

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

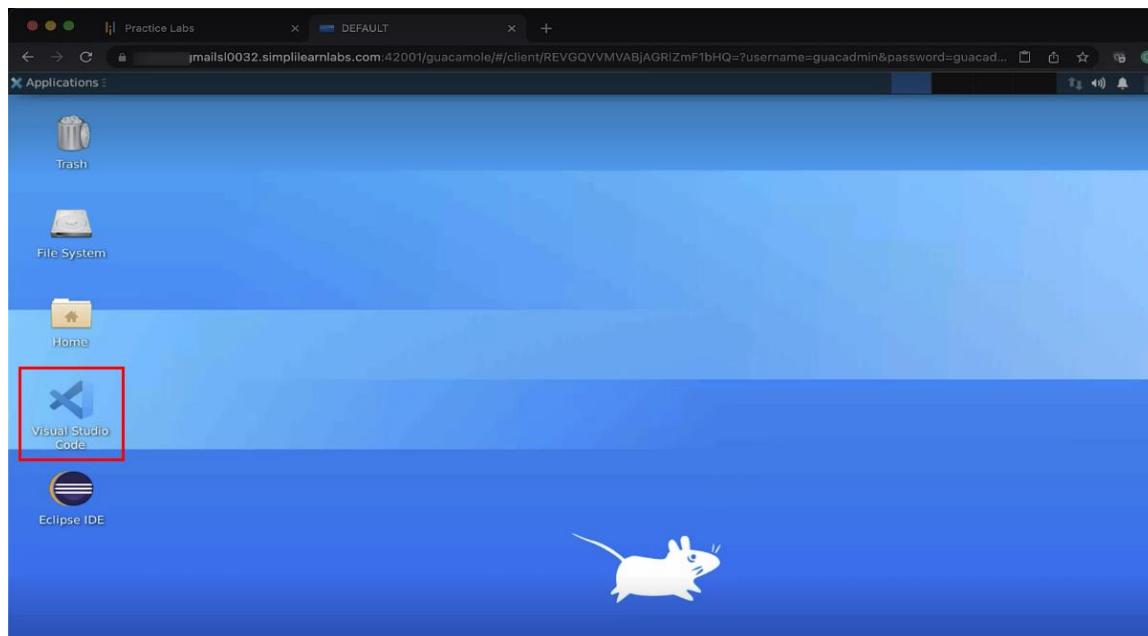
Prerequisites: None

Steps to be followed:

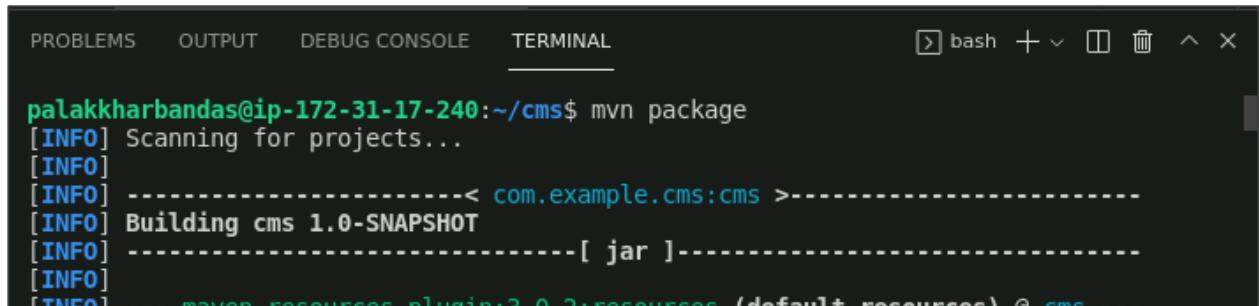
1. Create a target directory
2. Create a directory, temporary files, and logs
3. Handling Error

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



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

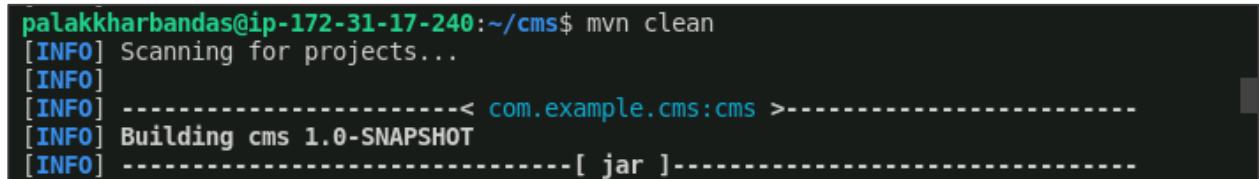


```
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ cms ---
[INFO] Building jar: /home/palakkharbandas/cms/target/cms-1.0-SNAPSHOT.jar
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 2.808 s
[INFO] Finished at: 2024-05-27T12:25:43Z
[INFO] -----
```

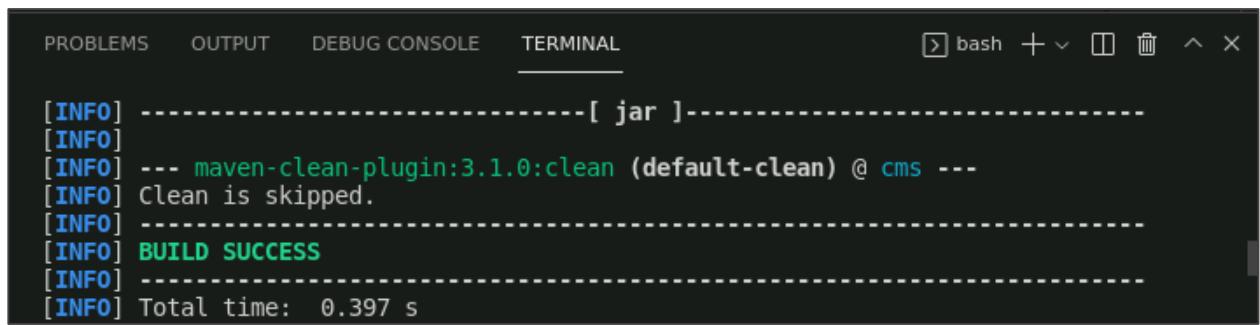
Note: Please refer to the previous demo on how to create the CMS project

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]
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
```



The screenshot shows a terminal window with the following output:

```
[INFO] -----[ jar ]-----
[INFO]
[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 a Visual Studio Code interface with the following details:

- Title Bar:** pom.xml - cms - Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer View:** Shows a tree structure under CMS:
 - logs
 - dev
 - dumps.tmp
 - prod
 - app.log

A file named pom.xml is selected in the tree.
- Editor View:** The main editor area displays the content of the pom.xml file. The XML code is as follows:

```
<?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>
```

- Bottom Status Bar:** Shows the command line output:

```
[**/*.tmp]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.464 s
[INFO] Finished at: 2024-05-27T12:41:00Z
[INFO]
```

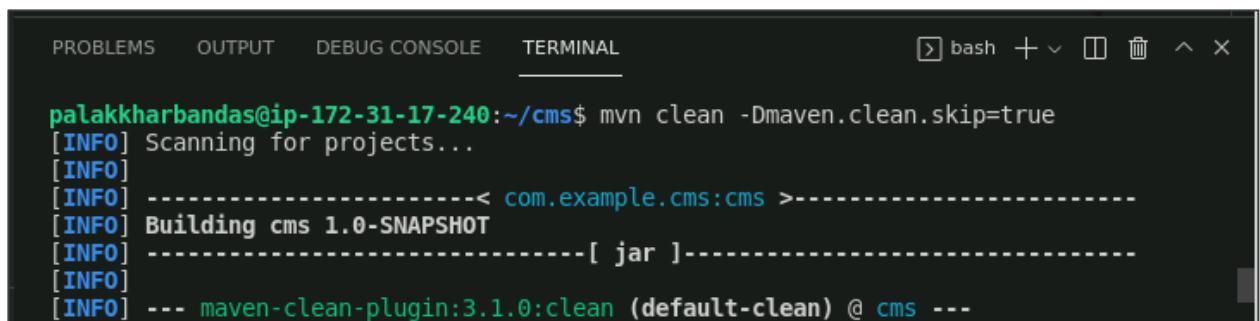
These lines of code are used in Maven to configure the behaviour of a plugin.

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 ]-----
```

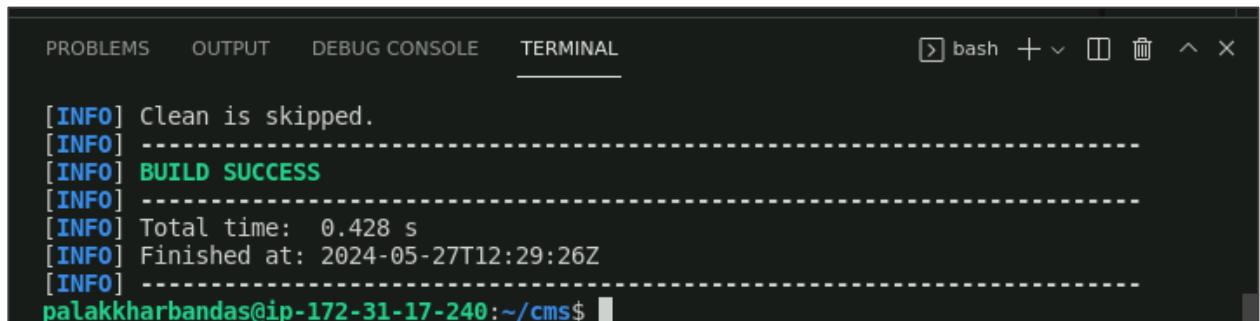
```
[INFO] -----[ jar ]-----  
[INFO]  
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---  
[INFO] Clean is skipped.  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 0.397 s
```

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



A screenshot of a terminal window titled "TERMINAL". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal content shows the command "mvn clean -Dmaven.clean.skip=true" being run. The output indicates that the clean step is skipped because the skip parameter is set to true. The build continues normally, creating a jar file and marking it as a success.

```
balakharbandas@ip-172-31-17-240:~/cms$ mvn clean -Dmaven.clean.skip=true
[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 ---
```



A screenshot of a terminal window titled "TERMINAL". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal content shows the command "mvn clean -Dmaven.clean.skip=true" being run. The output clearly states "[INFO] Clean is skipped." followed by "[INFO] BUILD SUCCESS". The build time and finish time are also displayed. The command prompt "palakharbandas@ip-172-31-17-240:~/cms\$" is visible at the bottom.

```
[INFO] Clean is skipped.
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  0.428 s
[INFO] Finished at: 2024-05-27T12:29:26Z
[INFO]
palakharbandas@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 a terminal window with the following output:

```
[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$ 
```

2.2 Create random files in both the directories, say **app.log** and **dumps.tmp**

The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows a project structure under 'CMS' named 'prod'. It includes 'logs', 'dev', 'app.log', 'dumps.tmp', and 'src'.
- Code Editor (Top Right):** Displays the contents of 'pom.xml' with the following code snippet:


```

<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>
```
- Terminal (Bottom Right):** Shows terminal output from a Linux command-line session:


```

[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$ touch logs/dev/app.log logs/dev/dumps.tmp log
s/prod/app.log
palakkharbandas@ip-172-31-17-240:~/cms$
```
- Bottom Navigation:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, along with a bash terminal icon.

2.3 To tell the **clean** command which files to clean and which files to keep, add the below code in the **pom.xml** file under **<configuration>**:

```

<configuration>
    <filesets>
        <fileset>
            <directory>logs/dev</directory>
            <includes>
                <include>**/*.log</include>
            </includes>
            <excludes>
                <exclude>**/*.tmp</exclude>
            </excludes>
        </fileset>
    </filesets>
</configuration>
```

```

<version>3.1.0</version>
<configuration>
<filesets>
<fileset>
<directory>logs/dev</directory>
<includes>
<include>**/*.log</include>
</includes>
<excludes>
<exclude>**/*.tmp</exclude>
</excludes>
</fileset>
</filesets>
</configuration>

```

[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\$ touch logs/dev/app.log logs/dev/dumps.tmp log
s/prod/app.log
palakkharbandas@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

```

<version>3.1.0</version>
<configuration>
<filesets>
<fileset>
<directory>logs/dev</directory>
<includes>
<include>**/*.log</include>
</includes>
<excludes>
<exclude>**/*.tmp</exclude>
</excludes>
</fileset>
</filesets>
</configuration>

```

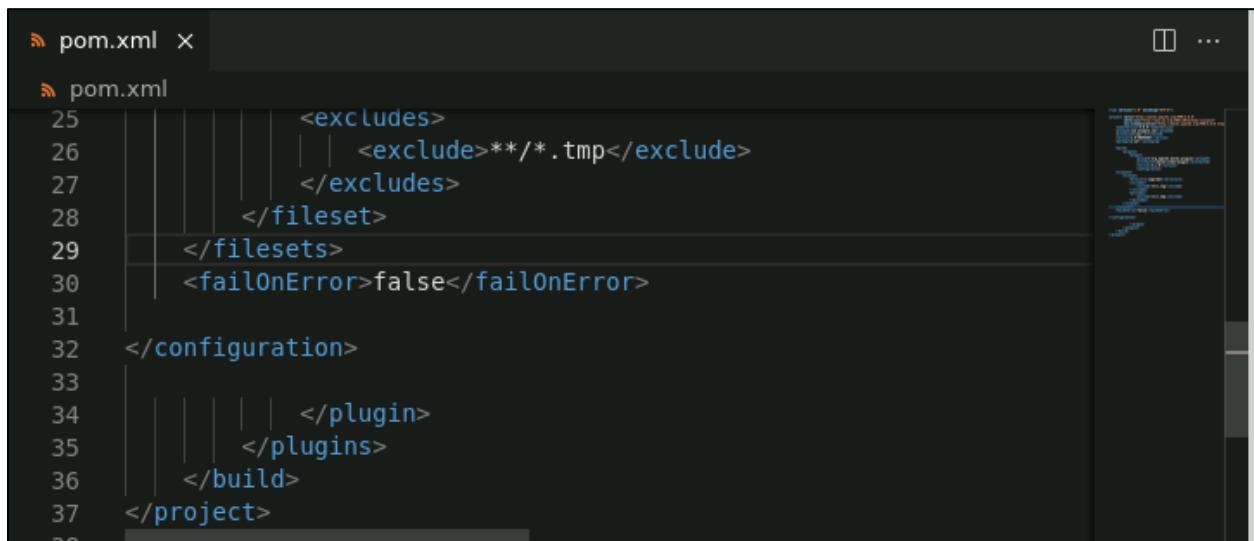
palakkharbandas@ip-172-31-17-240:~/cms\$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example.cms:cms >-----
[INFO] Building cms 1.0-SNAPSHOT
[INFO] -----[jar]-----
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ cms ---

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>
  <failOnErrors>false</failOnErrors>
</configuration>
```



The screenshot shows a code editor with a dark theme. A file named 'pom.xml' is open. The code contains the following XML configuration:

```
pom.xml
25   |     <excludes>
26   |       |     <exclude>**/*.tmp</exclude>
27   |     </excludes>
28   |   </filesets>
29   |   <failOnErrors>false</failOnErrors>
30
31 </configuration>
32
33   |     |     |   </plugin>
34   |     |     | </plugins>
35   |     |   </build>
36
37 </project>
38
```

Below the code editor, a terminal window shows the command being run and its output:

```
palakkharbandas@ip-172-31-17-240:~/cms$ mvn clean -Dmaven.clean.failOnErrors=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.

