Manjula Nannuri

Day 24_Assignment

Task 1: Build Lifecycle

Demonstrate the use of Maven lifecycle phases (clean, compile, test, package, install, deploy) by executing them on a sample project and documenting what happens in each phase.

Certainly! Below is a brief explanation of each Maven lifecycle phase and its purpose, along with a demonstration of executing them on a sample project:

1. **clean**: This phase removes all files generated by the previous builds. It ensures that the project starts from a clean state.

2. **compile**: This phase compiles the source code of the project.

3. **test**: This phase runs the tests of the project using a suitable testing framework like JUnit.

4. **package**: This phase packages the compiled code (along with resources) into distributable formats such as JAR, WAR, or EAR.

5. **install**: This phase installs the packaged artifact into the local Maven repository. The artifact can be used as a dependency by other local Maven projects.

6. **deploy**: This phase deploys the packaged artifact to a remote repository, making it available for other projects or developers.

Here's how you can execute these lifecycle phases on a sample Maven project:

1. **Create a Sample Maven Project**:

First, create a new Maven project using the following command:

```
mvn archetype:generate -DgroupId=com.example -DartifactId=my-maven-project -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

This command creates a simple Maven project with a basic directory structure and a sample Java class.

2. **Navigate to the Project Directory**:

```
cd my-maven-project
```

3. **Execute Maven Lifecycle Phases**:

Execute the Maven lifecycle phases one by one using the following commands:

- **clean**: This command removes all files generated by previous builds.

```
mvn clean
```

- **compile**: This command compiles the source code of the project.

```
mvn compile
```

```

- **test**: This command runs the tests of the project.

```
mvn test
```

- **package**: This command packages the compiled code into a JAR file.

```
mvn package
```

- **install**: This command installs the packaged artifact into the local Maven repository.

```
mvn install
```

- **deploy**: This command deploys the packaged artifact to a remote repository. Since we don't have a remote repository configured, this command will not perform any action.

```
mvn deploy
```

After executing each command, Maven will display the progress of the build process and any relevant output or errors.
```

4. **View the Output**:

You can view the generated artifacts (e.g., JAR file) in the `target` directory of your Maven project after executing the `package` phase.

By executing these Maven lifecycle phases, you can observe how Maven manages the build process of a project, from compiling the source code to packaging and deploying the artifacts. Each phase contributes to the overall build lifecycle and serves a specific purpose in the software development process.