

# Spring Maven Project

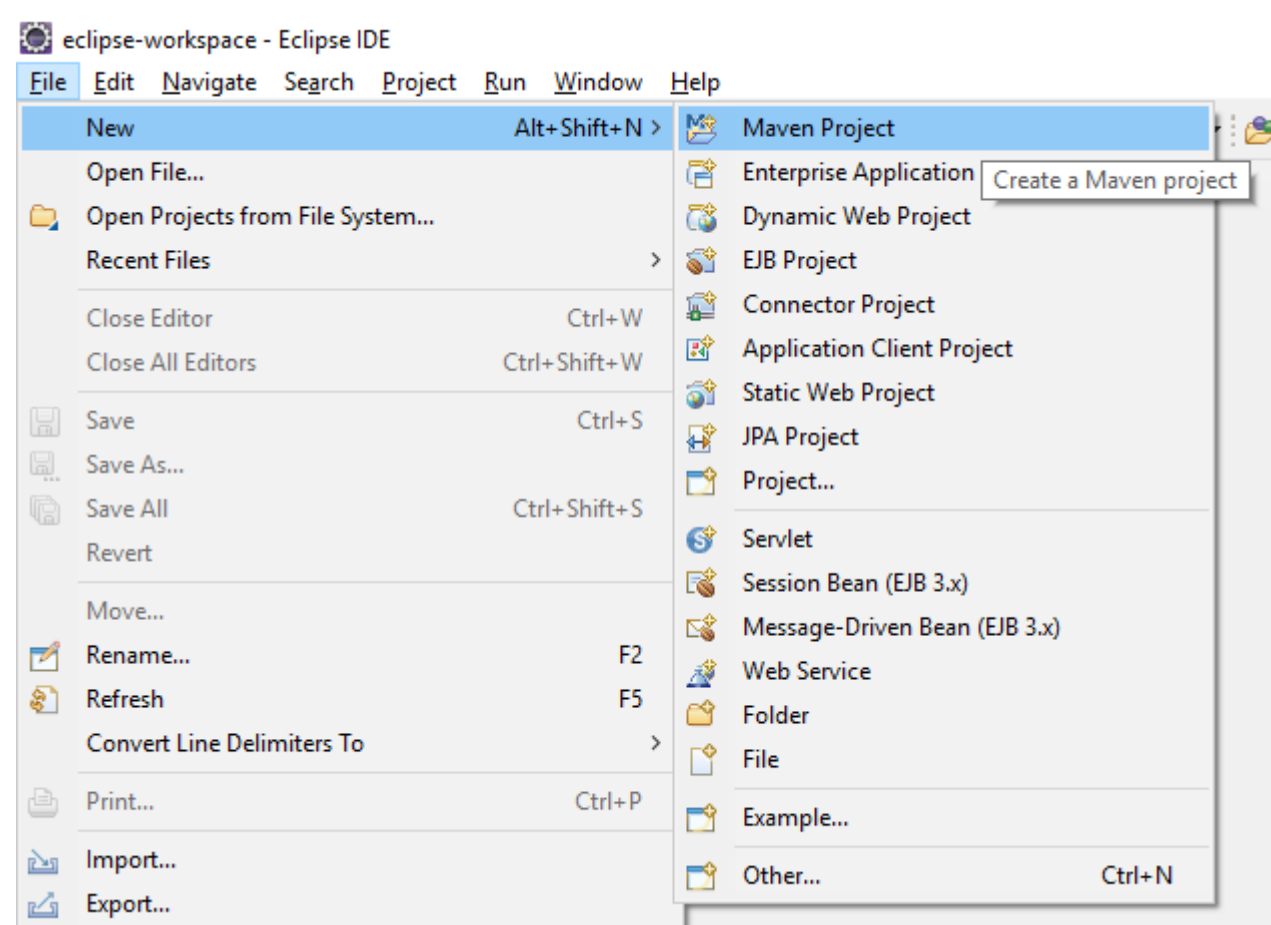
based Spring project. Maven is automation tool which acts as dependency manager for Java applications.

for Spring and installed Eclipse, Java, and Apache Tomcat Server.

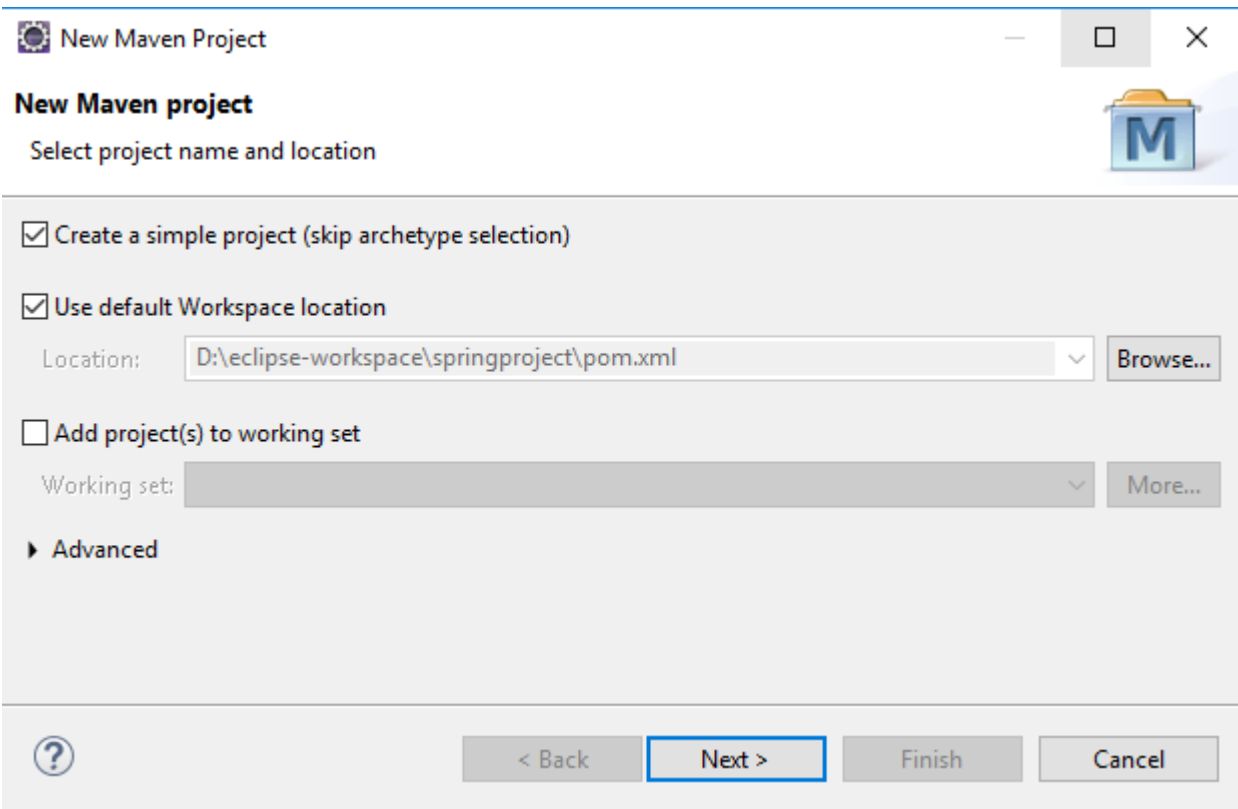
If you have not setup environment yet then we recommend you to read our **previous topic**.

## Step1: Create Maven Project

Open Eclipse and click on **File** menu and then select **New --> Maven Project** as we did in the given screenshot.



After selecting maven, it will open a window like below and will ask to select archetype. Here, we clicked on Checkbox and then **press next** button.



The screenshot shows the 'New Maven Project' dialog box in the Eclipse IDE. The title bar reads 'New Maven Project'. Below the title bar, the text 'New Maven project' is displayed, followed by the instruction 'Select project name and location'. A small icon of a folder with a blue 'M' is visible on the right. The dialog contains several options: a checked checkbox for 'Create a simple project (skip archetype selection)', a checked checkbox for 'Use default Workspace location', and an unchecked checkbox for 'Add project(s) to working set'. The 'Location' field is set to 'D:\eclipse-workspace\springproject\pom.xml' with a 'Browse...' button next to it. The 'Working set' field is empty with a 'More...' button next to it. An 'Advanced' section is collapsed. At the bottom, there are four buttons: a help button (question mark), '< Back', 'Next >' (which is highlighted with a blue border), 'Finish', and 'Cancel'.

## Step 2: Configure the Project

Set project name by providing group id and artifact id. Artifact id represents the project name, So give a meaning name for your project after that select packaging that specifies the plugin goals that are executed during each **Maven build** phase. After that **press Finish** button and your maven project is ready.

The screenshot shows the 'New Maven Project' dialog box in Eclipse. The title bar says 'New Maven Project'. Below the title bar, there's a section titled 'New Maven project' with a sub-label 'Configure project' and a Maven logo icon. The dialog is divided into several sections: 'Artifact' with fields for Group Id (com.studytonight), Artifact Id (spring-demo), Version (0.0.1-SNAPSHOT), Packaging (jar), Name, and Description; 'Parent Project' with fields for Group Id, Artifact Id, and Version, along with 'Browse...' and 'Clear' buttons; and an 'Advanced' section which is currently collapsed. At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'. A help icon (?) is also present in the bottom left corner.

New Maven Project

New Maven project

Configure project

Artifact

Group Id: com.studytonight

Artifact Id: spring-demo

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

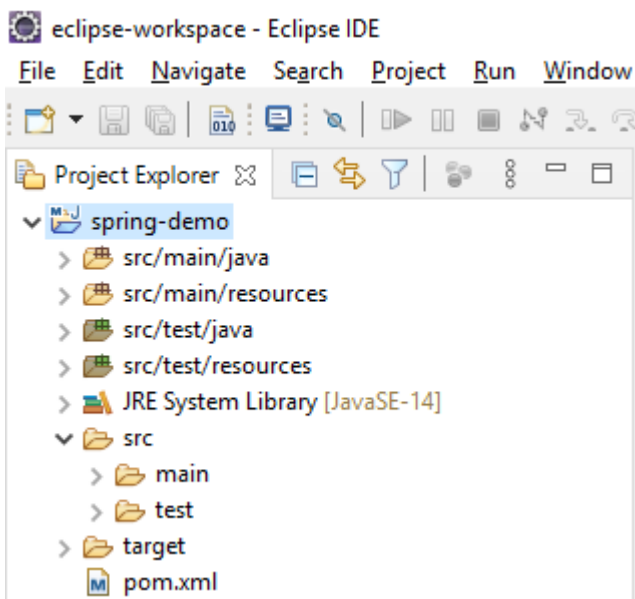
Browse... Clear

Advanced

< Back Next > Finish Cancel

# Maven Project Stucture

After above step, Eclipse creates a new project that look like the below screeshot. This project contains a **pom.xml** file that is used to configure maven project. We will discuss that later in our tutorial.



Fine, till here, we have created a maven based spring project successfully. Now, **in next chapter, we will learn to create a simple hello world application** to understand the flow and structure of the application.