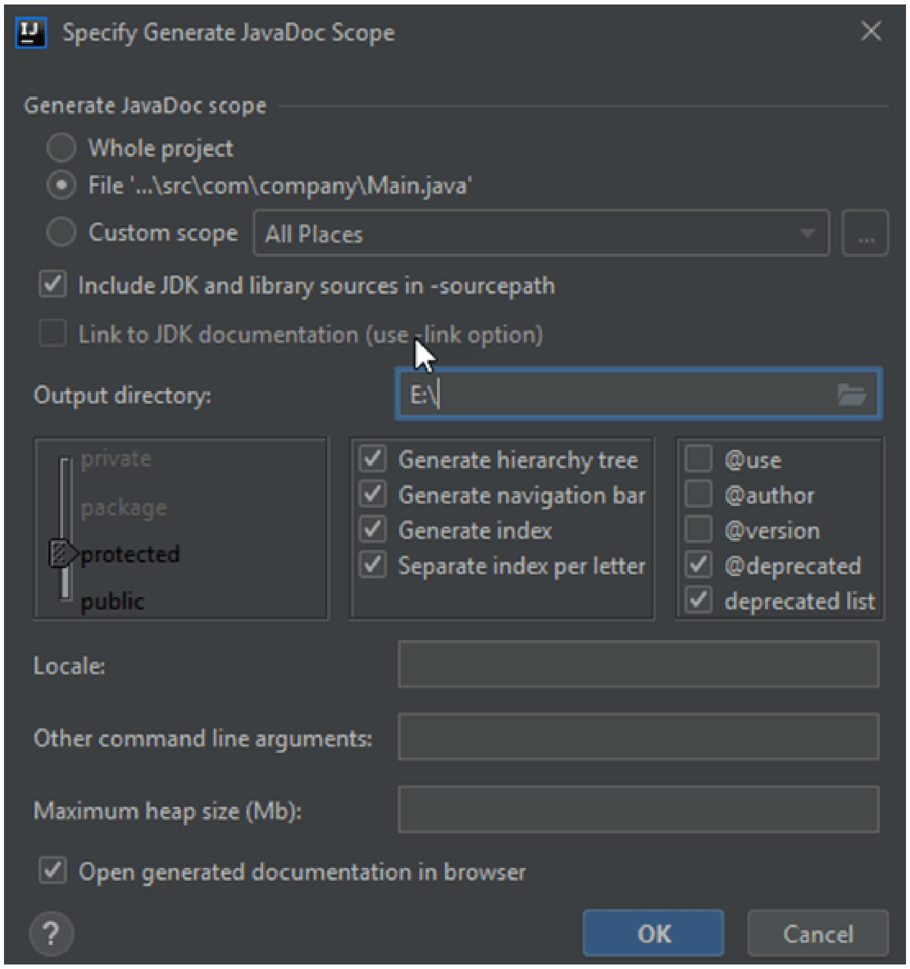
Option 3: Intellij

select Tools -> Generate JavaDoc

select a scope — a set of files or directories for which you want to generate the reference [Standard one is Whole Project]

Set the Output Directory and click OK.


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MySwingProject src com company Main
Project
Main.java
MySwingProject C:\Users\sanj\IdeaProjec
.idea
out
src
com.company
Main
MySwingProject.iml
External Libraries
Tasks & Contexts
Save File as Template...
Save Project as Template...
Manage Project Templates...
Generate JavaDoc...
IDE Scripting Console
JShell Console...
Groovy Console...
Kotlin


Specify Generate JavaDoc Scope

Generate JavaDoc scope

- Whole project
- [✓ File '...\src\com\company\Main.java']
- Custom scope: All Places

- [✓] Include JDK and library sources in -sourcepath
- [ ] Link to JDK documentation (use -link option)

Output directory: E:\

private / package / [✓ protected] / public

- [✓] Generate hierarchy tree
- [✓] Generate navigation bar
- [✓] Generate index
- [✓] Separate index per letter

- [ ] @use
- [ ] @author
- [ ] @version
- [✓] @deprecated
- [✓] deprecated list

Locale:

Other command line arguments:

Maximum heap size (Mb):

- [✓] Open generated documentation in browser