

Lesson 01 Demo 03

Exploring Maven Clean Plugin

Objective: To explore the Maven Clean Plugin by creating a target directory and managing temporary files through Maven commands and configurations

Tools Required: Visual Studio Code and Maven

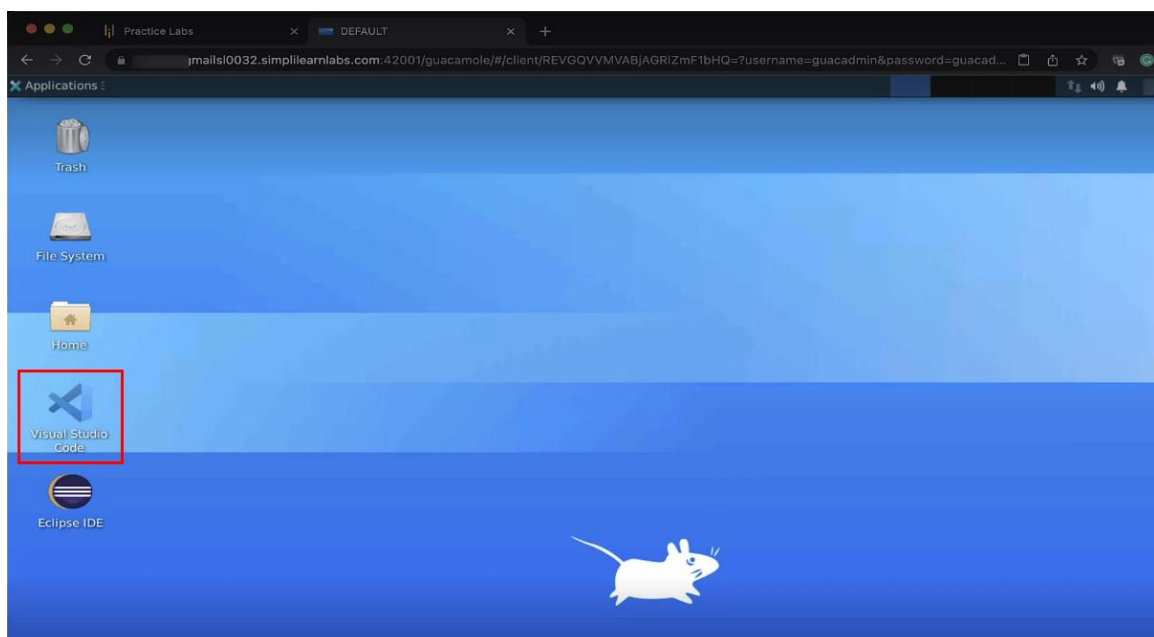
Prerequisites: None

Steps to be followed:

1. Create a target directory
2. Create a directory, temporary files, and logs
3. Handling Error

Step 1: Create a target directory

1.1 Open the Visual Studio Code editor



- 1.2 Go to the CMS project directory and execute the **mvn package** command to compile and build the JAR file

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
palakkharbandas@ip-172-31-17-240:~/cms$ mvn package
[INFO] Scanning for projects...
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ cms ---

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ cms ---
[INFO] Building jar: /home/palakkharbandas/cms/target/cms-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.808 s
[INFO] Finished at: 2024-05-27T12:25:43Z
[INFO] -----

```

Note: Please refer to the previous demo on how to create the CMS project

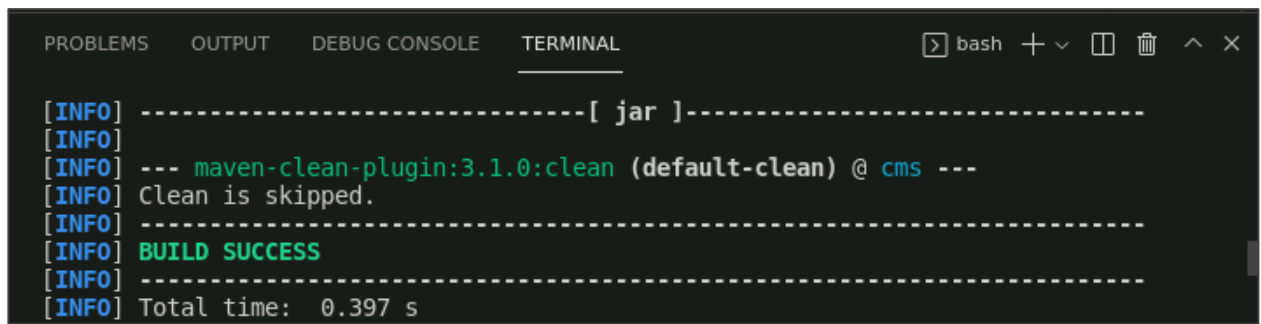
- 1.3 In the terminal of the Visual Studio Code, use the clean command to clean the project and delete the build artifacts:

mvn clean

```

palakkharbandas@ip-172-31-17-240:~/cms$ mvn clean
[INFO] Scanning for projects...
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----

```

A screenshot of an IDE terminal window. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal shows the output of a Maven build. It starts with '[INFO] -----[jar]-----', followed by '[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---', then '[INFO] Clean is skipped.', then '[INFO] BUILD SUCCESS' in green, and finally '[INFO] Total time: 0.397 s'. The terminal window has a dark background and standard window controls at the top right.

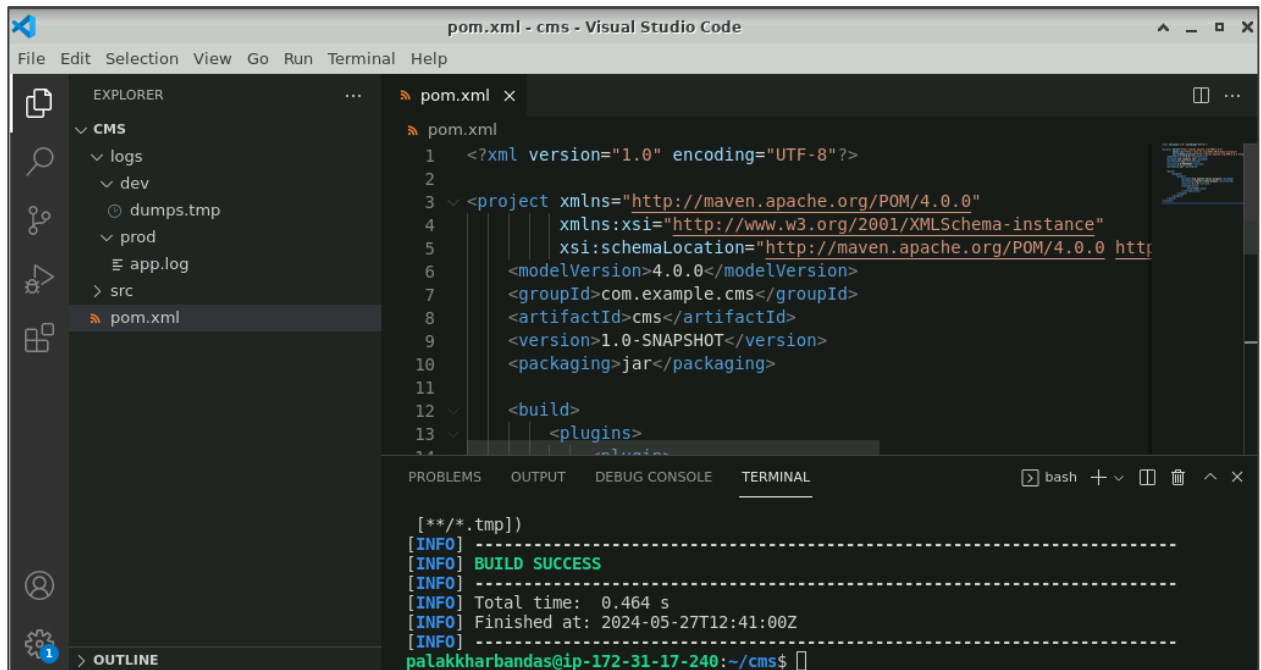
```
[INFO] -----[ jar ]-----  
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---  
[INFO] Clean is skipped.  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 0.397 s
```

The command **mvn clean** is used to clean the project by deleting all the build artifacts from the previous build. This is often done before building the project again to ensure a clean build without any leftover files from previous builds.

1.4 Add Clean Plugin Configuration in pom.xml:

```
<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.cms</groupId>
  <artifactId>cms</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-clean-plugin</artifactId>
        <version>3.1.0</version>
        <configuration>
          <skip>true</skip>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```



The screenshot shows the Visual Studio Code interface with the `pom.xml` file open. The file contains the following XML content:

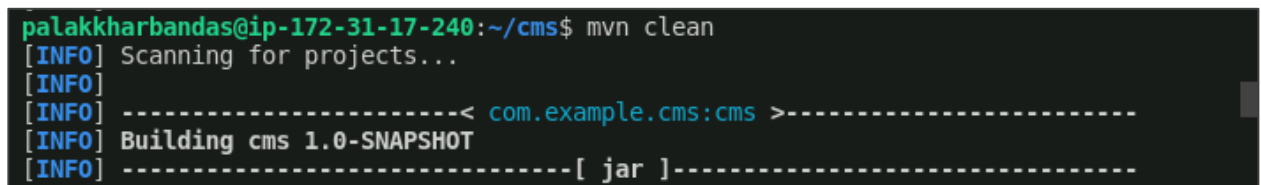
```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example.cms</groupId>
  <artifactId>cms</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <build>
    <plugins>
```

The terminal output at the bottom shows the following messages:

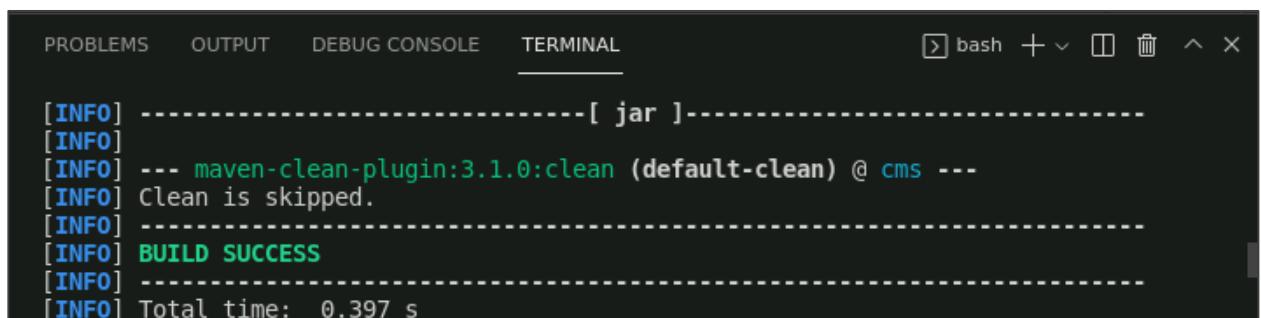
```
[**/*.tmp])
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.464 s
[INFO] Finished at: 2024-05-27T12:41:00Z
[INFO] -----
palakkharbandas@ip-172-31-17-240:~/cms$
```

These lines of code are used in Maven to configure the behaviour of a plugin.

1.5 Run the `mvn clean` command, which will clean the project and delete the build artifacts.



```
palakkharbandas@ip-172-31-17-240:~/cms$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
```



```
[INFO] -----[ jar ]-----
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---
[INFO] Clean is skipped.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.397 s
```

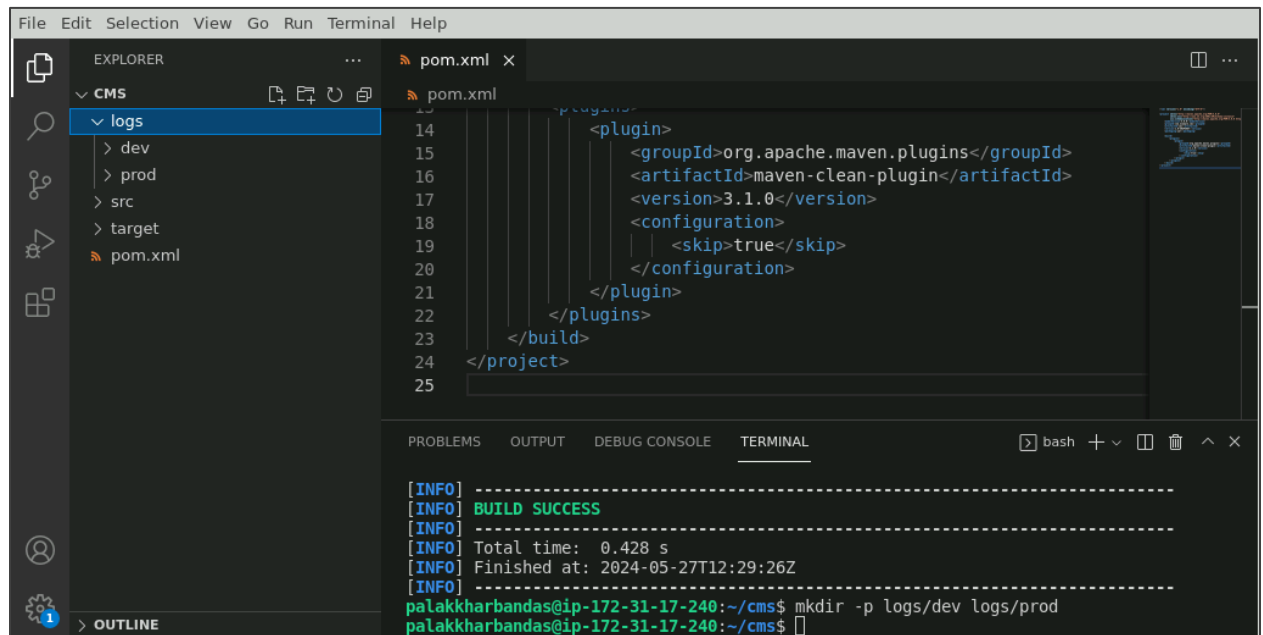
- 1.6 Without using `<configuration>` in the `pom.xml` file, you can skip the command from deleting the target folder by specifying the skip parameter in the `clean` command
- `mvn clean -Dmaven.clean.skip=true`

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  bash + - [ ] [ ] ^ X
palakkharbandas@ip-172-31-17-240:~/cms$ mvn clean -Dmaven.clean.skip=true
[INFO] Scanning for projects...
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  bash + - [ ] [ ] ^ X
[INFO] Clean is skipped.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.428 s
[INFO] Finished at: 2024-05-27T12:29:26Z
[INFO] -----
palakkharbandas@ip-172-31-17-240:~/cms$
```

Step 2: Create a directory, temporary files, and logs

2.1 In the **CMS** folder, create a directory called **logs** and other directories as **dev** and **prod**:



The screenshot shows an IDE interface with the following components:

- EXPLORER:** Displays the project structure. The **CMS** folder is expanded, showing subfolders: **logs** (selected), **dev**, **prod**, **src**, **target**, and **pom.xml**.
- Editor:** Displays the **pom.xml** file. The content is as follows:

```
14 <!-->  
15 <!-->  
16 <!-->  
17 <!-->  
18 <!-->  
19 <!-->  
20 <!-->  
21 <!-->  
22 <!-->  
23 <!-->  
24 <!-->  
25 <!-->
```
- TERMINAL:** Shows the output of a Maven build and a directory creation command.

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 0.428 s  
[INFO] Finished at: 2024-05-27T12:29:26Z  
[INFO] -----  
palakkharbandas@ip-172-31-17-240:~/cms$ mkdir -p logs/dev logs/prod  
palakkharbandas@ip-172-31-17-240:~/cms$
```



```

pom.xml
17 <version>3.1.0</version>
18 <configuration>
19 <filesets>
20 <fileset>
21 <directory>logs/dev</directory>
22 <includes>
23 <include>**/*.log</include>
24 </includes>
25 <excludes>
26 <exclude>**/*.tmp</exclude>
27 </excludes>
28 </fileset>
29 </filesets>
30 </configuration>

[INFO] -----
[INFO] Total time: 0.428 s
[INFO] Finished at: 2024-05-27T12:29:26Z
[INFO] -----
palakharbandas@ip-172-31-17-240:~/cms$ mkdir -p logs/dev logs/prod
palakharbandas@ip-172-31-17-240:~/cms$ touch logs/dev/app.log logs/dev/dumps.tmp log
s/prod/app.log
palakharbandas@ip-172-31-17-240:~/cms$

```

2.4 Run the **mvn clean** command again and the log file will be removed from the **dev** directory

```

pom.xml
17 <version>3.1.0</version>
18 <configuration>
19 <filesets>
20 <fileset>
21 <directory>logs/dev</directory>
22 <includes>
23 <include>**/*.log</include>
24 </includes>
25 <excludes>
26 <exclude>**/*.tmp</exclude>
27 </excludes>
28 </fileset>
29 </filesets>
30 </configuration>

palakharbandas@ip-172-31-17-240:~/cms$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---

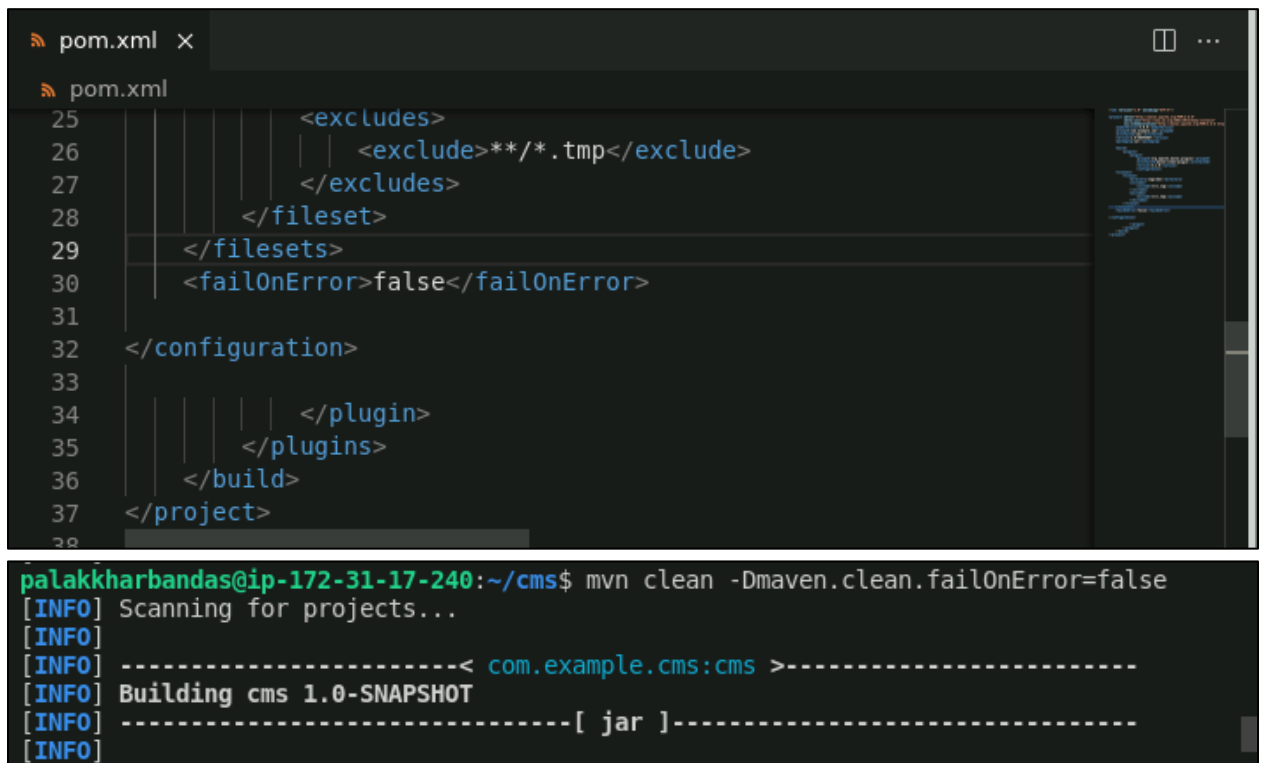
```

This command will remove the .log files from the logs/dev directory while excluding the .tmp files.

Step 3: Handling Error

3.1 To ignore errors during the clean process, add the following configuration to the pom.xml:

```
<configuration>
  <failOnError>false</failOnError>
</configuration>
```



The screenshot shows an IDE with a file named `pom.xml` open. The XML content is as follows:

```
25 |         <excludes>
26 |             |         <exclude>**/*.tmp</exclude>
27 |             |     </exclude>
28 |         </fileset>
29 |     </filesets>
30 |     <failOnError>false</failOnError>
31 |
32 <!-->
33 |
34 |         </plugin>
35 |     </plugins>
36 | </build>
37 </project>
38
```

Below the editor, a terminal window shows the execution of the command `mvn clean -Dmaven.clean.failOnError=false`. The output is:

```
palakkharbandas@ip-172-31-17-240:~/cms$ mvn clean -Dmaven.clean.failOnError=false
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
```

By following these steps, you have successfully explored the Maven Clean Plugin, created, and managed a target directory, configured the plugin to handle temporary files, and ensured a clean build environment. This process demonstrated the effective use of Maven commands and configurations to manage build artifacts and clean-up project directories.

