



Maven training

Overview:

Maven is an open-source build tool from Apache Group. It is a widely-used and popular tool to build, publish, and deploy software development projects. It is used to build and manage diverse Java-based projects. A powerful project management tool, it is based on the project object model (POM) concept.

A java-based tool, Maven is used to creating projects coded in Scala, C#, Ruby, and others. It simplifies the day-to-day work of Java developers and helps them with various tasks. It offers excellent interoperability by way of plugins with IDEs such as Eclipse, and can be installed and operated on popular OSs - Windows, and Ubuntu.

This tutorial provides a comprehensive overview of Maven, its uses, and in-depth information on working on it. Learners with sound knowledge of software development and basics of Java programming language can easily pick up this tool, which is becoming increasingly popular today with developers and employers alike.

Content:

Introduction

- Download Notes
- What is Maven
- Why Maven?
- Maven Installation and Setup
- Maven installation details

Maven Project Creation and Key Concepts

- A simple java projects
- POM XML
- Specify Compiler Source and Target
- Building the project from commandline
- Plugins
- Life Cycle Phases
- Coordinates
- Repositories
- Assignment
- Maven project creation and Key Concepts

Maven in Eclipse

- Usecase
- Create a Maven Project in Eclipse
- Customize the compiler settings
- Create the Data Access Layer
- Implement the Data Access Layer
- Upgrading JUnit Version
- Using JUnit 5

- Writing Unit Tests
- Building from the Command Line
- Create the Business Object Layer
- Add Spring Dependencies
- Dependency Management
- Build the project
- Maven in Eclipse

Maven Web Application

- Create the web application
- Create the Servlets
- Add The Servlet Dependency
- Servlet API Update
- Create the product.html
- Maven Web Application

Multi Module Project Creation

- Create the Parent POM
- Update the Child POMs
- Build the multi module project
- Add the product services dependency
- Code the Display Product Servlet
- Build the final project
- Run the application
- Troubleshooting
- Multi module project creation

Organizing the multi module project

- Move the projects to a parent folder
- Troubleshooting Update
- Import the projects in to eclipse
- Import all the projects at once

Scopes

- The Six Scopes
- Scopes in our projects
- Scopes

Dependency Management

- Manage Dependencies
- Manage Plugins

Profiles

- Usecase
- Create the project
- Create Configuration
- Configure Profiles
- Profiles in action

- Profiles

JaCoCo Code Coverage and Sonar

- Configure Jacoco Test Coverage
- Install and Launch Sonar
- Troubleshooting
- Generate a sonar report

Additional Content - Spring Boot Quick Start

- Problems with traditional spring apps
- Spring Boot Features
- Install STS
- Configure JDK in STS
- Spring Boot in action
- Create a spring boot project
- Spring boot starters in pom
- @SpringBootApplication
- @SpringBootTest
- Do Dependency Injection
- Testing
- Spring Boot Quick Start

