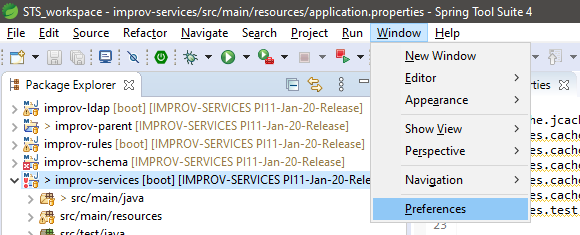
How to Set Up Your Local STS Eclipse IDE for the IMPROV-SERVICES Java Maven Project

# How to Add a JDK to Your Local STS Eclipse IDE

## From the main window 🡪 “Window” 🡪 “Preferences”



## In the “Preferences” window 🡪 “Installed JREs” 🡪 “Add…”

Graphical user interface, application

Description automatically generated

## In the “Add JRE” window 🡪 “Standard VM” 🡪 “Next”

Graphical user interface, text, application

Description automatically generated

## In the next window 🡪 “Directory…” 🡪 select a local JDK installed on your Desktop. (Needs to be Java 1.8 compatible.) 🡪 “Finish”

Must be a JDK as you will need the Java development tools included in the JDK for generating the Swagger classes.

Graphical user interface, text, application, email

Description automatically generated

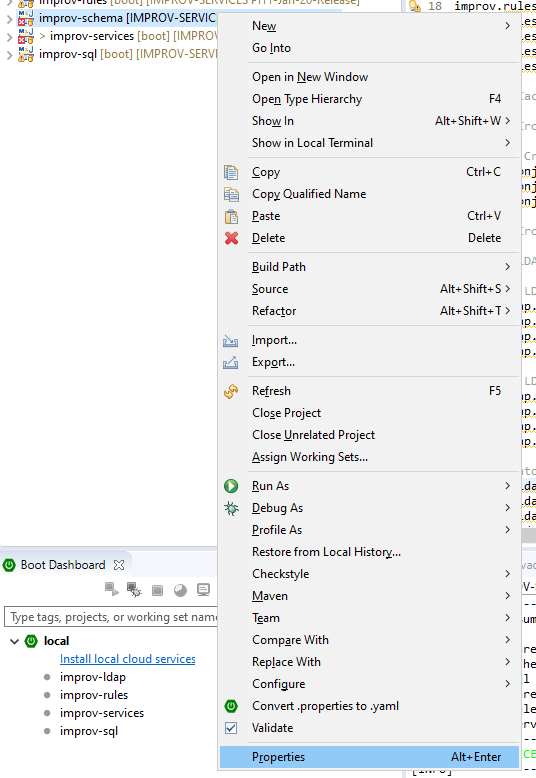
## In the “Preferences” window 🡪 select your newly added JDK 🡪 “Apply and Close”

Graphical user interface, application

Description automatically generated

# How to Specify the JDK in the improv-schema project

## From the main window, right click the improv-schema project 🡪 “Properties”



## In the “Properties from improv-schema” window, select “Java Build Path” 🡪 “JRE System Library” 🡪 “Edit…”

Graphical user interface, text

Description automatically generated

## In the “Edit Library” window, select “Workspace default JRE”. (The JDK you set in the “Preferences” window will be listed.) 🡪 “Finish”

Graphical user interface, text, application, email

Description automatically generated

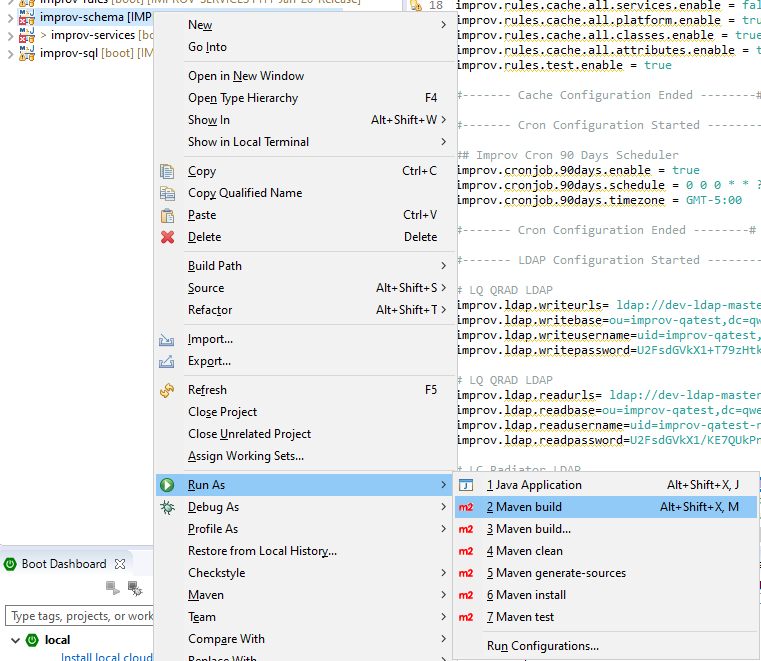
## In the “Properties from improv-schema” window, “Apply and Close”

Graphical user interface, text, application

Description automatically generated

# How to Add the improv-schema Project JAR to the improv-services Project

## From the main window, right click the improv-schema project 🡪 “Run As” 🡪 “Maven build”



## From the “Run Configurations” window 🡪 Set “Goals” to “install” 🡪 “Apply” 🡪 “Run”

Graphical user interface, text, application, email

Description automatically generated

## Wait for the build to complete. Once done, make sure the “improv-schema-1.0.0.jar” is in the “improv-schema” project

From the main window, right click the improv-schema project 🡪 “target” folder

Text

Description automatically generated with medium confidence

If the JAR is not in the folder, refresh the “improv-schema” project. Right clock the “improv-schema” project 🡪 “Refresh”

A screenshot of a computer

Description automatically generated

## From the main window, right click the improv-services project 🡪 “Properties”

A screenshot of a computer

Description automatically generated

## From the “Properties for improv-services” window 🡪 “Libraries” tab 🡪 “Add JARS…”

Graphical user interface, text, application

Description automatically generated

## From the “JAR Selection” window 🡪 “improv-schema” 🡪 “target” 🡪 “improv-schema-1.0.0.jar” 🡪 “OK”

Graphical user interface, text, application

Description automatically generated

## From the “Properties for improv-services” window 🡪 “Apply and Close”

Graphical user interface, text, application

Description automatically generated

## You will now see the “improv-schema-1.0.0.jar” being referenced by the “improv-services” project.

From the main window 🡪 “Referenced Libraries” 🡪 “improv-schema-1.0.0.jar”

Graphical user interface, text, application

Description automatically generated

# How to Build the multi-module IMPROV-SERVICES Project

## From the main window, right click the “improv-parent” project 🡪 “Run As” 🡪 “Maven build”

A screenshot of a computer

Description automatically generated

## From the “Run Configurations” window 🡪 Set “Goals” to “clean install” 🡪 Check “Skip Tests” box 🡪 “Apply” 🡪 “Run”

Graphical user interface, text, application

Description automatically generated

## Wait for the build to complete successfully.

Table

Description automatically generated