



Spring web project

SREYA B

Creating Maven project with dependencies

Project
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Java** ☐ Kotlin ☐ Groovy
☒ **Maven**

Spring Boot
☐ 3.4.0 (SNAPSHOT) ☐ 3.4.0 (M1) ☐ 3.3.3 (SNAPSHOT) ☒ **3.3.2**
☐ 3.2.9 (SNAPSHOT) ☐ 3.2.8

Project Metadata

Group

com.demoproject

Artifact

demo

Name

demo

Description

Demo project for Spring Boot

Package name

com.demo

Packaging

☒ **Jar** ☐ War

Java

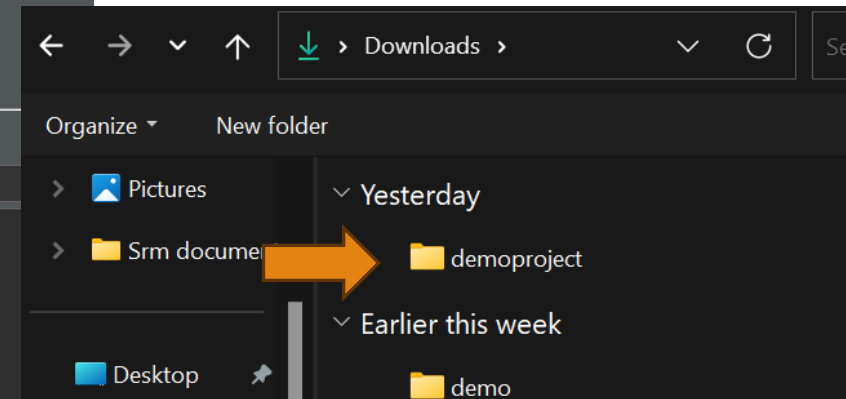
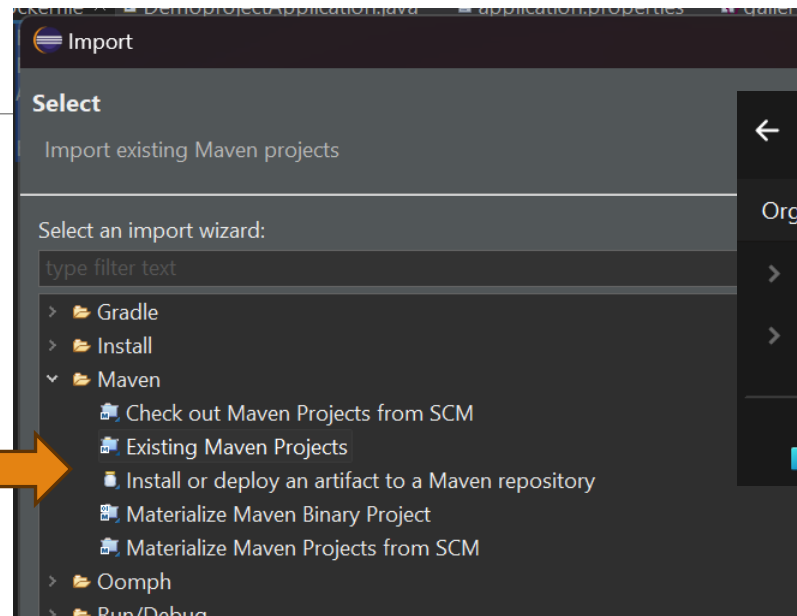
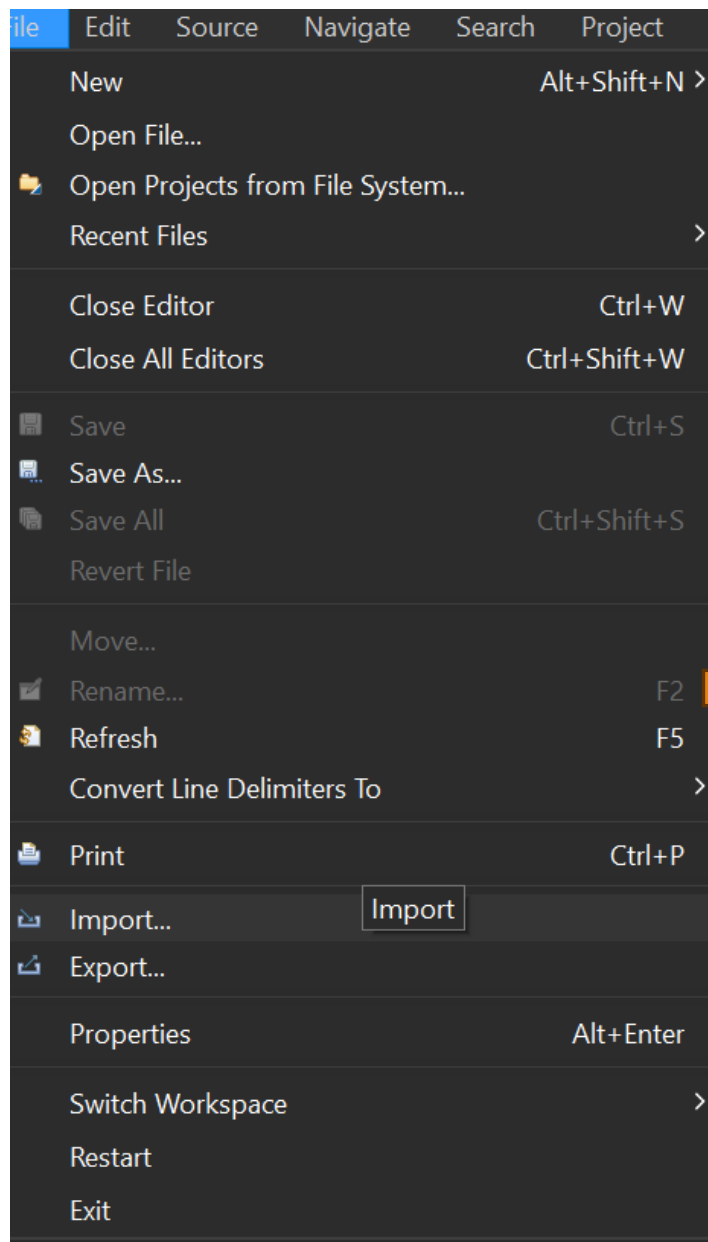
☐ 22 ☐ 21 ☒ **17**

Dependencies

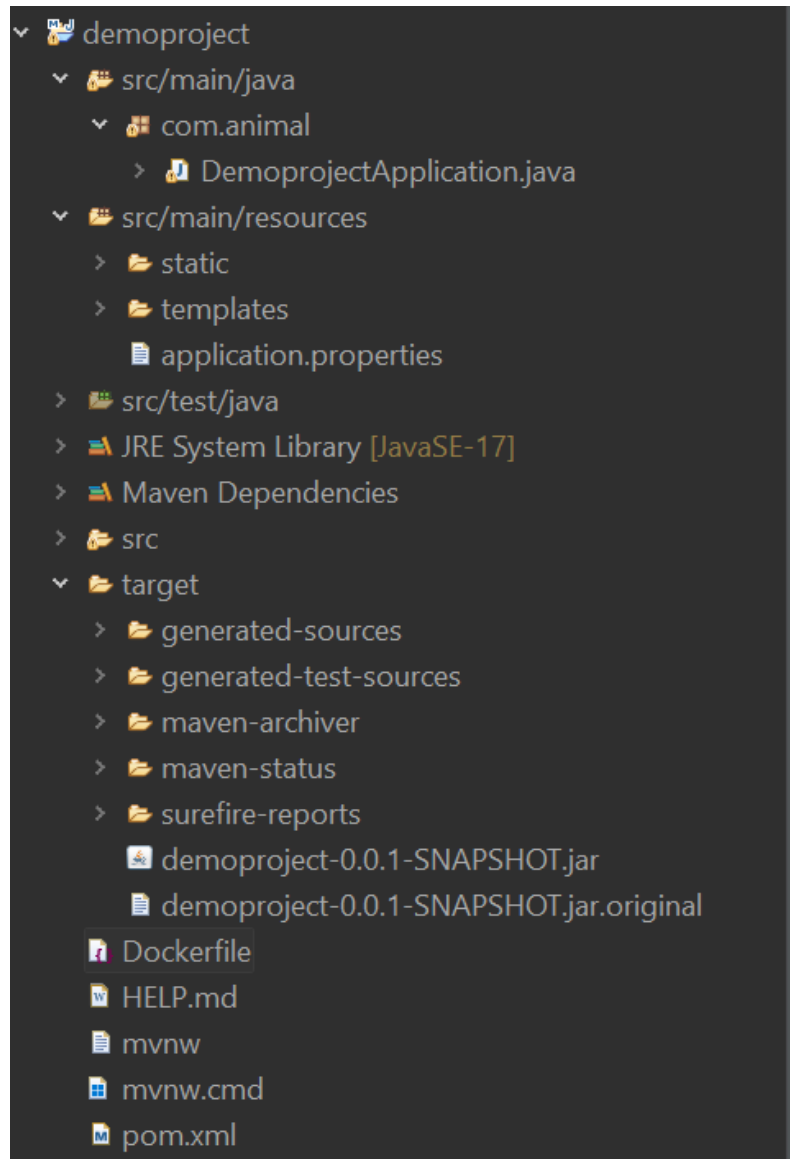
ADD DEPENDENCIES... CTRL + B

Spring Web **WEB**
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Thymeleaf **TEMPLATE ENGINES**
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.



Maven project directory



- **DemoprojectApplication.java** is the main application class, annotated with `@SpringBootApplication`.
- It serves as the entry point for your Spring Boot application. The main method within this class runs the application.
- Created static and templates folders in **src/test/resources**.
- Static folder contains another folder called Images.
- This folder holds Thymeleaf templates (.html files). These templates are rendered by the Spring Boot framework and presented as the UI.
- The application.properties file configures settings like server port and database connections for your Spring Boot application.

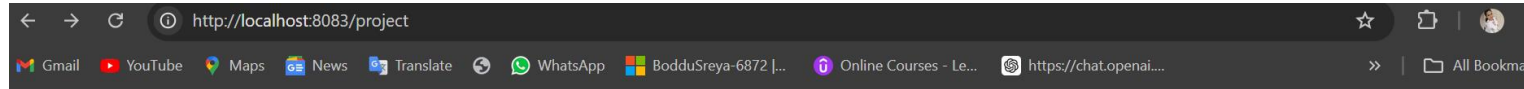
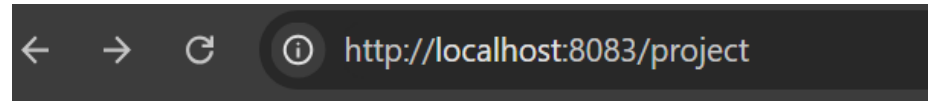
```

1 package com.animal;
2
3 import org.springframework.boot.SpringApplication;
4
5 @Controller
6 @SpringBootApplication
7 public class DemoprojectApplication {
8     @Controller
9     public static class AnimalController {
10
11         @GetMapping("/project")
12         public String showGallery() {
13             return "gallery";
14         }
15
16         @GetMapping("/animalDetail/{animal}")
17         public String showAnimalDetail(@PathVariable("animal") String animal, Model model) {
18             String animalName;
19             String animalDescription;
20             String animalImage;
21
22             switch (animal) {
23                 case "rhino":
24                     animalName = "Rhino";
25                     animalDescription = "Rhinoceroses are large, herbivorous mammals.";
26                     animalImage = "rhino.jpg";
27                     break;
28                 case "bear":
29                     animalName = "Bear";
30                     animalDescription = "Bears are large, omnivorous mammals.";
31                     animalImage = "bear.jpg";
32                     break;
33                 default:
34                     animalName = "Unknown";
35                     animalDescription = "No information available for this animal.";
36                     animalImage = "unknown.jpg";
37             }
38
39             model.addAttribute("animalName", animalName);
40             model.addAttribute("animalDescription", animalDescription);
41             model.addAttribute("animalImage", animalImage);
42
43             return "animalDetail";
44         }
45     }
46 }
47
48 public static void main(String[] args) {
49     SpringApplication.run(DemoprojectApplication.class, args);
50 }

```

[illegible]


```
DemoprojectApplication.java application.properties ×
1 spring.application.name=demoproject
2 server.port=8083
```



Animal Gallery



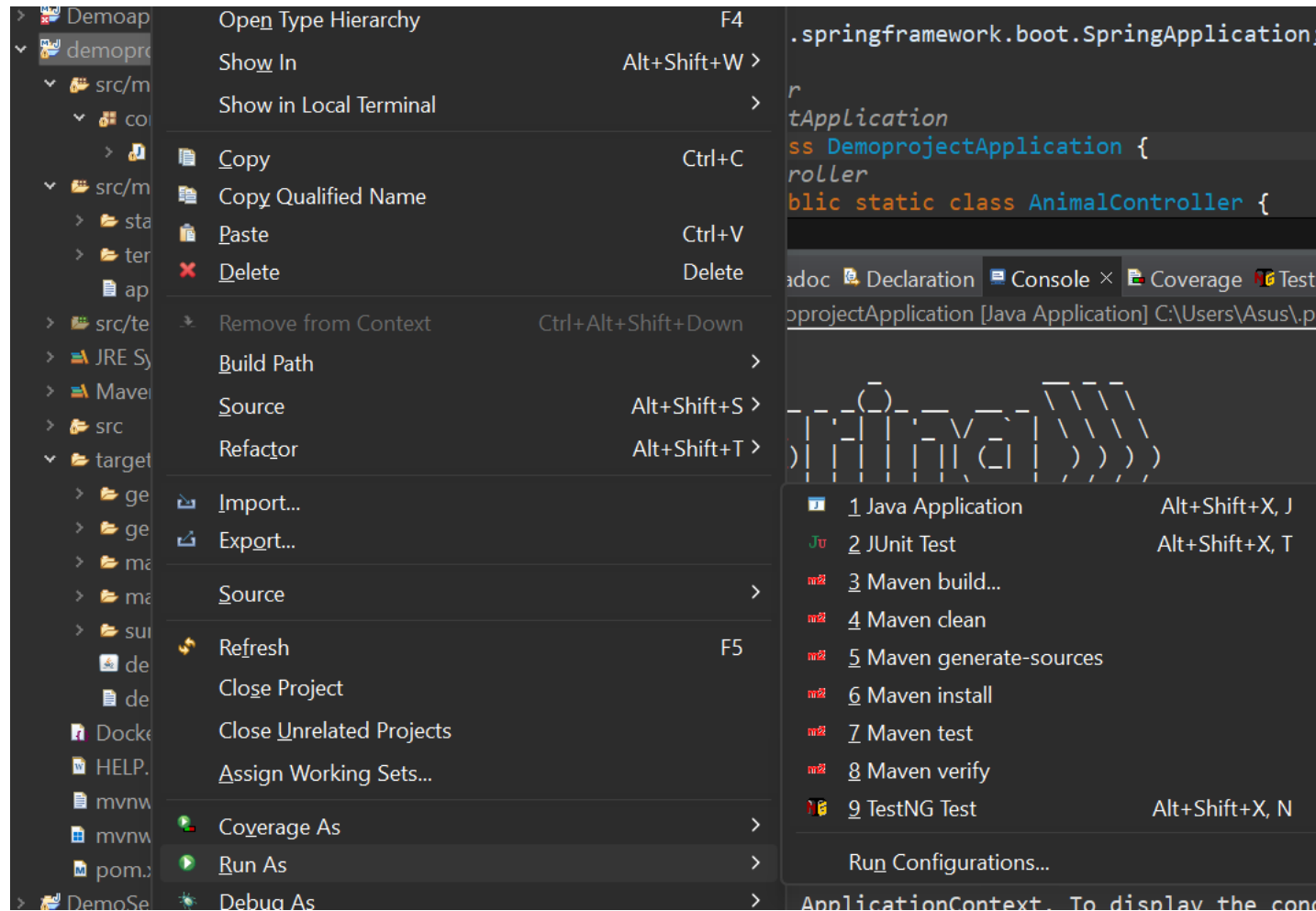
Lion



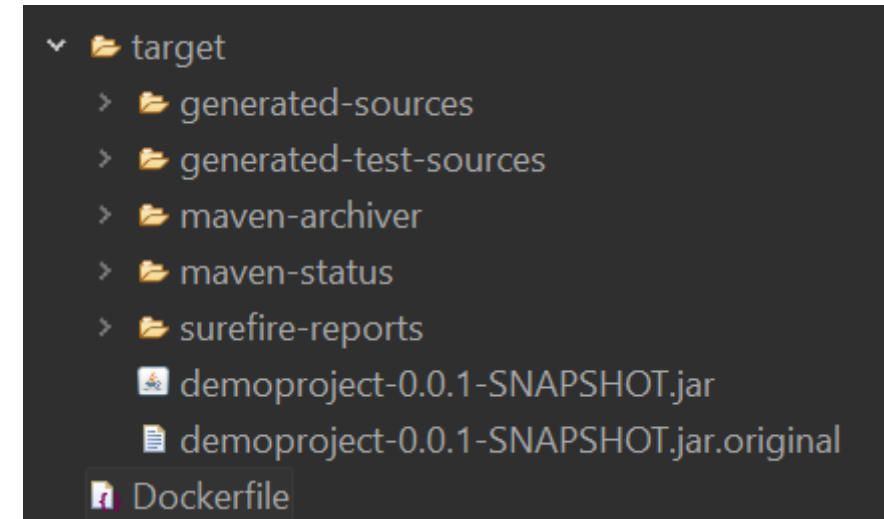
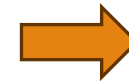
Lions are large carnivorous felines.

[Back to Gallery](#)

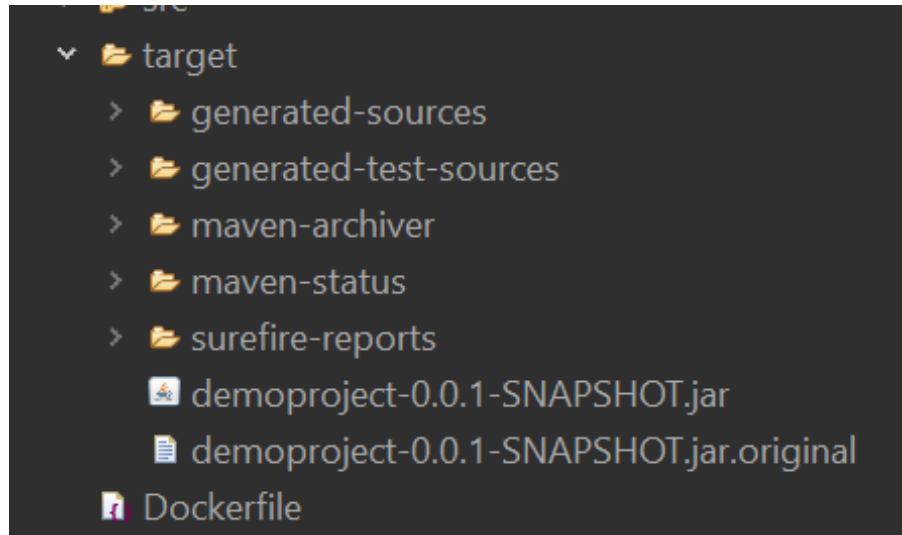
Creation of jar file



Demoproject -> Run as -> Maven install



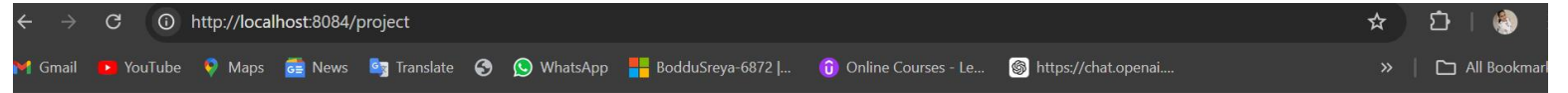
Creating a dockerfile



```
Dockerfile x DemoprojectApplication.java application.properties ga
1 From openjdk:21
2 Expose 8084
3 ADD target/demoproject-0.0.1-SNAPSHOT.jar app.jar
4
5 ENTRYPOINT ["java", "-jar", "app.jar"]
```


- Build docker image : `docker build -t app .`
- Run docker container : `docker run -p 8084:8083 app`

<http://localhost:8084/project>



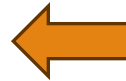
Animal Gallery

Lion



Lions are large carnivorous felines.

[Back to Gallery](#)



- Login to dockerhub
- Tag the docker image: `docker tag app sreya859/app:latest`
- Push the image to docker: `docker push sreya859/app:latest`

```
C:\Users\Asus\Downloads\demoproject\demoproject>docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
app                  latest       cd2e13f5a5ec   5 minutes ago  528MB
demoaqe              latest       9755ee008b17   5 minutes ago  528MB
demoque              latest       1e9ffd3b40e1   9 minutes ago  528MB
sreya859/demoaqe     latest       e6bb6e912b98   32 minutes ago 528MB

C:\Users\Asus\Downloads\demoproject\demoproject>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
a2a804a9d9a1   app      "java -jar app.jar"      9 seconds ago Up 9 seconds  8084/tcp, 0.0.0.0:8084->8083/tcp    app-container2

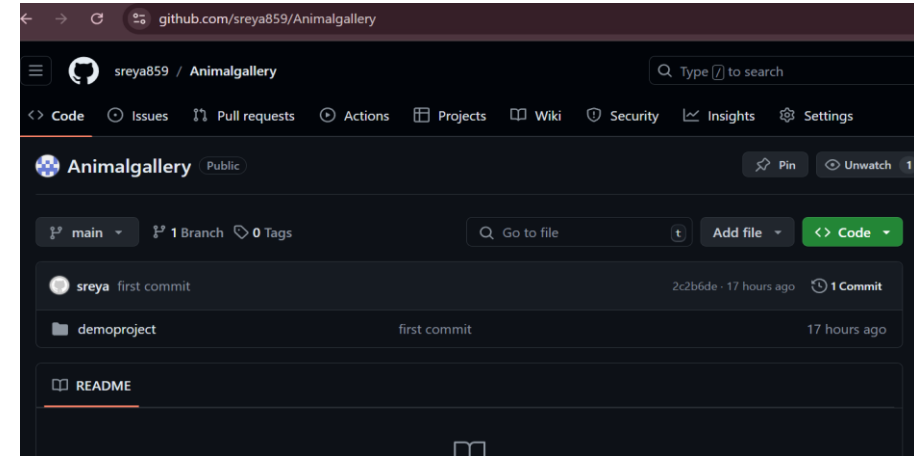
C:\Users\Asus\Downloads\demoproject\demoproject>docker tag app sreya859/app:latest

C:\Users\Asus\Downloads\demoproject\demoproject>docker push sreya859/app:latestest
The push refers to repository [docker.io/sreya859/app]
tag does not exist: sreya859/app:latestest

C:\Users\Asus\Downloads\demoproject\demoproject>docker push sreya859/app:latest
The push refers to repository [docker.io/sreya859/app]
2afe7bc0d8b5: Pushed
10359c5dc4ba: Mounted from sreya859/demoaqe
601b48657e0c: Mounted from sreya859/demoaqe
b42107e74152: Mounted from sreya859/demoaqe
latest: digest: sha256:99a4d2dae538adb8fee3393e5e20069d8ba0df45c6737f6b023dc9fb3bf51c86 size: 1166
```

Image pushed in docker

	Name	Tag	Status	Created	Size	Actions
	app	latest	In use	6 hours ago	528.08 MB	▶ ⋮ 🗑️
	sreya859/app	latest	In use	6 hours ago	528.08 MB	▶ ⋮ 🗑️



Jenkins pipeline

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.jar (157 kB at 1.0 MB/s)
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\NakshatraLibrary\pom.xml to
C:\WINDOWS\system32\config\systemprofile\.m2\repository\com\amdocs\demo\0.0.1-SNAPSHOT\demo-0.0.1-SNAPSHOT.pom
[INFO] Installing C:\ProgramData\Jenkins\.jenkins\workspace\NakshatraLibrary\target\demo-0.0.1-SNAPSHOT.jar to
C:\WINDOWS\system32\config\systemprofile\.m2\repository\com\amdocs\demo\0.0.1-SNAPSHOT\demo-0.0.1-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:27 min
[INFO] Finished at: 2024-08-01T11:07:25+05:30
[INFO] -----
Finished: SUCCESS
```



THANK YOU!