



# Creating a Maven Project in Spring Tool Suite (STS)

This document provides a comprehensive step-by-step guide for creating a Maven project using Spring Tool Suite (STS). It covers the entire process from opening STS to configuring project details, along with common issues and their solutions. By following these instructions, you will be able to set up a Maven project efficiently and effectively.

## Step-by-Step Guide

### 1. Open Spring Tool Suite (STS)

Launch the Spring Tool Suite application on your computer.

### 2. Create New Maven Project:

- Click on **File** → **New** → **Project**.
- In the New Project wizard, expand the "Maven" folder.
- Select **Maven Project**.
- Click **Next**.

### 3. Select Project Location:

- Ensure that "Create a simple project (skip archetype selection)" is unchecked.
- Click **Next**.

### 4. Choose Archetype:

- From the Catalog, choose **internal**.
- In the Filter box, type **maven-archetype-webapp** for a web application or **maven-archetype-quickstart** for a simple Java project.
- Select the appropriate archetype (usually version 1.0 or the latest).
- Click **Next**.

### 5. Configure Project Details:

- **Group Id:** Enter your organization's reverse domain (e.g., **com.example**).
- **Artifact Id:** Enter your project name (e.g., **my-webapp**).
- **Version:** Specify your version number (default: **0.0.1-SNAPSHOT**).
- **Package:** Set the base package name (usually the same as Group Id).
- Click **Finish**.

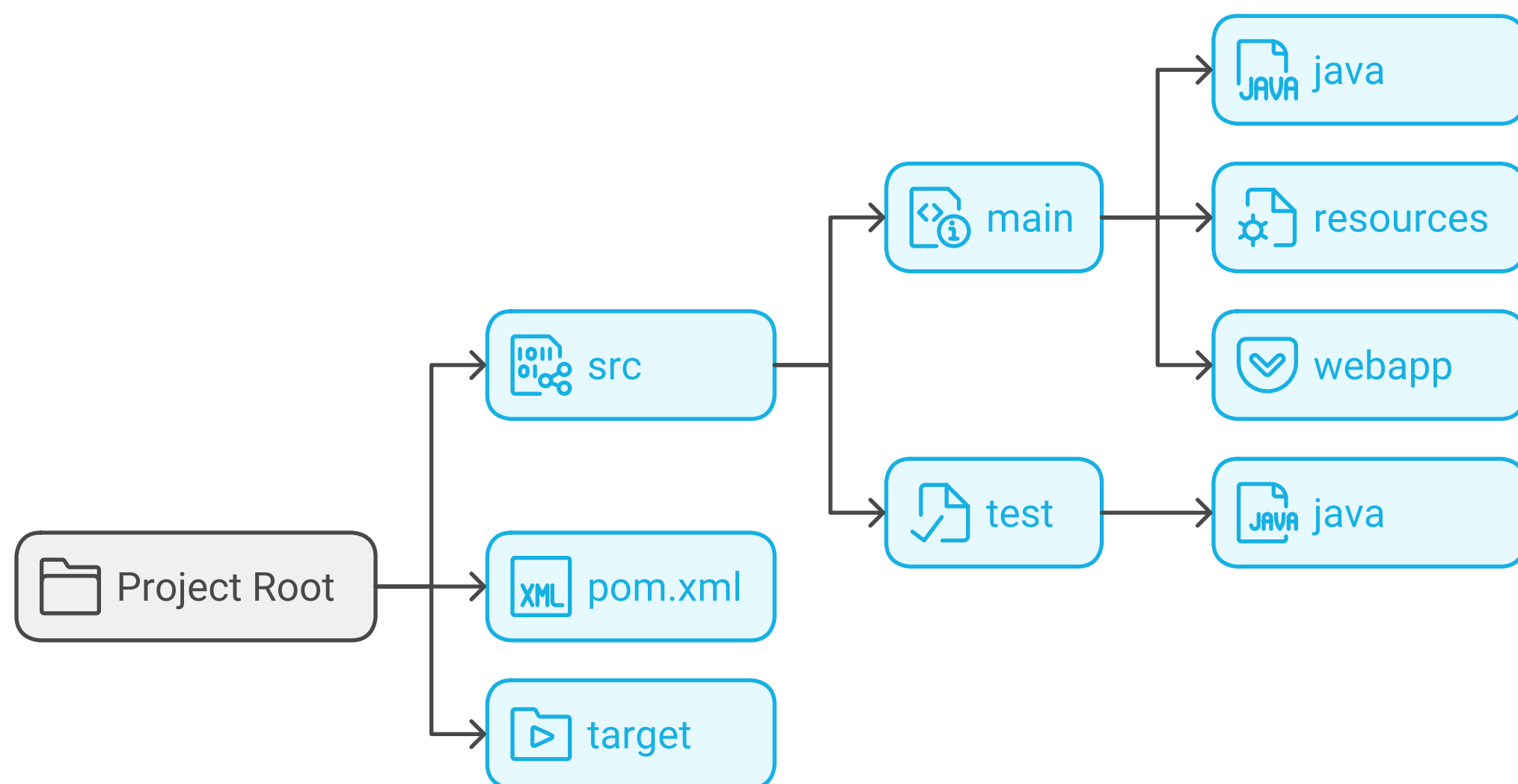
### 6. Project Structure:

After the project is created, the structure will look like this:

```

project-name/
├── src/
│   ├── main/
│   │   ├── java/    (Java source files)
│   │   ├── resources/ (Configuration files)
│   │   └── webapp/   (Web resources, if web project)
│   └── test/
│       └── java/    (Test files)
├── pom.xml
└── target/        (Compiled files)

```



## 7. Basic POM.xml Structure:

Here is a basic structure for your **pom.xml** file:

```

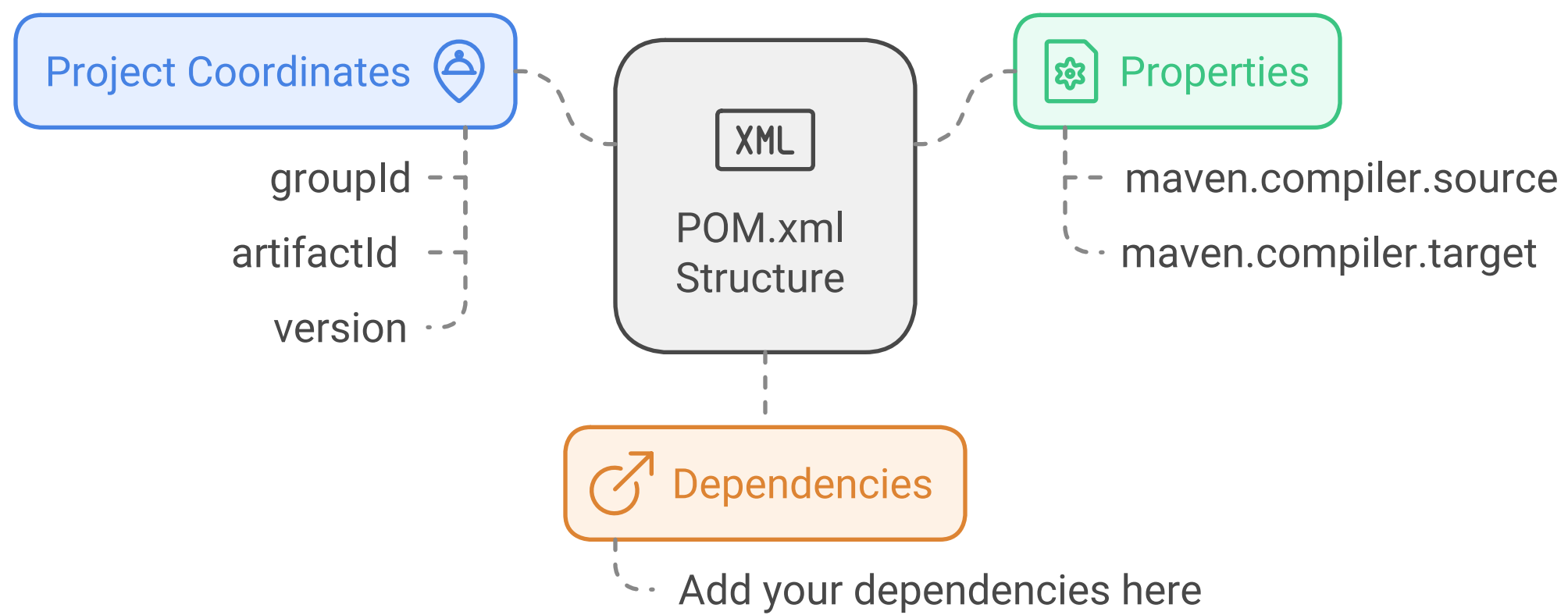
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>my-webapp</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- Add your dependencies here -->
  </dependencies>
</project>

```



## 8. Update Project (if needed):

- Right-click on the project.
- Select **Maven** → **Update Project**.
- Check "Force Update of Snapshots/Releases".
- Click **OK**.

## 9. Additional Configuration (Optional):

- Right-click on the project → **Properties**.
- **Java Compiler**: Set the JDK compliance level.
- **Java Build Path**: Configure source folders.
- **Project Facets**: Enable or disable relevant facets.

## 10. Verify Installation:

- Check for any error markers in the Project Explorer.
- Build the project by selecting **Project** → **Clean...** → **Clean**.
- Check the Console for any build errors.

## Common Issues & Solutions:

### 1. If you see red marks in pom.xml:

- Right-click on the project → **Maven** → **Update Project**.

### 2. If dependencies aren't downloading:

- Check your internet connection.
- Verify the **settings.xml** in your Maven configuration.
- Try clearing the Maven local repository.

### 3. For JDK compliance errors:

- Update Java compiler settings in **pom.xml**.
- Check the project Java Build Path.

Would you like me to elaborate on any of these steps or explain how to add specific dependencies to your project?