

FPT SOFTWARE QUY NHON AI Valley

# SPRING BOOT DevOps Tools

KienNCT3 - Mentor HaiNV21

# What is “DevOps” in context of Spring Boot

- In the context of Spring Boot, **DevOps** is a method of enhancing collaboration between software development (**Dev**) and IT operations (**Ops**) teams to automate and optimize the process of building, testing, deploying, and maintaining Spring Boot applications.
- DevOps emphasizes the integration of development, testing, and operations processes to deliver high-quality software quickly and reliably.
- Ways to apply DevOps to Spring Boot apps:

# Ways to apply **DevOps** to Spring Boot apps:

## 1) Continuous Integration (CI):

- DevOps encourages the use of CI tools like Jenkins, GitLab to automatically build and test Spring Boot apps every time source code changes are pushed to version repositories like Git.
- CI steps for Spring Boot applications typically include compiling source code, running unit tests, and creating executable files such as JAR files.

## 2) Continuous Deployment (CD):

- CD extends CI by automating the deployment process. DevOps typically involves deploying Spring Boot applications to different environments (development, testing, production) using automated scripting or deployment tools.
- CD steps for Spring Boot apps can include deploying executables to cloud platforms (AWS, Azure, Google Cloud), configuring database migrations, and updating configuration applications.

# Ways to apply **DevOps** to Spring Boot apps:

## 3) Infrastructure as Code (IaC)

- DevOps encourages the use of infrastructure-as-code tools like Terraform or AWS CloudFormation to create and manage the infrastructure needed to run Spring Boot applications.
- IaC allows teams to define infrastructure configurations in version management files, ensuring consistency across environments.

## 4) Automated Testing:

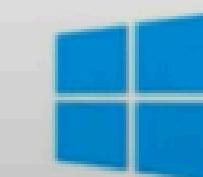
- DevOps encourages the use of automated testing practices to ensure the quality and reliability of Spring Boot applications.
- Automated testing includes unit tests, integration tests, and end-to-end tests, performed as part of CI pipelines to provide rapid feedback to developers.

# What is Docker?

When developing application, it usually needs ***belonging dependencies*** such as binary files, libraries, runtime environment...

- For example, Java -> JDK, Maven...; Angular -> NodeJS; C# -> .NET core, .Net Framework

Docker is a technology allows us to wrap the *application* and its *dependencies* into **one package**, which are **portable** (run anywhere) and **executable** (run anytime).



Mac OS

Application

Binary-Runtime-Library



# What is Docker *container, image*?

"Docker is a technology allows us to wrap the *application* and its *dependencies* into **one package**, which are **portable** (run anywhere) and **executable** (run anytime)"

- Those packages are called **images**.
- When executing an image, we get a **container**.
- Containers functions like a virtual machine with fully provided features such as file system, network interface, process tree...
- In the end, they are just **processes** running on the OS and managed by Docker.

# Basic commands of Docker

## Syntax

docker <component> <command>

### Components:

- image
- container
- network
- volume
- ...

### Commands:

- ls: list
- run
- exec
- stop
- pull
- prune
- ...

# Basic commands of Docker: image

`docker image pull <image>`

`docker image pull <image>:<tag>`

`docker image push <image>:<tag>`

`docker image ls | docker images`

`docker image prune`

## Short-hand:

`docker pull`

`docker push`

# Basic commands of Docker: container

`docker container run <image>`

`docker container ls | docker container ls -a`

`docker ps | docker ps -a`

`docker container stop <container_id>`

C:\Users\TRUNG KIEN>`docker rm -f 5bc3`  
5bc3

`docker container prune`

`docker container exec <container_id> <command>`

## Short-hand:

`docker run`

`docker stop`

`docker exec`

# Example of Docker commands:

```
PS C:\Users\TRUNG KIEN> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
springboot-docker.jar	latest	f829daca654c	26 hours ago	517MB
springboot-devops.jar	latest	88b7f4153693	30 hours ago	517MB
springboot_devops	1.0	3e0500be49c0	2 days ago	511MB
spring-boot-docker.jar	latest	89340ffde611	3 days ago	493MB
my-backend-app	1.0	901ac946368d	5 days ago	491MB
my_docker_file	1.0	0edf4c288264	5 days ago	491MB
paketobuildpacks/run-jammy-base	latest	530c11846f48	9 days ago	106MB
postgres	latest	b9390dd1ea18	5 weeks ago	431MB
gcr.io/k8s-minikube/kicbase	v0.0.42	dbc648475405	4 months ago	1.2GB
docker/getting-started	latest	3e4394f6b72f	15 months ago	47MB
spring-jpa	latest	d3526f892d82	44 years ago	328MB
paketobuildpacks/builder-jammy-base	latest	f5b0161594a1	44 years ago	1.42GB

```
PS C:\Users\TRUNG KIEN> docker ps
```

```
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

```
PS C:\Users\TRUNG KIEN> docker ps -a
```

```
CONTAINER ID IMAGE
```

```
PORTS NAMES
```

```
f2b81637e482  postgres:latest
```

```
    dockercompose-postgres-1
```

```
28aacbe0cbfc  spring-jpa
```

```
    clever_greider
```

```
016b0273a95a  efddfb330b44
```

```
    sweet_bose
```

```
3a3ad9a4ea3a  springboot-docker.jar
```

```
    clever_meitner
```

```
f5138edd88e0  springboot-devops.jar
```

```
    stupefied_proskuriakova
```

```
ee50c62b6b3c  gcr.io/k8s-minikube/kicbase:v0.0.42
```

```
    minikube
```

```
2c0e83fdc890  spring-boot-docker.jar
```

```
    sharp_mclean
```

```
59efaba82800  docker/getting-started
```

```
    dreamy_cohen
```

COMMAND	CREATED	STATUS
"docker-entrypoint.s..."	25 hours ago	Exited (0) 22 hours ago
"/cnb/process/web"	25 hours ago	Exited (1) 25 hours ago
"/cnb/process/web"	25 hours ago	Exited (1) 25 hours ago
"java -jar /springbo..."	26 hours ago	Exited (143) 25 hours ago
"java -jar /springbo..."	30 hours ago	Exited (130) 30 hours ago
"/usr/local/bin/entr..."	46 hours ago	Exited (137) 43 hours ago
"java -jar /spring-b..."	3 days ago	Exited (143) 3 days ago
"/docker-entrypoint..."	9 days ago	Exited (0) 9 days ago

# Example of Docker commands:

```
PS C:\Users\TRUNG KIEN> docker container prune
```

WARNING! This will remove all stopped containers.

Are you sure you want to continue? [y/N] y

Deleted Containers:

```
f2b81637e4827a9f82501db59441146b123009919044de1b05930904ac7aad12  
28aacbe0cbfc9363723be8649a8688237c439a0bd810839cb5bbfd2959a68ac  
016b0273a95a0798a6db0a711f53477c7e043724e85e1cb36f45477b415267ad  
3a3ad9a4ea3a1f04b91fc6613acc8e13cf614453ae6299d59973cd35850165f3  
f5138edd88e07d94b549d11fba2299671535bab518511c9bef200177245121ac  
ee50c62b6b3caede932ba21b7230148f3546b9197665804356da3e620a73716d  
2c0e83fdc8904fd9ac077bc371a34b6e5e0e45de35df09a10a1dccca5cf9f2ff  
59efaba828007dcf421d39dd9f8bca3ecb9229dd359af48360ef997e28e7baa1
```

Total reclaimed space: 5.063MB

```
C:\Users\TRUNG KIEN>docker ps -qa  
7186c46693fa  
e5918f760d19  
d477272ffb7a  
2ae60774d54e  
dcbb00b57afb  
be5c907e1127  
04cebd4cdae  
b4f5bcf4dea0  
2c12a3a3d4c6  
a402e17da32e  
54422fb2d94a  
e602660c7682
```

```
PS C:\Users\TRUNG KIEN> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

How to keep session when deploy app to docker?

# Dockerfile in Docker:

Dockerfile is the template that allows us to instruct Docker to build our image step by step, for example:

```
Dockerfile ×  
1 FROM openjdk:17-alpine3.14  
2 LABEL autho  
3 EXPOSE 8080  
4 ADD target/  
5 ENTRYPPOINT
```

```
FROM openjdk:17-alpine3.14  
WORKDIR /application  
COPY build/libs/awesome-app-1.0.jar ./  
CMD ["java", "-jar", "awesome-app-1.0.jar"]
```

Line 1: This line specifies the base image for our Docker container.

Line 3: This line indicates that the container will listen on port 8080. It's just a hint for anyone running the container to know which port to expose.

Line 4: The **ADD** instruction copies the Spring Boot JAR file (located at target/springboot-devops.jar) into the container.

Line 5: The **ENTRYPOINT** specifies the command to run when the container starts. In this case, it runs the Spring Boot application by executing the JAR file (**springboot-devops.jar**) using the Java runtime (java -jar).

## Syntax

- docker build -t <image\_name>:<tag>

The difference between ADD and COPY in dockerfile

# Dockerfile in Docker:

```
PS C:\SpringBoot_DevOps> docker build -t my-app.jar:1.0 .
[+] Building 5.0s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 189B
=> [internal] load metadata for docker.io/library/openjdk:17
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 46.02MB
=> CACHED [1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb7d8
=> [2/2] ADD target/springboot-docker.jar my-app.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:8564573cc7832a0760d8da293542173d8bb5864cb6f3ce24595ec02e5932ab42
=> => naming to docker.io/library/my-app.jar:1.0
```

```
PS C:\SpringBoot_DevOps> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
my-app.jar          1.0      8564573cc783   5 seconds ago  517MB
```

```
PS C:\SpringBoot_DevOps> docker build -t my-app .  
[+] Building 5.0s (8/8) FINISHED  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 189B  
=> [internal] load metadata for docker.io/library/openjdk:11-jdk-alpine  
=> [auth] library/openjdk:pull token for registry.docker.io  
=> [internal] load .dockerignore  
=> => transferring context: 2B  
=> [internal] load build context  
=> => transferring context: 46.02MB  
=> CACHED [1/2] FROM docker.io/library/openjdk:11-jdk-alpine  
=> [2/2] ADD target/springboot-docker.jar my-app.jar  
=> exporting to image  
=> => exporting layers  
=> => writing image sha256:8564573cc7832a0760c3f3e03333333  
=> => naming to docker.io/library/my-app.jar:11-jdk-alpine
```

```
PS C:\SpringBoot_DevOps> docker run -p 9090:8080 springboot-devops.jar

.
-----
/\ / _ _ - _(_)_ _ _ \ \ \
( ( )\__| |_ | |_ | | \_` | \ \ \
\W _ _)| |_)| | | | | | (| | ) ) )
' |____| .__|_|_|_|_|_\_, | / / /
=====|_|=====|_|=/_/_/_/
:: Spring Boot ::          (v3.2.4)

2024-03-26T22:43:17.683Z INFO 1 --- [SpringBoot_DevOps] [           main] o.k.s.SpringBootDevOpsApplication      : Starting SpringBootDevOpsApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 1 (/springboot-devops.jar) started by root in /
2024-03-26T22:43:17.703Z INFO 1 --- [SpringBoot_DevOps] [           main] o.k.s.SpringBootDevOpsApplication      : No active profile set, falling back to default profile: "default"
2024-03-26T22:43:19.044Z INFO 1 --- [SpringBoot_DevOps] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-03-26T22:43:19.061Z INFO 1 --- [SpringBoot_DevOps] [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-03-26T22:43:19.062Z INFO 1 --- [SpringBoot_DevOps] [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-03-26T22:43:19.106Z INFO 1 --- [SpringBoot_DevOps] [           main] o.a.c.c.C.[Tomcat].[localhost].[]       : Initializing Spring embedded WebApplication context
2024-03-26T22:43:19.108Z INFO 1 --- [SpringBoot_DevOps] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1253 ms
2024-03-26T22:43:19.238Z INFO 1 --- [SpringBoot_DevOps] [           main] o.s.b.a.w.s.WelcomePageHandlerMapping  : Adding welcome page template: index
2024-03-26T22:43:19.553Z INFO 1 --- [SpringBoot_DevOps] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
```

# Docker-compose

- Docker Compose is a tool used to manage and deploy multi-container applications in Docker.
- It allows us to use a simple configuration file to define services, their associated configurations, and how they interact with each other, instead of having to execute many individual Docker commands.

```
docker-compose.yml ×  
1 version: "3.8"  
2 ➤ services:  
3 ➤   springboot-app:  
4     image: springboot-app  
5     build: .  
6   ports:  
7     - 8080:8080
```

```
PS C:\SpringBoot_DevOps> docker-compose up --build  
[+] Building 2.5s (8/8) FINISHED  
=> [springboot-app internal] load build definition from Dockerfile  
=> => transferring dockerfile: 189B  
=> [springboot-app internal] load metadata for docker.io/library/openjdk:17  
=> [springboot-app auth] library/openjdk:pull token for registry-1.docker.io  
=> [springboot-app internal] load .dockerrcignore  
=> => transferring context: 2B  
=> [springboot-app internal] load build context  
=> => transferring context: 78B  
=> [springboot-app 1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbbaef9f87662e7b3f38cb7d8  
=> CACHED [springboot-app 2/2] ADD target/springboot-docker.jar my-app.jar  
=> [springboot-app] exporting to image  
=> => exporting layers  
=> => writing image sha256:b802441a4263c2f55658bf766d83524aee0d0af5311af1c17379bc0e7770f8af  
=> => naming to docker.io/library/springboot-app  
[+] Running 1/1  
✓ Container springboot_devops-springboot-app-1 Recreated  
Attaching to springboot-app-1  
Gracefully stopping... (press Ctrl+C again to force)  
Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:8080 -> 0.0.0.0:8080: listen tcp 0.0.0.0:8080: bind: Only one usage of each socket address (protocol/network address/port) is normally permitted.
```

Check result:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
springboot-app	latest	b802441a4263	4 days ago	517MB

# Docker-compose

```
docker-compose.yml ×  
1 version: "3.8"  
2 services:  
3   springboot-app:  
4     container_name: springboot-app  
5     image: springboot-app  
6     build: .  
7     ports:  
8       - 8080:8080
```

```
PS C:\SpringBoot_DevOps> docker-compose up -d --build  
[+] Building 0.0s (0/0) docker:default  
[+] Building 2.1s (8/8) FINISHED  
=> [springboot-app internal] load build definition from Dockerfile  
=> => transferring dockerfile: 189B  
=> [springboot-app internal] load metadata for docker.io/library/openjdk:17  
=> [springboot-app auth] library/openjdk:pull token for registry-1.docker.io  
=> [springboot-app internal] load .dockerignore  
=> => transferring context: 2B  
=> [springboot-app internal] load build context  
=> => transferring context: 78B  
=> [springboot-app 1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaeaf9f87662e7b3f38cb7d8  
=> CACHED [springboot-app 2/2] ADD target/springboot-docker.jar my-app.jar  
=> [springboot-app] exporting to image  
=> => exporting layers  
=> => writing image sha256:b802441a4263c2f55658bf766d83524aee0d0af5311af1c17379bc0e7770f8af  
=> => naming to docker.io/library/springboot-app  
[+] Running 1/2  
- Container springboot_devops-springboot-app-1 Recreated  
✓ Container springboot-app Started
```

Check result:

```
PS C:\SpringBoot_DevOps> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
0d2df0718ef2 springboot-app "java -jar /my-app.j..." 5 seconds ago Up 5 seconds 0.0.0.0:8080->8080/tcp springboot-app
```

# Docker-compose

```
docker-compose.yml ×  
1 version: "3.8"  
2 services:  
3   springboot-app:  
4     container_name: springboot-app  
5     image: springboot-app  
6     build: .  
7     ports:  
8       - 8080:8080
```

```
PS C:\SpringBoot_DevOps> docker-compose up -d --build  
[+] Building 0.0s (0/0) docker:default  
[+] Building 2.1s (8/8) FINISHED  
=> [springboot-app internal] load build definition from Dockerfile  
=> => transferring dockerfile: 189B  
=> [springboot-app internal] load metadata for docker.io/library/openjdk:17  
=> [springboot-app auth] library/openjdk:pull token for registry-1.docker.io  
=> [springboot-app internal] load .dockerignore  
=> => transferring context: 2B  
=> [springboot-app internal] load build context  
=> => transferring context: 78B  
=> [springboot-app 1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb7d8  
=> CACHED [springboot-app 2/2] ADD target/springboot-docker.jar my-app.jar  
=> [springboot-app] exporting to image  
=> => exporting layers  
=> => writing image sha256:b802441a4263c2f55658bf766d83524aee0d0af5311af1c17379bc0e7770f8af  
=> => naming to docker.io/library/springboot-app  
[+] Running 1/2  
- Container springboot_devops-springboot-app-1 Recreated  
✓ Container springboot-app Started
```

Check result:

```
PS C:\SpringBoot_DevOps> docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
0d2df0718ef2 springboot-app "java -jar /my-app.j..." 5 seconds ago Up 5 seconds 0.0.0.0:8080->8080/tcp springboot-app
```

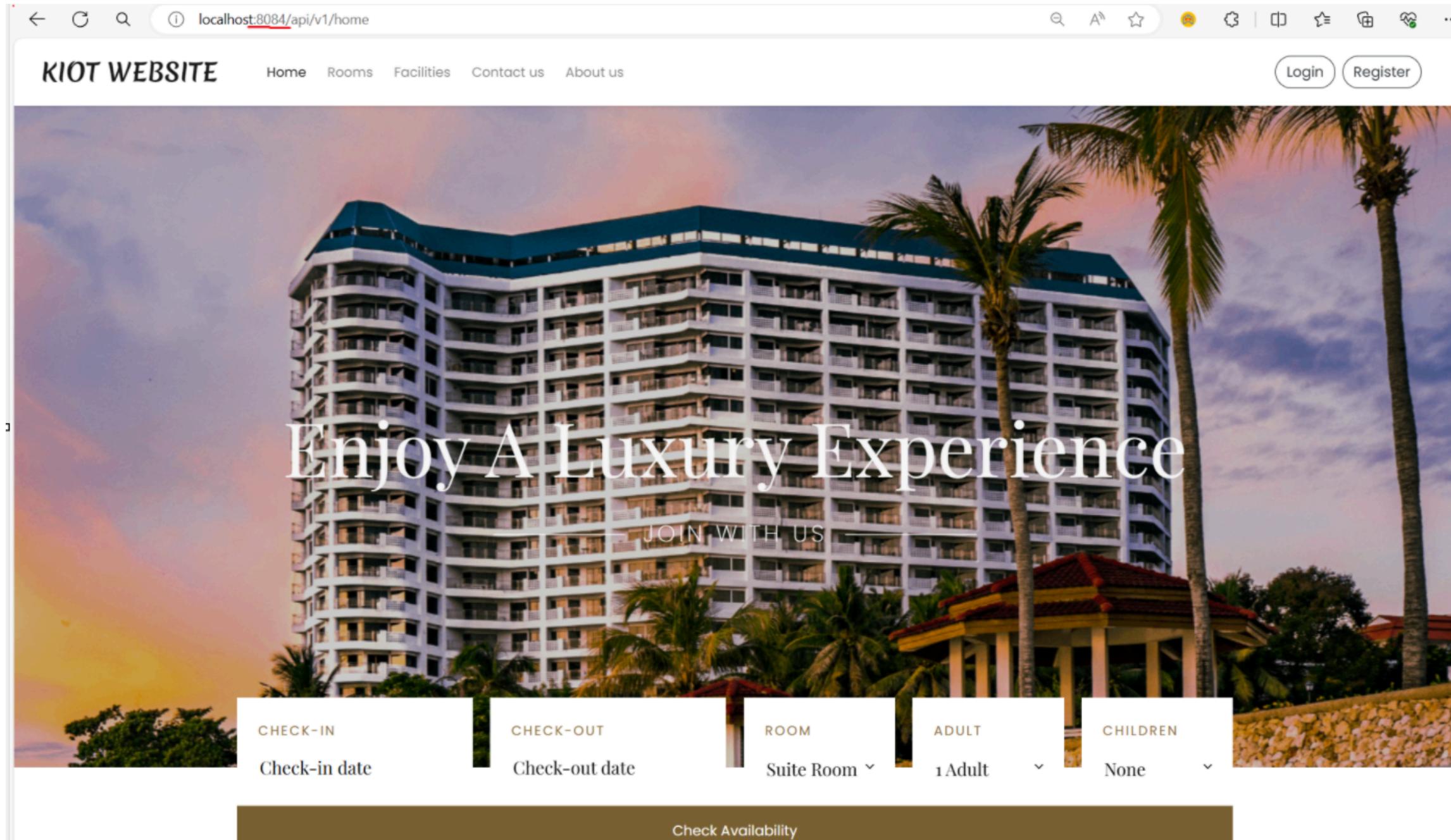
To remove docker container, we can use this command:

```
PS C:\SpringBoot_DevOps> docker-compose down  
[+] Running 2/2  
✓ Container springboot-app Removed  
✓ Network springboot_devops_default Removed
```

```
+ docker-compose.yml ×  
1 version: "3.8"  
2 ➤ services:  
3 ▶  springboot-app:  
4     container_name: springboot-app  
5     image: springboot-app  
6     build: .  
7     ports:  
8       - 8084:8080
```

```
PS C:\SpringBoot_DevOps> docker-compose up --build  
2024/04/02 10:44:25 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed  
[+] Building 1.2s (7/7) FINISHED  
=> [springboot-app internal] load build definition from Dockerfile  
=> => transferring dockerfile: 189B  
=> [springboot-app internal] load metadata for docker.io/library/openjdk:17  
=> [springboot-app internal] load .dockerignore  
=> => transferring context: 2B  
=> [springboot-app internal] load build context  
=> => transferring context: 78B  
=> [springboot-app 1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb7d8  
=> CACHED [springboot-app 2/2] ADD target/springboot-docker.jar my-app.jar  
=> [springboot-app] exporting to image  
=> => exporting layers  
=> => writing image sha256:b802441a4263c2f55658bf766d83524aee0d0af5311af1c17379bc0e7770f8af  
=> => naming to docker.io/library/springboot-app  
[+] Running 1/1  
✓ Container springboot-app Recreated
```

Check result:



# Running Spring Boot Applications With Minikube

- **Kubernetes** is an open-source system developed by Google (also called K8s), used to automatically deploy, manage, and scale container applications.
  - It provides a platform for managing and arrange containers and services across a **distributed** network.
- 
- **Minikube** is a tool that helps deploy a simple Kubernetes cluster on localhost or without to deploy a complex Kubernetes cluster on a production environment.



# Running Spring Boot Applications With Minikube

```
C:\Users\TRUNG KIEN>minikube version  
minikube version: v1.32.0  
commit: 8220a6eb95f0a4d75f7f2d7b14cef975f050512d
```

Check version of Minikube

To start a cluster, make sure Docker desktop is running, start from administrator terminal and run:

```
C:\Users\TRUNG KIEN>minikube start --driver=docker  
* minikube v1.32.0 on Microsoft Windows 11 Home Single Language 10.0.22631.3296 Build 22631.3296  
* Using the docker driver based on existing profile  
* Starting control plane node minikube in cluster minikube  
* Pulling base image ...  
* Updating the running docker "minikube" container ...  
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...  
* Verifying Kubernetes components...  
  - Using image docker.io/kubernetesui/dashboard:v2.7.0  
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8  
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5  
* Some dashboard features require the metrics-server addon. To enable all features please run:  
  
  minikube addons enable metrics-server  
  
* Enabled addons: storage-provisioner, default-storageclass, dashboard  
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Check minikube in docker desktop:

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started
<input type="checkbox"/>	<a href="#">minikube</a>	<a href="#">gcr.io/k8s-minikube/kicbase:v0.0.42</a>	Running	199.07%		44 seconds ago

# Running Spring Boot Applications With Minikube

Check minikube status:

```
C:\Users\TRUNG KIEN>minikube status  
minikube  
type: Control Plane  
host: Running  
kubelet: Running  
apiserver: Running  
kubeconfig: Configured
```

Check cluster info:

```
C:\Users\TRUNG KIEN>kubectl cluster-info  
Kubernetes control plane is running at https://127.0.0.1:51143  
CoreDNS is running at https://127.0.0.1:51143/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy  
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

# Running Spring Boot Applications With Minikube

Check node:

```
C:\Users\TRUNG KIEN>kubectl get node
NAME      STATUS    ROLES      AGE      VERSION
minikube  Ready     control-plane  3d22h   v1.28.3
```

Connect Docker client to Docker server managed by Minikube

```
C:\Users\TRUNG KIEN>minikube docker-env
SET DOCKER_TLS_VERIFY=1
SET DOCKER_HOST=tcp://127.0.0.1:51140
SET DOCKER_CERT_PATH=C:\Users\TRUNG KIEN\.minikube\certs
SET MINIKUBE_ACTIVE_DOCKERD=minikube
REM To point your shell to minikube's docker-daemon, run:
REM @FOR /f "tokens=*" %i IN ('minikube -p minikube docker-env --shell cmd') DO @%i

C:\Users\TRUNG KIEN> @FOR /f "tokens=*" %i IN ('minikube -p minikube docker-env --shell cmd') DO @%i
```

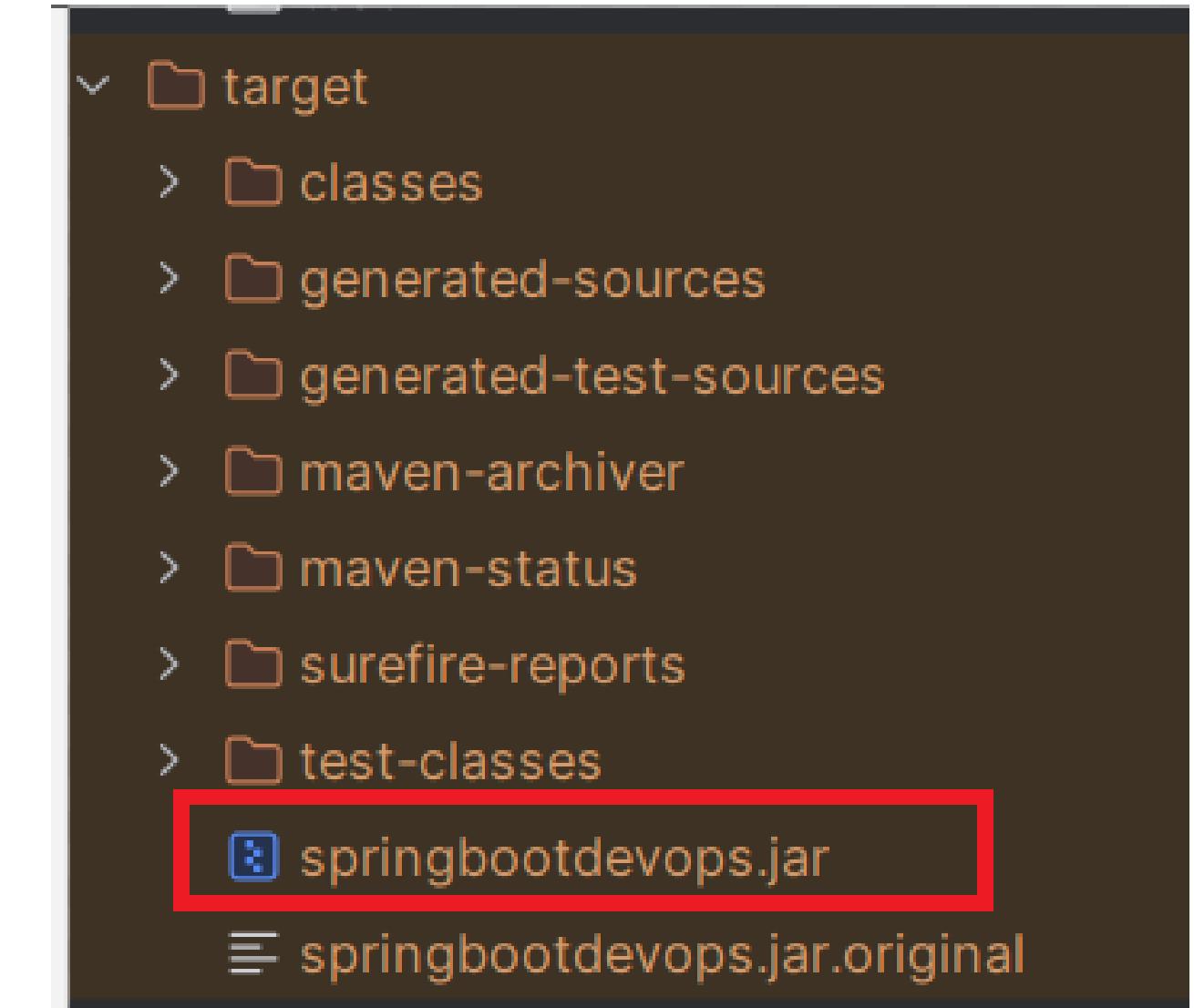
# Running Spring Boot Applications With Minikube

Assume we have Dockerfile:

```
Dockerfile
1 FROM openjdk:17
2 LABEL authors="TRUNG KIEN"
3 EXPOSE 8080
4 ADD target/springbootdevops.jar springbootdevops.jar
5 ENTRYPOINT ["java", "-jar", "/springbootdevops.jar"]
```

C:\Users\TRUNG KIEN>docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
registry.k8s.io/kube-apiserver	v1.28.3	537434729123	5 months ago	126MB
registry.k8s.io/kube-controller-manager	v1.28.3	10baa1ca1706	5 months ago	122MB
registry.k8s.io/kube-scheduler	v1.28.3	6d1b4fd1b182	5 months ago	60.1MB
registry.k8s.io/kube-proxy	v1.28.3	bfc896cf80fb	5 months ago	73.1MB
registry.k8s.io/etcd	3.5.9-0	73deb9a3f702	10 months ago	294MB
registry.k8s.io/coredns/coredns	v1.10.1	ead0a4a53df8	13 months ago	53.6MB
registry.k8s.io/pause	3.9	e6f181688397	17 months ago	744kB
kubernetesui/dashboard	<none>	07655ddf2eeb	18 months ago	246MB
kubernetesui/metrics-scrapers	<none>	115053965e86	22 months ago	43.8MB
gcr.io/k8s-minikube/storage-provisioner	v5	6e38f40d628d	2 years ago	31.5MB



```
C:\Users\TRUNG KIEN>cd C:\SpringBoot_DevOps
```

```
C:\SpringBoot_DevOps>docker build -t springbootdevops:1.0 .
```

[+]	Building 92.3s (7/7) FINISHED	docker:default
=>	[internal] load .dockerignore	0.0s
=>	=> transferring context: 2B	0.0s
=>	[internal] load build definition from Dockerfile	0.0s
=>	=> transferring dockerfile: 208B	0.0s
=>	[internal] load metadata for docker.io/library/openjdk:17	1.3s
=>	[internal] load build context	1.5s
=>	=> transferring context: 39.60MB	1.5s
=>	[1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb	89.4s
=>	=> resolve docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb7	0.1s
=>	=> sha256:38a980f2cc8accf69c23deae6743d42a87eb34a54f02396f3fcfd7c2d06e2c5b 42.11MB / 42.11MB	38.5s
=>	=> sha256:de849f1cfbe60b1c06a1db83a3129ab0ea397c4852b98e3e4300b12ee57ba111 13.53MB / 13.53MB	11.6s
=>	=> sha256:a7203ca35e75e068651c9907d659adc721dba823441b78639fde66fc988f042f 187.53MB / 187.53MB	82.6s
=>	=> sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb7d8 1.04kB / 1.04kB	0.0s
=>	=> sha256:98f0304b3a3b7c12ce641177a99d1f3be56f532473a528fda38d53d519caf13 954B / 954B	0.0s
=>	=> sha256:5e28ba2b4cdb3a7c3bd0ee2e635a5f6481682b77eabf8b51a17ea8bfelc05697 4.45kB / 4.45kB	0.0s
=>	=> extracting sha256:38a980f2cc8accf69c23deae6743d42a87eb34a54f02396f3fcfd7c2d06e2c5b	4.5s
=>	=> extracting sha256:de849f1cfbe60b1c06a1db83a3129ab0ea397c4852b98e3e4300b12ee57ba111	1.1s
=>	=> extracting sha256:a7203ca35e75e068651c9907d659adc721dba823441b78639fde66fc988f042f	6.3s
=>	[2/2] ADD target/springbootdevops.jar springbootdevops.jar	1.2s
=>	exporting to image	0.3s
=>	=> exporting layers	0.3s
=>	=> writing image sha256:07f2cb94f79db41f43b9d7794e115c7b2f9e3d25cc91b1cc93c2f57ec2a71052	0.0s
=>	=> naming to docker.io/library/springbootdevops:1.0	0.0s

View build details: [docker-desktop://dashboard/build/default/default/lmi66y7boi0xl3m3ti5oy9p88](http://docker-desktop://dashboard/build/default/default/lmi66y7boi0xl3m3ti5oy9p88)

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview](#)

# Running Spring Boot Applications With Minikube

Check docker image:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
springbootdevops	1.0	07f2cb94f79d	About a minute ago	511MB
registry.k8s.io/kube-apiserver	v1.28.3	537434729123	5 months ago	126MB
registry.k8s.io/kube-scheduler	v1.28.3	6d1b4fd1b182	5 months ago	60.1MB
registry.k8s.io/kube-controller-manager	v1.28.3	10baa1ca1706	5 months ago	122MB
registry.k8s.io/kube-proxy	v1.28.3	bfc896cf80fb	5 months ago	73.1MB
registry.k8s.io/etcd	3.5.9-0	73deb9a3f702	10 months ago	294MB
registry.k8s.io/coredns/coredns	v1.10.1	ead0a4a53df8	13 months ago	53.6MB
registry.k8s.io/pause	3.9	e6f181688397	17 months ago	744kB
gcr.io/k8s-minikube/storage-provisioner	v5	6e38f40d628d	2 years ago	31.5MB

Create Deployment:

```
C:\SpringBoot_DevOps>kubectl create deployment spring-boot-bookhotel --image=springbootdevops:1.0 --port=8080
deployment.apps/spring-boot-bookhotel created
```

```
C:\SpringBoot_DevOps>kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
spring-boot-bookhotel   1/1     1           1          17s
```

# View deployment description

```
C:\SpringBoot_DevOps>kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
spring-boot-bookhotel   1/1      1           1          17s

C:\SpringBoot_DevOps>kubectl describe deployment spring-boot-bookhotel
Name:           spring-boot-bookhotel
Namespace:      default
CreationTimestamp:  Tue, 26 Mar 2024 13:28:51 +0700
Labels:         app=spring-boot-bookhotel
Annotations:    deployment.kubernetes.io/revision: 1
Selector:       app=spring-boot-bookhotel
Replicas:       1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=spring-boot-bookhotel
  Containers:
    springbootdevops:
      Image:      springbootdevops:1.0
      Port:       8080/TCP
      Host Port:  0/TCP
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>
  Conditions:
    Type        Status  Reason
    ----        ----  -----
    Available   True    MinimumReplicasAvailable
    Progressing True    NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  spring-boot-bookhotel-97896bd67 (1/1 replicas created)
Events:
  Type        Reason
  ----        -----
  
```

```
C:\SpringBoot_DevOps>kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
spring-boot-bookhotel-97896bd67-jc94n	1/1	Running	0	21m

Check pods

```
C:\SpringBoot_DevOps>kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
spring-boot-bookhotel   1/1       1           1          35m

C:\SpringBoot_DevOps>kubectl expose deployment spring-boot-bookhotel --type=NodePort
service/spring-boot-bookhotel exposed

C:\SpringBoot_DevOps>kubectl get service
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes        ClusterIP   10.96.0.1      <none>          443/TCP      48m
spring-boot-bookhotel   NodePort    10.111.27.248  <none>          8080:30591/TCP  10s

C:\SpringBoot_DevOps>minikube service spring-boot-bookhotel --url
http://127.0.0.1:52179
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

Create a service in Kubernetes named **spring-boot-hotel** and configures it to use the **NodePort** service type, and see the URL of service

# Result on localhost:

KIOT WEBSITE

Welcome To Kiot Hotel

LUXURY - MODERN - CONVENIENT

CHECK-IN  
Check-in date

CHECK-OUT  
Check-out date

ROOM  
Suite Room

ADULT  
1 Adult

CHILDREN  
None

127.0.0.1:53879

KIOT WEBSITE

Enjoy A Luxury Experience

JOIN WITH US

CHECK-IN  
Check-in date

CHECK-OUT  
Check-out date

ROOM  
Suite Room

ADULT  
1 Adult

CHILDREN  
None

127.0.0.1:53879

KIOT WEBSITE

Greatest Thanks From Your Booking

JOIN WITH US

CHECK-IN  
Check-in date

CHECK-OUT  
Check-out date

ROOM  
Suite Room

ADULT  
1 Adult

CHILDREN  
None

127.0.0.1:53879

KIOT WEBSITE

We Are Always Highly Appreciate Your Feelings

JOIN WITH US

CHECK-IN  
Check-in date

CHECK-OUT  
Check-out date

ROOM  
Suite Room

ADULT  
1 Adult

CHILDREN  
None

127.0.0.1:53879

Dashboard

Booking

Customers

Rooms

Staff

Pricing

Apps

Employees

Accounts

Payroll

Calendar

Service

Good Morning Kien Nguyen!

Dashboard

236

Total Booking



180

Available Rooms



1538

Enquiry



364

Collections



VISITORS

ROOMS BOOKED

Booking

View All

Booking ID	Name	Email ID	Aadhar Number	Room Type	Number	Status
BKG-0001	KienNCT3	[email protected]	12414786454545	Double	631-254-6480	<span>INACTIVE</span>
BKG-0002	HaiNV21	[email protected]	5456223232322	Double	830-468-1042	<span>INACTIVE</span>
BKG-0003	TrongPM	[email protected]	454543232625	Single	508-335-5531	<span>INACTIVE</span>

KIOT WEBSITE

localhost:8080/api/v1/home?#

We Are Always Highly Appreciate Your Feelings

JOIN WITH US

CHECK-IN: 4/25/2024

CHECK-OUT: 5/3/2024

ROOM: Classic Room

ADULT: 2 Adult

CHILDREN: 2 Children

Check Availability

User Login

Email address: admin

Password: ....

LOGIN

Forgot your password?

Login Register

KIOT WEBSITE

localhost:8080/api/v1/home?#

Welcome to Hotel

CHECK-IN: 4/25/2024

CHECK-OUT: 5/3/2024

ROOM: Classic Room

ADULT: 2 Adult

CHILDREN: 2 Children

Check Availability

User Registration

Note: Your details must match with your ID(passport, driving license, etc) that will be required during check-in process.

Your Fullname: admin

Email address:

Phone number:

Date of birth: mm/dd/yyyy

Address:

Password: ....

Confirm your password:

REGISTER

Login Register

To delete a deployment on minikube, we can do steps like this:

```
PS C:\SpringBoot_DevOps> kubectl delete deployment spring-boot-bookhotel  
deployment.apps "spring-boot-bookhotel" deleted
```

```
PS C:\SpringBoot_DevOps> kubectl get deployments  
No resources found in default namespace.
```



default

Search



## Workloads

### Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Ingress Classes

Services

### Config and Storage

Config Maps

Persistent Volume Claims

Secrets

Storage Classes

### Replica Sets

Name	Images	Labels	Pods	Created
spring-boot-bookhotel	springbootdevops:1.0	app: spring-boot-bookhotel	1 / 1	43.minutes.ago

### Deployments

Name	Images	Labels	Pods	Created
spring-boot-bookhotel	springbootdevops:1.0	app: spring-boot-bookhotel	1 / 1	43.minutes.ago

### Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
spring-boot-bookhotel-97896bd67-jc94n	springbootdevops:1.0	app: spring-boot-bookhotel pod-template-hash: 97896bd67	minikube	Running	0	-	-	43.minutes.ago

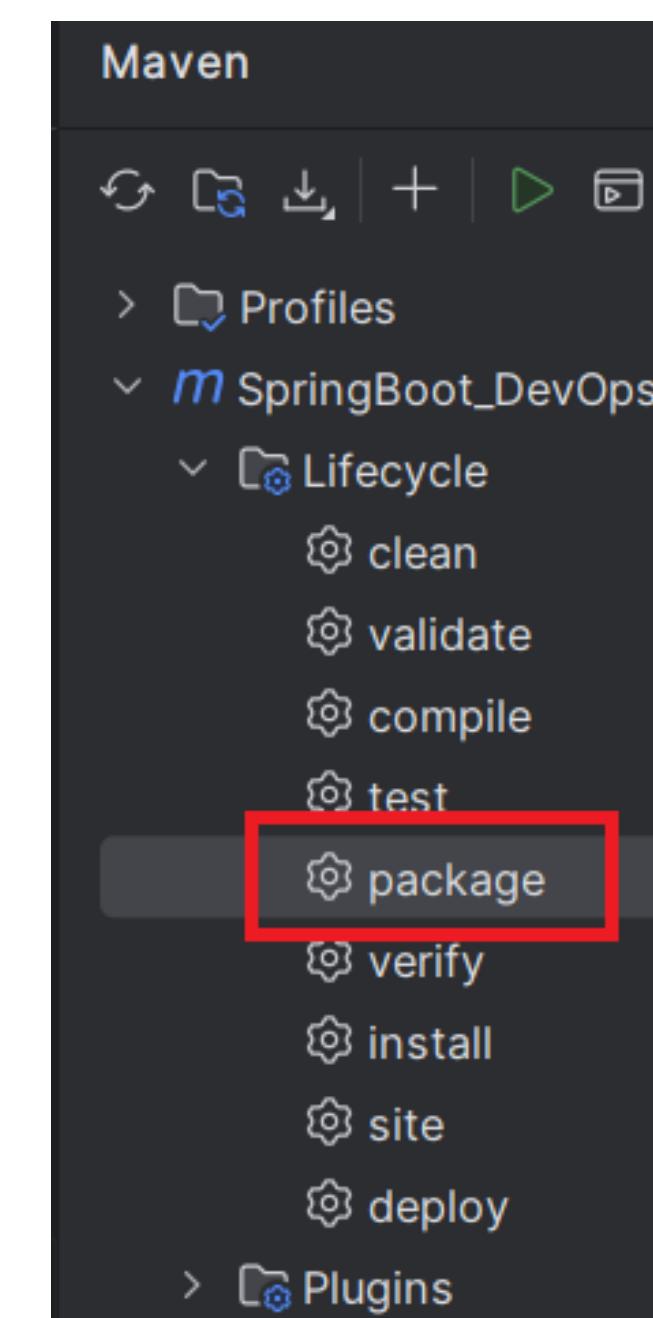
### Replica Sets

Name	Images	Labels	Pods	Created
spring-boot-bookhotel-97896bd67	springbootdevops:1.0	app: spring-boot-bookhotel pod-template-hash: 97896bd67	1 / 1	43.minutes.ago

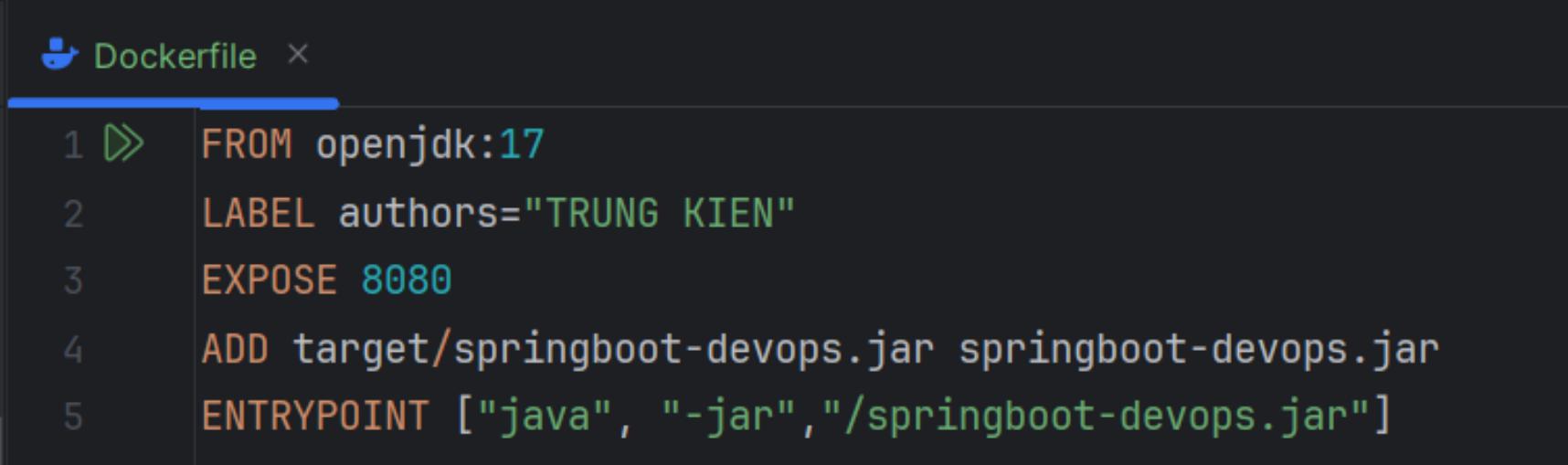
# Dockerizing a Spring Boot Application

- Dockerizing is a process of converting a Spring Boot application into a Docker container.
- When we dockerize an application, we create a Docker image that contains the application and can run it in an isolated environment. This simplifies deployment and makes the application easier to move and run across different environments

```
<build>
    <finalName>springboot-devops</finalName>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <executable>true</executable>
            </configuration>
        </plugin>
    </plugins>
```



# Dockerizing a Spring Boot Application



```
Dockerfile
1 FROM openjdk:17
2 LABEL authors="TRUNG KIEN"
3 EXPOSE 8080
4 ADD target/springboot-devops.jar springboot-devops.jar
5 ENTRYPOINT ["java", "-jar", "/springboot-devops.jar"]
```

This Dockerfile sets up an environment to run a Spring Boot application using OpenJDK 17, adds the JAR file to the container, and defines the entry point for running the application

- Line 1: This line specifies the base image for our Docker container.
- Line 3: This line indicates that the container will listen on port 8080. It's just a hint for anyone running the container to know which port to expose.
- Line 4: The **ADD** instruction copies the Spring Boot JAR file (located at `target/springboot-devops.jar`) into the container. This assumes that the JAR file has been built and is available in the specified location.
- Line 5: The **ENTRYPOINT** specifies the command to run when the container starts. In this case, it runs the Spring Boot application by executing the JAR file (`springboot-devops.jar`) using the Java runtime (`java -jar`).

# Dockerizing a Spring Boot Application

```
PS C:\SpringBoot_DevOps> docker build -t springboot-devops.jar:1.0 .
[+] Building 4.8s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 211B
=> [internal] load metadata for docker.io/library/openjdk:17
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 46.03MB
=> CACHED [1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbaf9f87662e7b3f38cb7d8
=> [2/2] ADD target/springboot-devops.jar springboot-devops.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:88b7f4153693e4048e0538f6eeec84f2b25a8a33f291f56c55f473ecd0f67af3
=> => naming to docker.io/library/springboot-devops.jar:1.0
```

View build details: docker-desktop://dashboard/build/default/default/vj4iyn4yhud2j0gv92f0um005

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview](#)

```
PS C:\SpringBoot_DevOps>
```

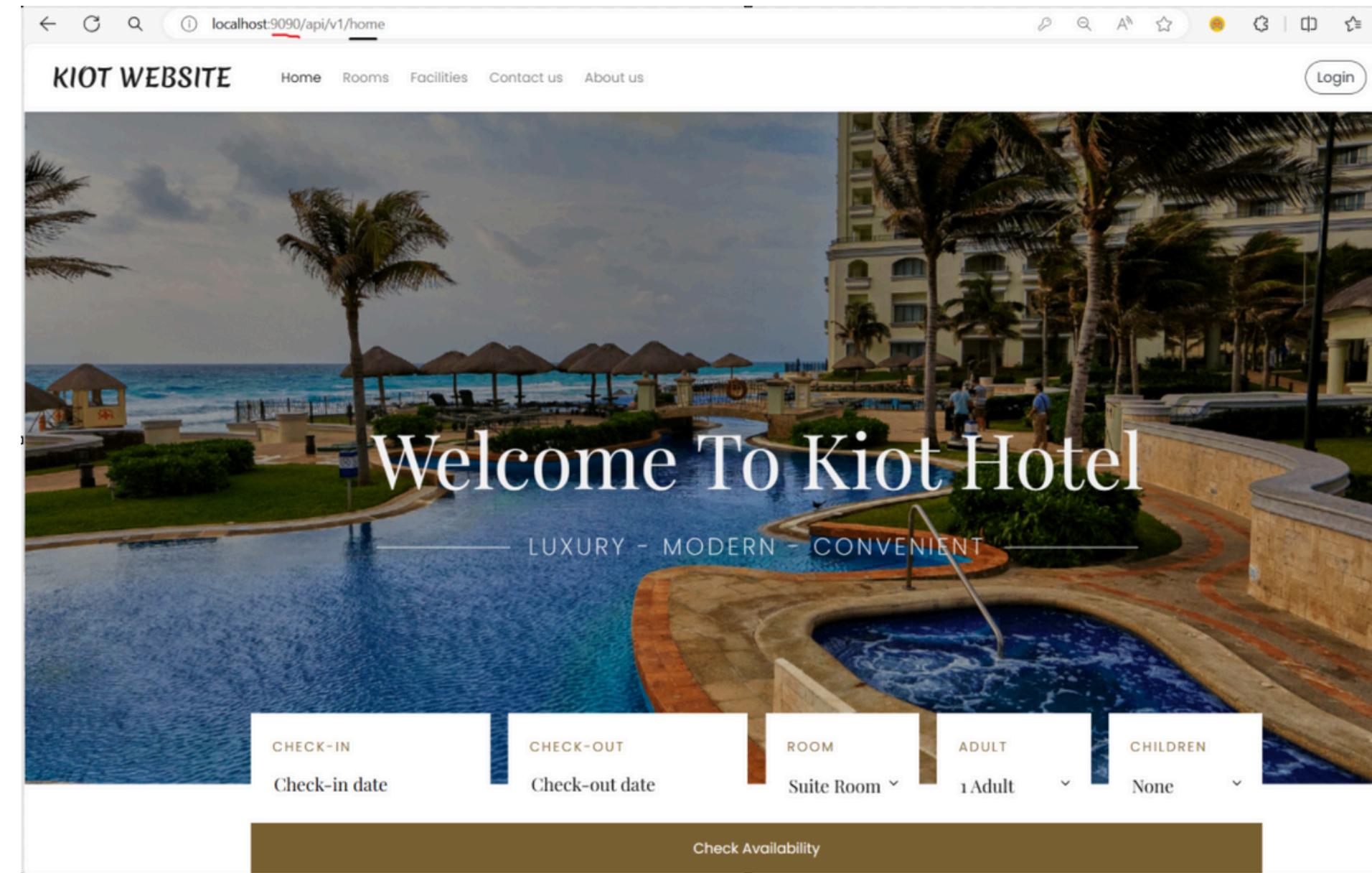
```
PS C:\SpringBoot_DevOps> docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
springboot-devops.jar	1.0	88b7f4153693	About a minute ago	517MB
springboot_devops	1.0	3e0500be49c0	20 hours ago	511MB
spring-boot-docker.jar	latest	89340ffde611	2 days ago	493MB
my-backend-app	1.0	901ac946368d	4 days ago	491MB
my_docker_file	1.0	0edf4c288264	4 days ago	491MB
gcr.io/k8s-minikube/kicbase	v0.0.42	dbc648475405	4 months ago	1.2GB
docker/getting-started	latest	3e4394f6b72f	15 months ago	47MB

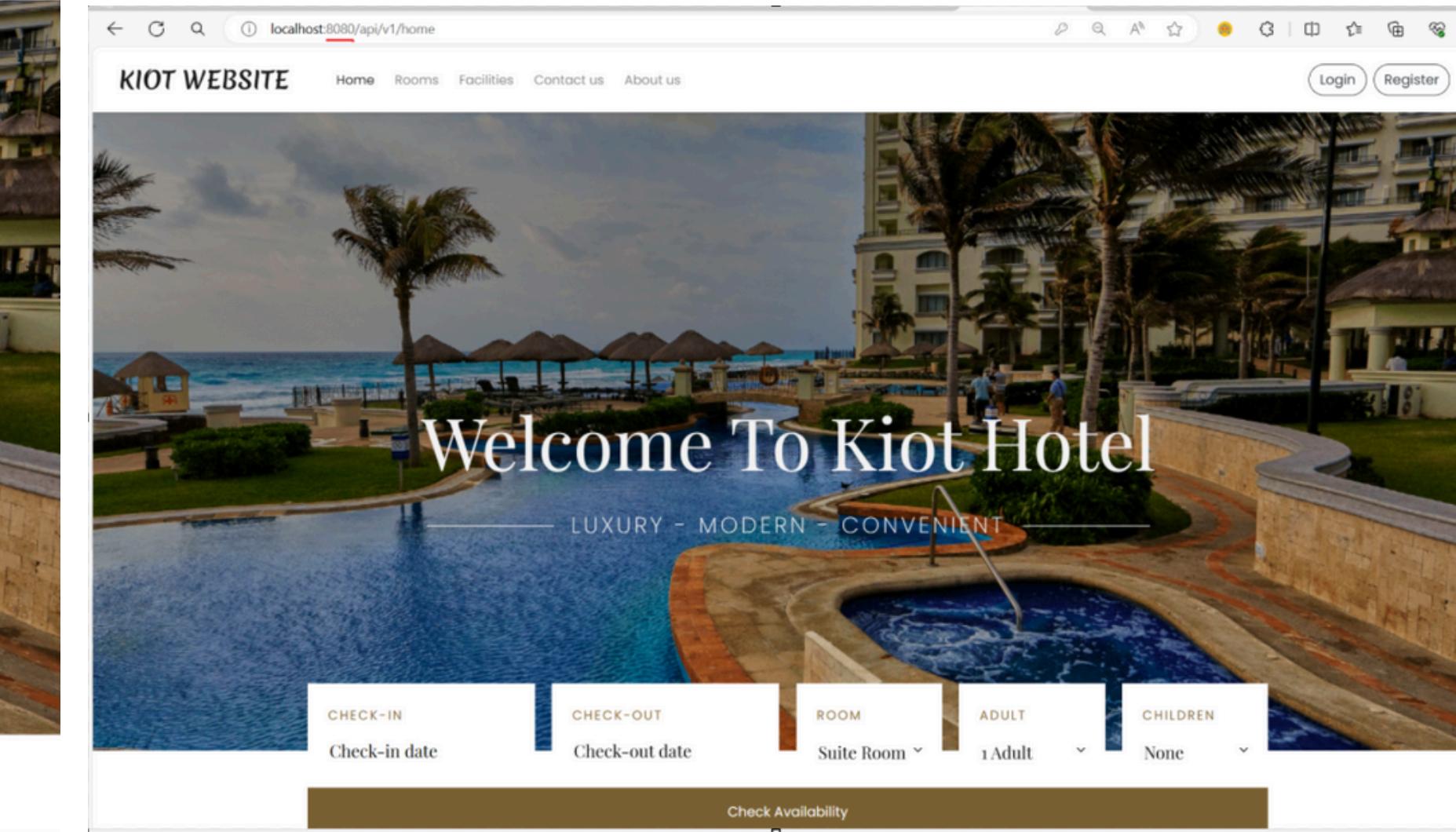
PS C:\SpringBoot\_DevOps> docker run -p 9090:8080 springboot-devops.jar

```
 .----  
 \\\ / _ _ - _(_)_ _ _ _ _ \\\ \  
 ( ( )\_\_ | ' _ | ' _ \ V _ | \ \ \ \ \  
 \\\ _ _| | _ | | _ | | ( _ | ) ) ) )  
 ' _ _ _| . _ | _ | _ | _ \_, | / / / /  
 =====|_|=====|_|/_=/_/_/_/  
 :: Spring Boot :: (v3.2.4)
```

```
2024-03-26T22:43:17.683Z INFO 1 --- [SpringBoot_DevOps] [main] o.k.s.SpringBootDevOpsApplication : Starting SpringBootDevOpsApplication v0.0.  
1-SNAPSHOT using Java 17.0.2 with PID 1 (/springboot-devops.jar started by root in /)  
2024-03-26T22:43:17.703Z INFO 1 --- [SpringBoot_DevOps] [main] o.k.s.SpringBootDevOpsApplication : No active profile set, falling back to 1 d  
efault profile: "default"  
2024-03-26T22:43:19.044Z INFO 1 --- [SpringBoot_DevOps] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)  
2024-03-26T22:43:19.061Z INFO 1 --- [SpringBoot_DevOps] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]  
2024-03-26T22:43:19.062Z INFO 1 --- [SpringBoot_DevOps] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10  
.1.19]  
2024-03-26T22:43:19.106Z INFO 1 --- [SpringBoot_DevOps] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicatio  
nContext  
2024-03-26T22:43:19.108Z INFO 1 --- [SpringBoot_DevOps] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization  
completed in 1253 ms  
2024-03-26T22:43:19.238Z INFO 1 --- [SpringBoot_DevOps] [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page template: index  
2024-03-26T22:43:19.553Z INFO 1 --- [SpringBoot_DevOps] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with co  
ntext path ''
```



Check result on localhost:



# Creating Docker Images with Spring Boot

1) The traditional way of building Docker images with Spring Boot is to use a **Dockerfile**

```
git Dockerfile × m pom.xml (SpringBoot_DevOps)

1 FROM openjdk:17
2 LABEL authors="TRUNG KIEN"
3 EXPOSE 8080
4 ADD target/springboot-docker.jar springboot-docker.jar
5 ENTRYPOINT ["java", "-jar", "/springboot-docker.jar"]
```

```
PS C:\SpringBoot_DevOps> docker build -t springboot-docker.jar .
[+] Building 2.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 211B
=> [internal] load metadata for docker.io/library/openjdk:17
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 78B
=> [1/2] FROM docker.io/library/openjdk:17@sha256:528707081fdb9562eb819128a9f85ae7fe000e2fbbaef9f87662e7b3f38cb7d8
=> CACHED [2/2] ADD target/springboot-docker.jar springboot-docker.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:f829daca654c28003b62091a65baec47120b5d3f8094d47afb37c8dd140d88ab
=> => naming to docker.io/library/springboot-docker.jar
```

We build “**springboot-docker.jar**”  
image

# Creating Docker Images with Spring Boot

1) The traditional way of building Docker images with Spring Boot is to use a **Dockerfile**

# Check image and run container from a image

```
FROM openjdk:17
LABEL authors="TRUNG KIEN"
EXPOSE 8080
ADD target/springboot-docker.jar springboot-docker.jar
ENTRYPOINT ["java", "-jar", "/springboot-docker.jar"]
```

Docker Images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
springboot-docker.jar	1.0	f829daca654c	6 seconds ago	517MB
springboot-devops.jar	1.0	88b7f4153693	4 hours ago	517MB

# Run container from image “springboot-docker.jar”

```
PS C:\SpringBoot_DevOps> docker run -p 2711:8080 springboot-docker.jar

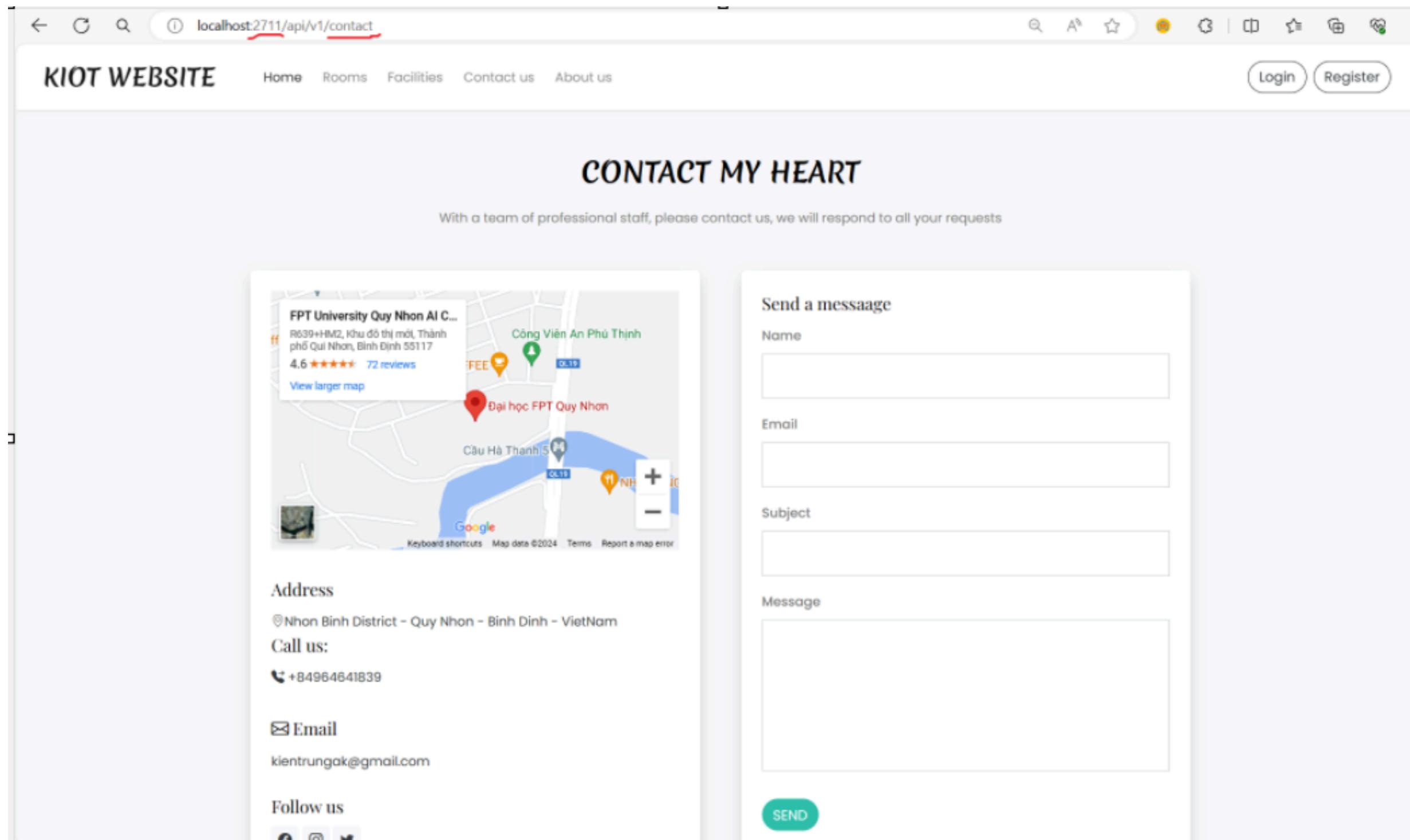
 .   _--_ 
/\ / __'_--_ _(_)_ -- _--_ \ \ \ \
( ( )\__| '_| | '_| \_` | \ \ \ \
\ \ \_)| |_)| | | | | | | ( | | ) ) ) )
' |_____| .__|_| |_|_|_| |_\__, | / / / /
=====|_|=====|_|=/_/_/_/
:: Spring Boot ::           (v3.2.4)

2024-03-27T02:23:37.866Z  INFO 1 --- [SpringBoot_DevOps] [           main] o.k.s.SpringBootDevOpsApplication      : Starting SpringBootDevOpsApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 1 (/springboot-docker.jar started by root in /)
2024-03-27T02:23:37.876Z  INFO 1 --- [SpringBoot_DevOps] [           main] o.k.s.SpringBootDevOpsApplication      : No active profile set, falling back to default profile: "default"
2024-03-27T02:23:39.902Z  INFO 1 --- [SpringBoot_DevOps] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-03-27T02:23:39.939Z  INFO 1 --- [SpringBoot_DevOps] [           main] o.apache.catalina.core.StandardService  : Starting service [Tomcat]
2024-03-27T02:23:39.940Z  INFO 1 --- [SpringBoot_DevOps] [           main] o.apache.catalina.core.StandardEngine   : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-03-27T02:23:40.064Z  INFO 1 --- [SpringBoot_DevOps] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplication
```

# Creating Docker Images with Spring Boot

1) The traditional way of building Docker images with Spring Boot is to use a **Dockerfile**

Result on external port:



# Creating Docker Images with Spring Boot

## 2) Use Buildpacks

Buildpacks are a tool that provides framework and application dependencies.

```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <configuration>
    <image>
      <name>spring-jpa</name>
    </image>
    <excludes>
      <exclude>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
      </exclude>
    </excludes>
  </configuration>
</plugin>
```

```
PS D:\Fsoft\Day5_ExtraJPA> mvn spring-boot:build-image
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.KienNguyenFPT:Day5_ExtraJPA >-----
[INFO] Building Day5_ExtraJPA 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:3.2.2:build-image (default-cli) > package @ Day5_ExtraJPA >>>
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ Day5_ExtraJPA ---
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO] Copying 7 resources from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ Day5_ExtraJPA ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ Day5_ExtraJPA ---
[INFO] skip non existing resourceDirectory D:\Fsoft\Day5_ExtraJPA\src\test\resources
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ Day5_ExtraJPA ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ Day5_ExtraJPA ---
[INFO] Using auto-detected provider org.apache.maven.surefire.junit3.JUnit3Provider
```

We use this command to build docker image

# Creating Docker Images with Spring Boot

## 2) Use Buildpacks

Result:

```
[INFO] Successfully built image 'docker.io/library/spring-jpa:latest'  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 06:28 min  
[INFO] Finished at: 2024-03-27T09:49:09+07:00  
[INFO] -----
```

```
PS D:\Fsoft\Day5_ExtraJPA> docker images  
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE  
springboot-docker.jar    1.0      f829daca654c  35 minutes ago  517MB  
springboot-docker.jar    latest    f829daca654c  35 minutes ago  517MB  
springboot-devops.jar   1.0      88b7f4153693  4 hours ago   517MB  
springboot-devops.jar   latest    88b7f4153693  4 hours ago   517MB  
springboot_devops       1.0      3e0500be49c0  24 hours ago  511MB  
spring-boot-docker.jar  latest    89340ffde611  2 days ago   493MB  
my-backend-app         1.0      901ac946368d  4 days ago   491MB  
my_docker_file         1.0      0edf4c288264  4 days ago   491MB  
paketobuildpacks/run-jammy-base  latest    530c11846f48  8 days ago   106MB  
gcr.io/k8s-minikube/kicbase    v0.0.42  dbc648475405  4 months ago  1.2GB  
docker/getting-started     latest    3e4394f6b72f  15 months ago  47MB  
paketobuildpacks/builder-jammy-base  latest    f5b0161594a1  44 years ago  1.42GB  
spring-jpa              latest    efddfb330b44  44 years ago  328MB
```

Check docker image

# Creating Docker Images with Spring Boot

## 2) Use Buildpacks

Run container from image “spring-jpa”

```
PS D:\Fsoft\Day5_ExtraJPA> docker run -p 1103:2403 spring-jpa
```

The internal port is 2403 because we config it in file .properties

```
server.error.whitelabel.enabled=false  
server.port=2403
```

# Deploy Spring Boot WAR into Tomcat Server

First, we need to download Tomcat 10 Server

10.1.19

Please see the [README](#) file for packaging information. It explains what every distribution contains.

## Binary Distributions

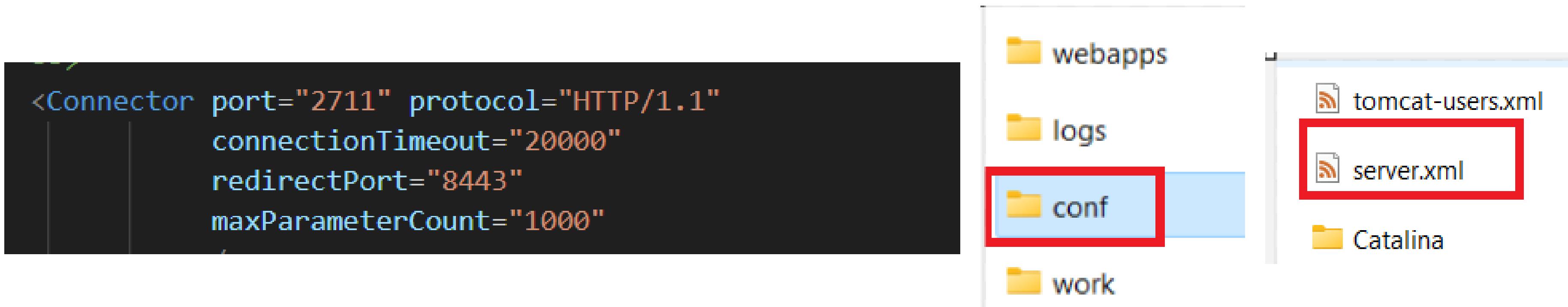
- Core:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
  - [32-bit Windows zip \(pgp, sha512\)](#)
  - [~~64-bit Windows zip \(pgp, sha512\)~~](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
  - [tar.gz \(pgp, sha512\)](#)
- Deployer:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
- Embedded:
  - [tar.gz \(pgp, sha512\)](#)
  - [zip \(pgp, sha512\)](#)

## Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

# Deploy Spring Boot WAR into Tomcat Server

Then, we config port to **2711** (optional) or use **8085** (default port) to access Tomcat sever (Change this config in **server.xml**, in **config** folder)



After that, we open terminal and run this command to access Tomcat server

```
PS C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\bin> .\startup.bat
Using CATALINA_BASE:   "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19"
Using CATALINA_HOME:   "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19"
Using CATALINA_TMPDIR: "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk-17"
Using CLASSPATH:       "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\bin\bootstrap.jar;C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""
```

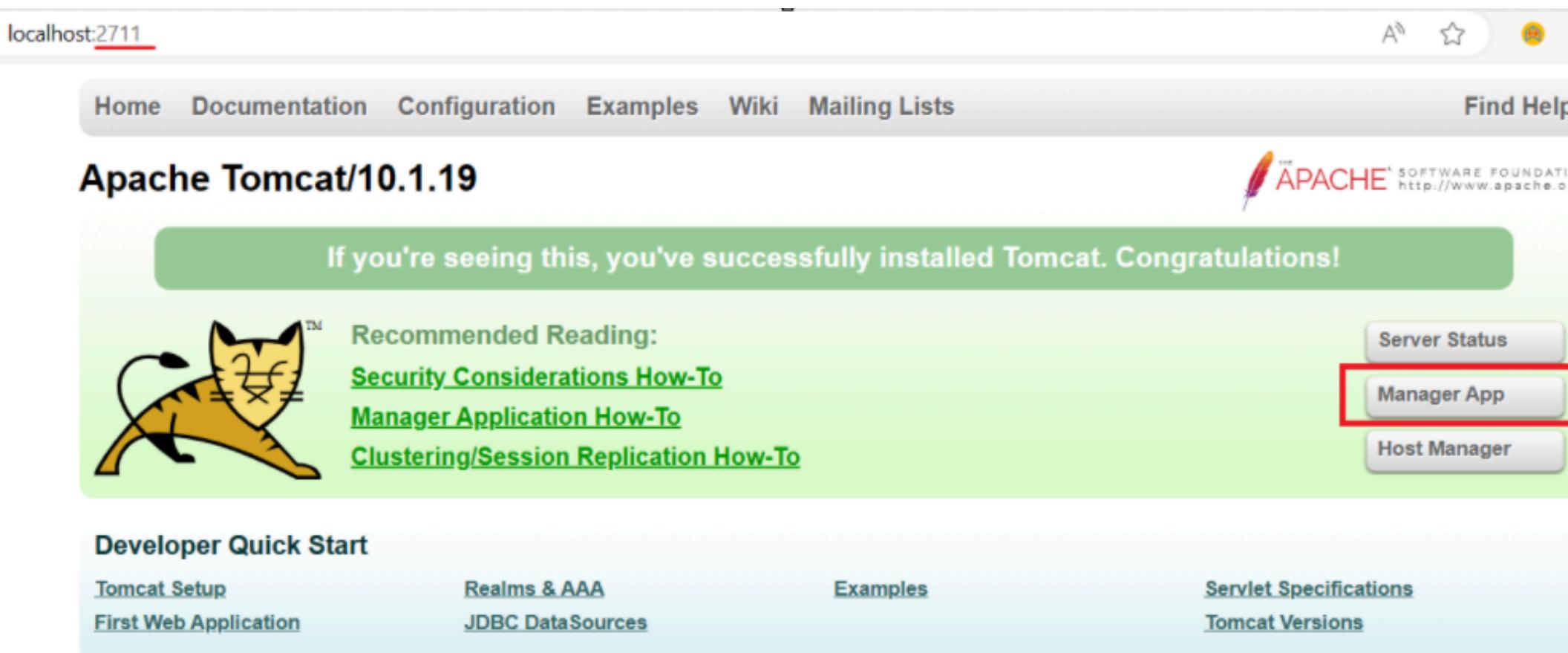
# Deploy Spring Boot WAR into Tomcat Server

```
PS C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\bin> .\startup.bat
Using CATALINA_BASE:  "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19"
Using CATALINA_HOME:   "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19"
Using CATALINA_TMPDIR: "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk-17"
Using CLASSPATH:       "C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\bin\bootstrap.jar;C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""
```

```
Tomcat - + ×
24-Mar-2024 15:26:11.331 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet engine: [Apache Tomcat/10.1.19]
24-Mar-2024 15:26:11.355 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\docs]
24-Mar-2024 15:26:11.880 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\docs] has finished in [517] ms
24-Mar-2024 15:26:11.880 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\examples]
24-Mar-2024 15:26:12.501 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\examples] has finished in [621] ms
24-Mar-2024 15:26:12.516 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\host-manager]
24-Mar-2024 15:26:12.593 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\host-manager] has finished in [77] ms
24-Mar-2024 15:26:12.593 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\manager]
24-Mar-2024 15:26:12.650 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\manager] has finished in [57] ms
24-Mar-2024 15:26:12.650 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\ROOT]
24-Mar-2024 15:26:12.674 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\Users\TRUNG KIEN\Downloads\apache-tomcat-10.1.19-windows-x64\apache-tomcat-10.1.19\webapps\ROOT] has finished in [24] ms
24-Mar-2024 15:26:12.688 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-2711"]
24-Mar-2024 15:26:12.752 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [1533] milliseconds
```

# Deploy Spring Boot WAR into Tomcat Server

Here is the UI of **Tomcat server** when we access successfully



The screenshot shows the Apache Tomcat 10.1.19 homepage. At the top, there is a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. A "Find Help" search bar is also present. The main content area features a green banner with the text "If you're seeing this, you've successfully installed Tomcat. Congratulations!" and a cartoon cat icon. Below the banner, there is a section titled "Recommended Reading" with links to Security Considerations How-To, Manager Application How-To, and Clustering/Session Replication How-To. To the right, there are three buttons: "Server Status", "Manager App", and "Host Manager", with "Manager App" being highlighted by a red box. The page also includes sections for Developer Quick Start, Managing Tomcat, Documentation, and Getting Help.

localhost:2711

Apache Tomcat/10.1.19

If you're seeing this, you've successfully installed Tomcat. Congratulations!

TM

Recommended Reading:

[Security Considerations How-To](#)  
[Manager Application How-To](#)  
[Clustering/Session Replication How-To](#)

Server Status

Manager App

Host Manager

Developer Quick Start

[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)  
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in: [\\$CATALINA\\_HOME/conf/tomcat-users.xml](#)

In Tomcat 10.1 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#) [Changelog](#) [Migration Guide](#) [Security Notices](#)

Documentation

[Tomcat 10.1 Documentation](#) [Tomcat 10.1 Configuration](#) [Tomcat Wiki](#)

Find additional important configuration information in: [\\$CATALINA\\_HOME/RUNNING.txt](#)

Developers may be interested in:

[Tomcat 10.1 Bug Database](#) [Tomcat 10.1 JavaDocs](#) [Tomcat 10.1 Git Repository at GitHub](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

[tomcat-announce](#) Important announcements, releases, security vulnerability notifications. (Low volume).  
[tomcat-users](#) User support and discussion  
[taglibs-user](#) User support and discussion for Apache Taglibs  
[tomcat-dev](#) Development mailing list, including commit messages

Other Downloads

[Tomcat Connectors](#) [Tomcat Native](#) [Tomcat](#)

Other Documentation

[Tomcat Connectors](#) [mod\\_jk Documentation](#) [Tomcat Native](#)

Get Involved

[Overview](#) [Source Repositories](#) [Mailing Lists](#)

Miscellaneous

[Contact](#) [Legal](#) [Sponsorship](#)

Apache Software Foundation

[Who We Are](#) [Heritage](#)

# Deploy Spring Boot WAR into Tomcat Server

Here is the UI of *Tomcat manager*, we access by configuring *username* and *password* in **tomcat-server.xml** file, belong to **config** folder

localhost:2711/manager/html

Sign in to access this site  
Authorization required by http://localhost:2711

Username   
Password

Sign in Cancel

```
<user username="root" password="trungkien" roles="manager-gui, manager-script" />
```

After login successfully:

localhost:2711/manager/html

THE APACHE SOFTWARE FOUNDATION

Tomcat Web Application Manager

Message: OK

Manager

List Applications    HTML Manager Help    Manager Help    Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

tomcat-users.xml

server.xml

Catalina

# Deploy Spring Boot WAR into Tomcat Server

Here is quick guide to deploy Spring Boot app to Tomcat server:

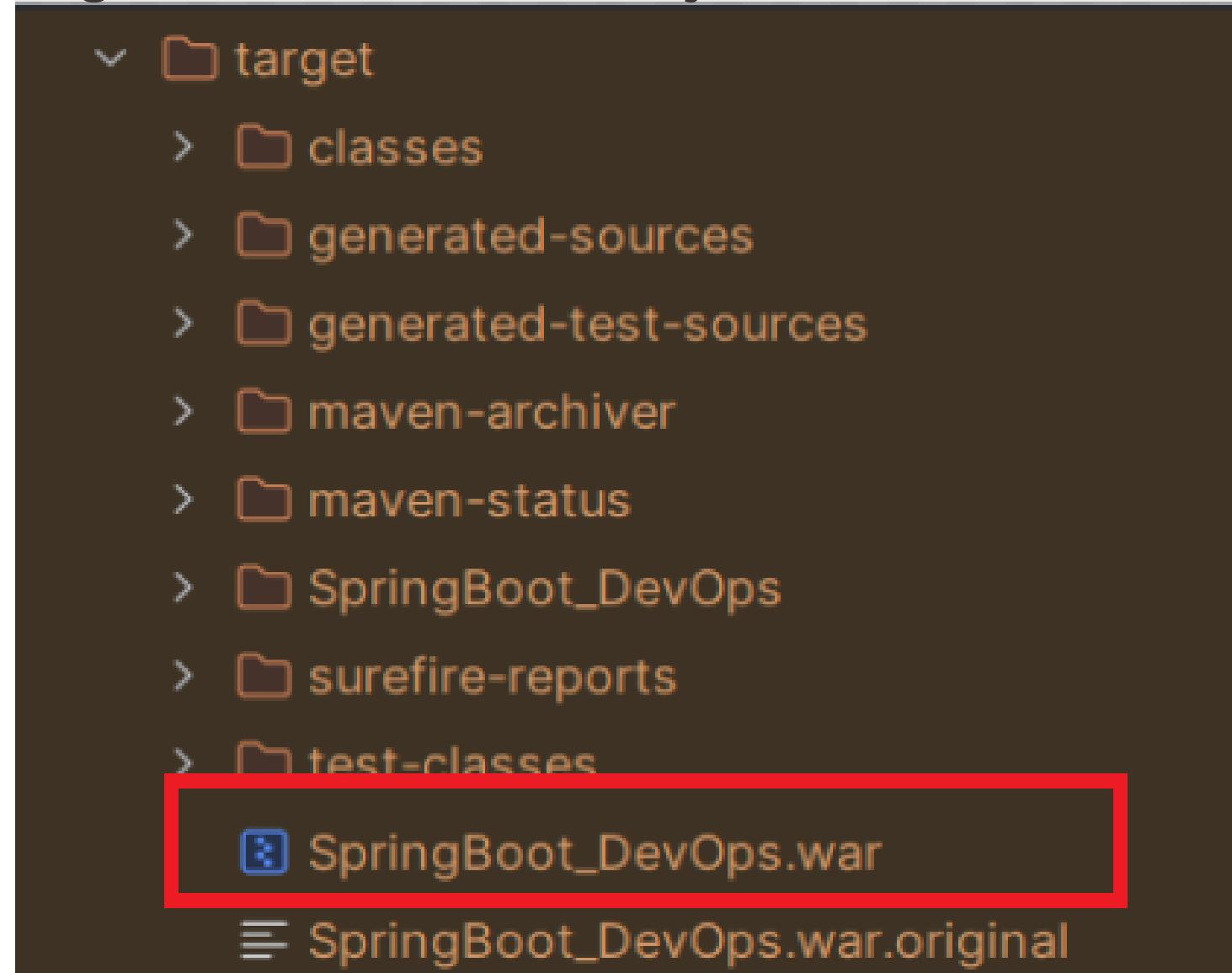
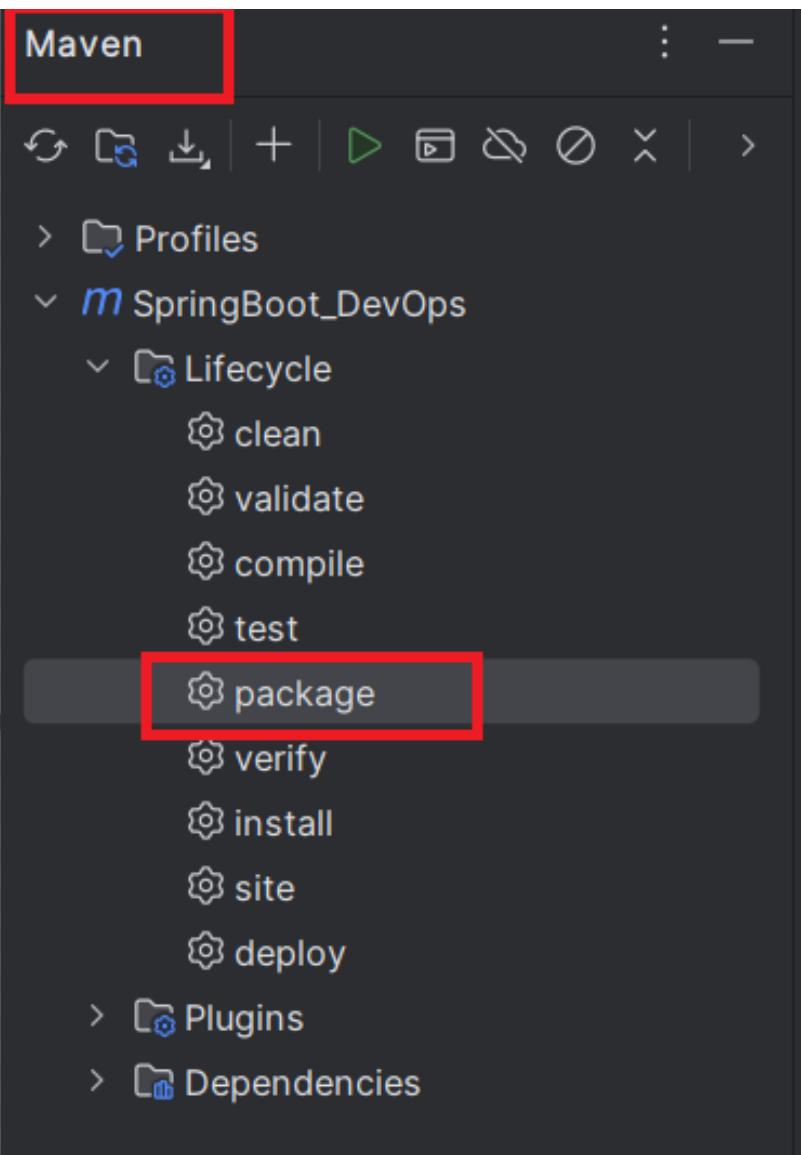
- 1) Create WAR packages.
- 2) Add Tomcat dependency in file `pom.xml`(Maven).
- 3) Extend `SpringBootServletInitializer` to main class.
- 4) Access to Tomcat manager and deploy Spring Boot application.

# 1) Create WAR packages and add *Tomcat* dependency

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
</dependency>
</dependencies>
<build>
    <finalName>${artifactId}</finalName>
```

```
<packaging>war</packaging>
```

After build WAR packages successfully:



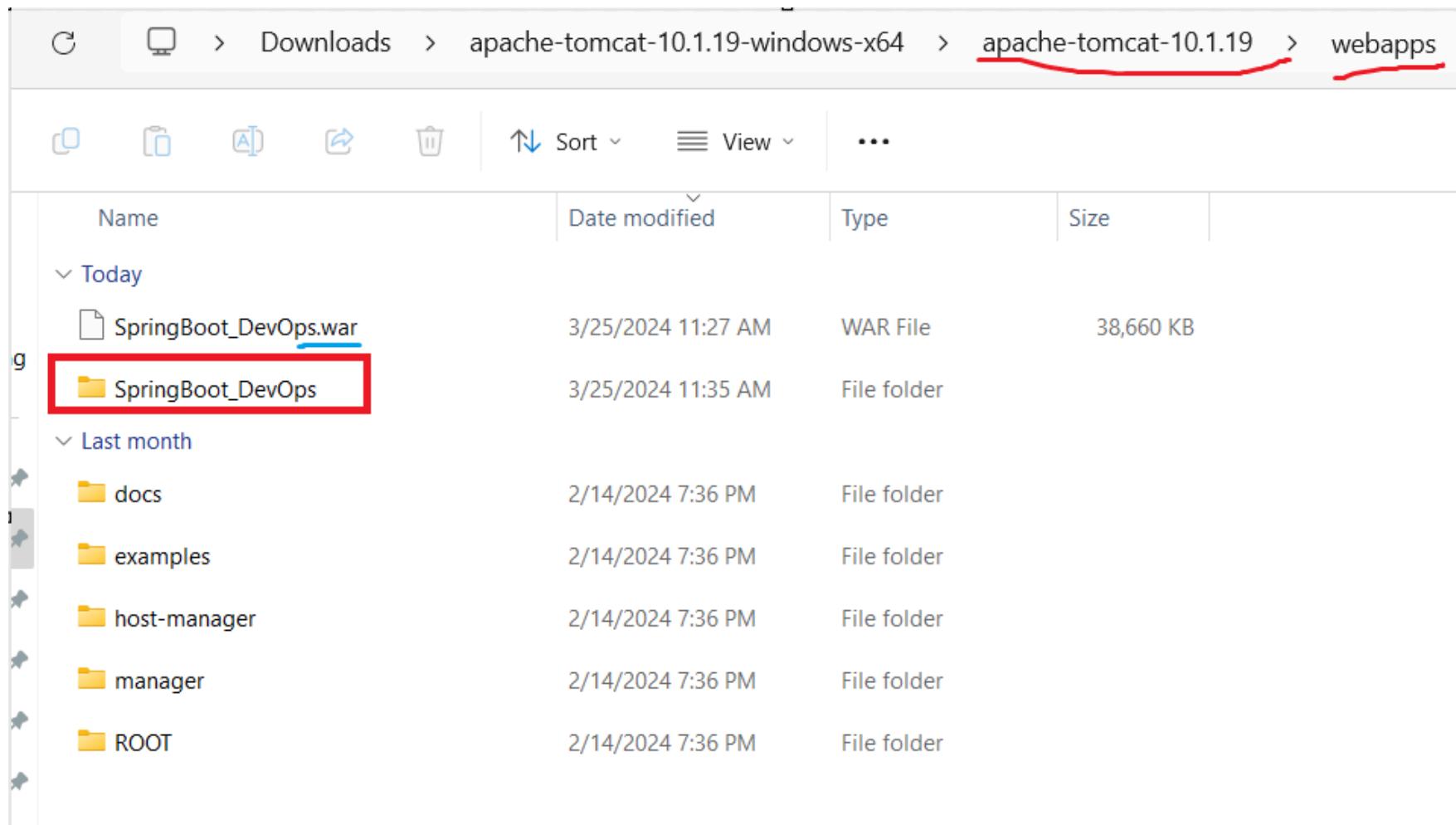
## 2) Extend *SpringBootServletInitializer* class to main method.

```
@SpringBootApplication  
public class SpringBootDevOpsApplication extends SpringBootServletInitializer {  
    new *  
    public static void main(String[] args) { SpringApplication.run(SpringBootDevOpsApplication.class, args);  
    new *  
    @Override  
    protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {  
        return builder.sources(SpringBootDevOpsApplication.class);  
    }  
}
```

Here is the way that a Spring Boot application can be configured to run on a **servlet server** like Tomcat, by extending from the *SpringBootServletInitializer* class and overriding the **configure** method.

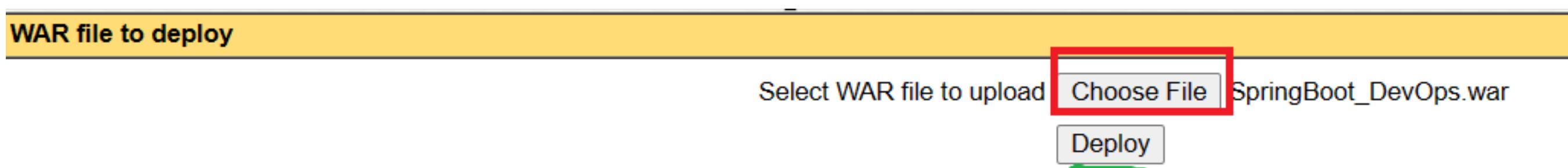
### 3) Access to Tomcat manager and deploy Spring Boot application.

#### a) First way to deploy:



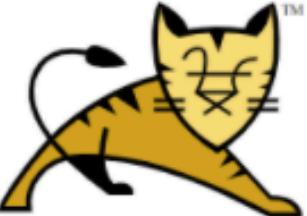
- Copy file “**SpringBoot\_DevOps.war**”, then paste into **apache-tomcat-10** folder, then project will automatically deploy into server.

#### b) Second way to deploy: Access to Tomcat manager and then choose file and deploy directly.



# After deploy successfully:

localhost:2711/manager/html



## Tomcat Web Application Manager

Message: OK

**Manager**

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

**Applications**

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/SpringBoot_DevOps	None specified		true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	<a href="#">Start</a> <a href="#">Stop</a> <a href="#">Reload</a> <a href="#">Undeploy</a> <a href="#">Expire sessions</a> with idle ≥ 30 minutes

Check result on localhost:

The screenshot shows a web browser window for the 'Wellcome | Kiot Hotel' website, accessed via 'localhost:2711/SpringBoot\_DevOps/home'. The page features a large background image of a tropical resort with palm trees, a swimming pool, and a multi-story building. Overlaid on the image is the text 'Welcome To Kiot Hotel' in large white letters and 'LUXURY - MODERN - CONVENIENT' in smaller white letters. At the bottom, there is a form for entering travel details: 'CHECK-IN' (with 'Check-in date' field), 'CHECK-OUT' (with 'Check-out date' field), 'ROOM' (with 'Suite Room' dropdown), 'ADULT' (with '1 Adult' dropdown), and 'CHILDREN' (with 'None' dropdown).

KIOT WEBSITE

Home Rooms Facilities Contact us About us

Login Register

# Welcome To Kiot Hotel

LUXURY - MODERN - CONVENIENT

CHECK-IN

Check-in date

CHECK-OUT

Check-out date

ROOM

Suite Room

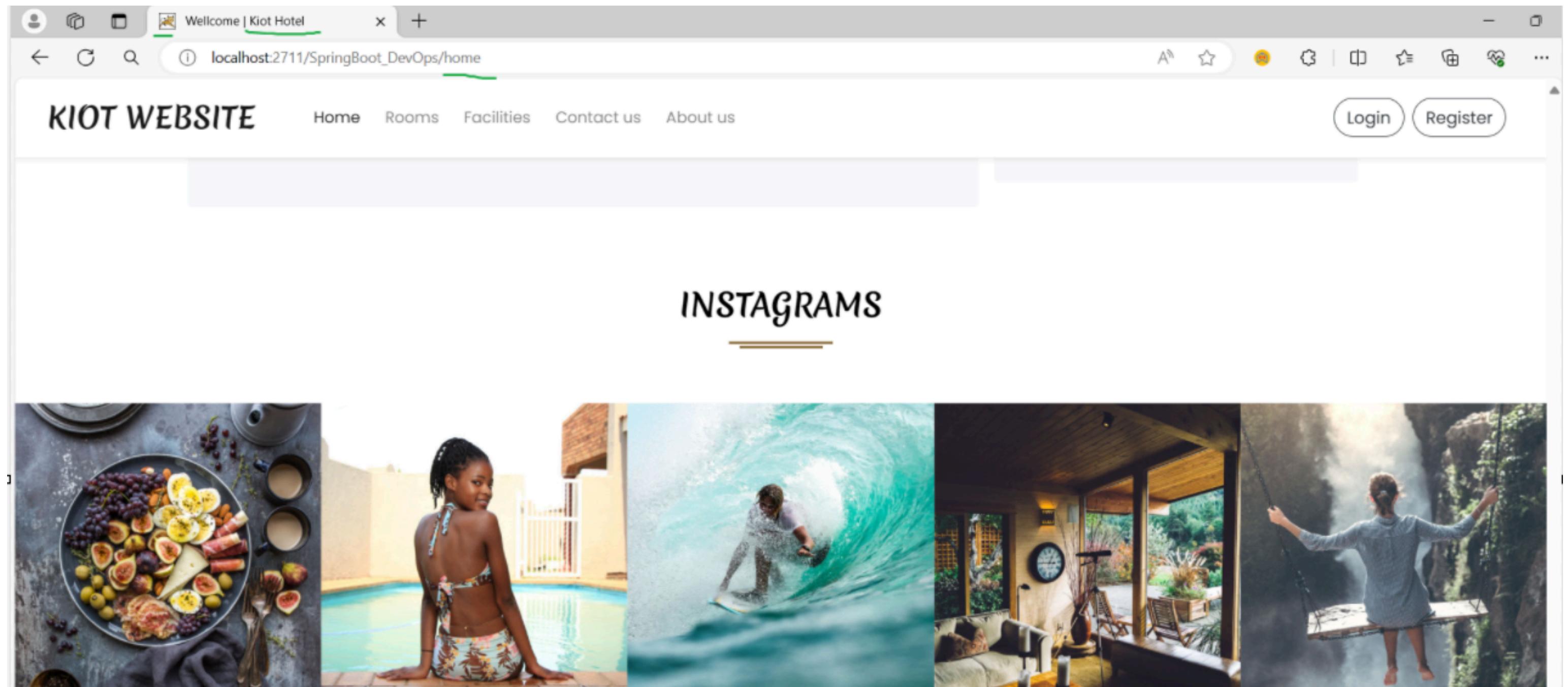
ADULT

1 Adult

CHILDREN

None

# Check result on localhost:



## KIOT WEBSITE

[Home](#) [Rooms](#) [Facilities](#) [Contact us](#) [About us](#) [Login](#) [Register](#)

## INSTAGRAMS



### KIOT HOTEL

Kiot Hotel was established in September 2023, with CEOs in King Of Loser team. With the goal of always making customers happy and building trust in the first place, we will always bring you services, providing excellent experience in this place.

#### Links

[Home](#)  
[Rooms](#)  
[Facilities](#)  
[Contact us](#)

#### Follow us

 [Gmail](#)  
 [Facebook](#)  
 [Twitter](#)  
 [Instagram](#)

# Check result on localhost:

KIOT WEBSITE

Home Rooms Facilities Contact us About us

Login Register

Suite Room

\$120.00 per night

Max: 3 Persons  
Size: 45 m<sup>2</sup>  
View: Sea View  
Bed: 1

Book now More details

Family Room

\$20.00 per night

Max: 3 Persons  
Size: 45 m<sup>2</sup>  
View: Sea View  
Bed: 1

Book now More details

Deluxe Room

\$150.00 per night

Max: 5 Persons  
Size: 45 m<sup>2</sup>  
View: Sea View  
Bed: 2

Book now More details

Advanced Search

Check In Date

Check Out Date

Room Type

0 Adult

0 Children

25000 - 50000

Search

Star Rating

# Check result on localhost:

localhost:2711/SpringBoot\_DevOps/contact

## KIOT WEBSITE

Home Rooms Facilities Contact us About us Login Register

# CONTACT US

With a team of professional staff, please contact us, we will respond to all your requests

FPT University Quy Nhon AI C...  
R639+HM2, Khu đô thị mới, Thành phố Qui Nhơn, Bình Định 55117  
4.6 ★★★★★ 72 reviews  
[View larger map](#)

Address

⑨Nhon Binh District - Quy Nhon - Binh Dinh - VietNam

Call us:

📞 +84964641839

Email

kientrungak@gmail.com

### Send a message

Name

Email

Subject

Message

# A Guide to Spring Boot Admin

- Spring Boot Admin is a web application, used for managing and monitoring Spring Boot applications.
- Each application is considered a client and registered with the admin server. Behind the scenes, this operation is performed by the Spring Boot Actuator endpoints.

Here are steps to configure a Spring Boot Admin server and make an application a client:

- 1) Admin server setup
- 2) Client server setup
- 3) Security setup (Highly recommend)

# A Guide to Spring Boot Admin

## 1) Admin server setup:

```
<dependency>
    <groupId>de.codecentric</groupId>
    <artifactId>spring-boot-admin-starter-server</artifactId>
</dependency>
```

After this, the `@EnableAdminServer` will be available, so we'll be adding it to the main class:

```
@EnableAdminServer
@SpringBootApplication
public class TestDay10Application {
    new *
    public static void main(String[] args) { SpringApplication.run(TestDay10Application.class, args); }
}
```

At this time, we're ready to start the server and register client applications.

The screenshot shows a web browser window with the URL `localhost:9090/applications` in the address bar. The page is titled "Spring Boot Admin". The navigation bar includes links for "Wallboard", "Applications" (which is highlighted in green), and "Journal". Below the navigation bar, there are buttons for "Applications 0", "Instances 0", and a "Filter" dropdown. A large, dark gray circular icon with a sad face is centered on the page, indicating that no applications are registered. The text "No applications registered." is displayed below the icon. A red box highlights the "Applications" button in the top navigation bar.

# A Guide to Spring Boot Admin

## 1) Admin server setup:

In some cases, we still want to set user, password for protect sensitive data,

```
<dependency>
  <groupId>de.codecentric</groupId>
  <artifactId>spring-boot-admin-starter-server</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Then we config our username, password in file .properties

```
application.properties ×

1 spring.application.name=SpringBoot_ExtraDevOps_1
2 server.port=9090
3 spring.security.user.name=admin
4 spring.security.user.password=trungkien
```

# A Guide to Spring Boot Admin

After set up, we access to Admin sever, then we must login to continue:

Please sign in

After login successfully to Admin server:

localhost:9090/applications

Spring Boot Admin Wallboard Applications Journal English trungkien

Applications 0 Instances 0 Filter

No applications registered.

# A Guide to Spring Boot Admin

## 2) Client setup:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
    <groupId>de.codecentric</groupId>
    <artifactId>spring-boot-admin-starter-client</artifactId>
    <version>${spring-boot-admin-starter-client.version}</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

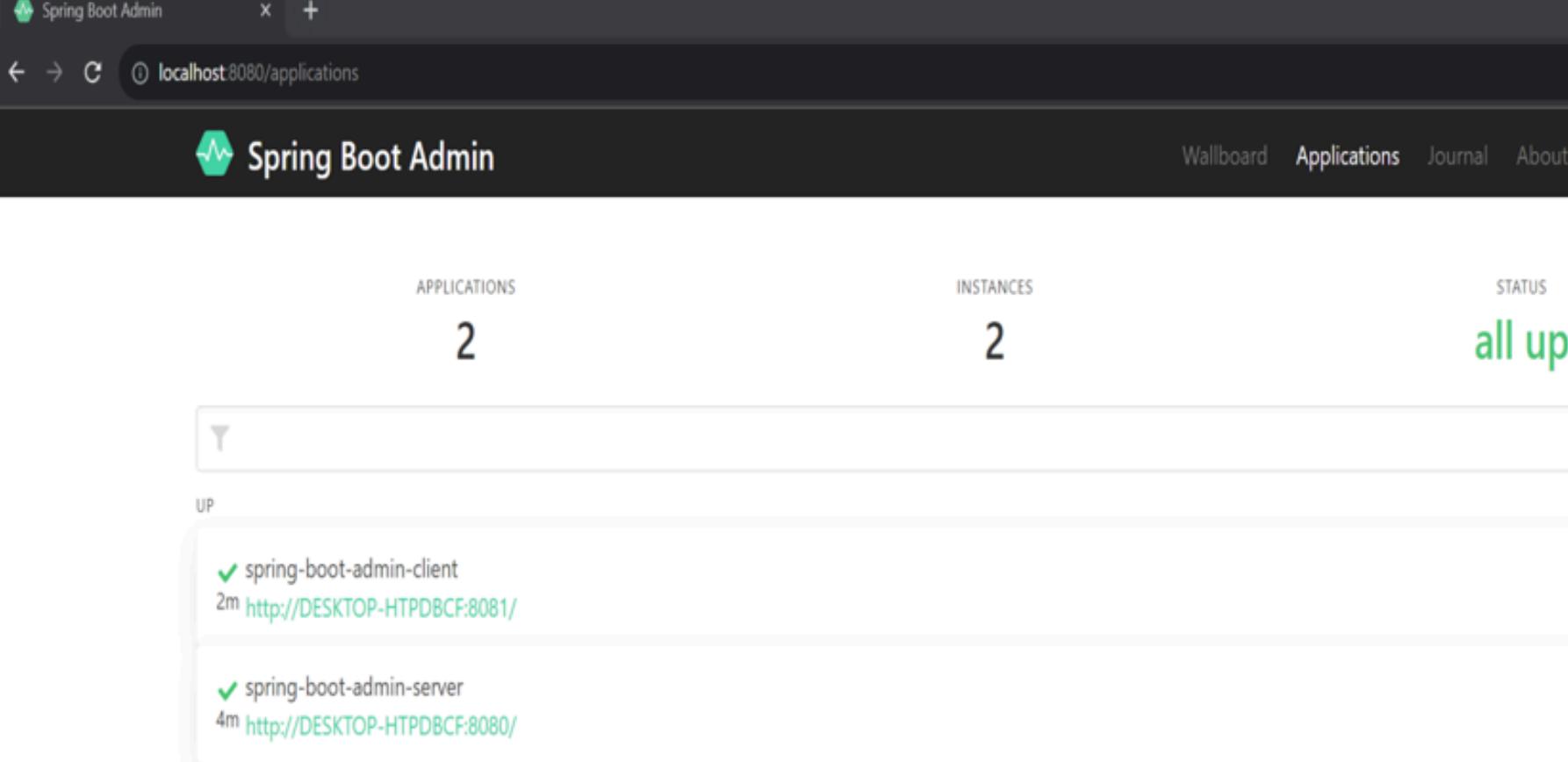
## application.properties

```
</> application.properties x
1 #basic auth credentials
2 spring.security.user.name=client
3 spring.security.user.password=client
4
5 #configs to connect to a secured server
6 spring.boot.admin.client.url=http://localhost:8080
7 spring.boot.admin.client.username=admin
8 spring.boot.admin.client.password=admin
9
10 #configs to give secured server info
11 spring.boot.admin.client.instance.metadata.user.name=${spring.security.user.name}
12 spring.boot.admin.client.instance.metadata.user.password=${spring.security.user.password}
13
14 #app config
15 spring.application.name=spring-boot-admin-client
16 server.port=8081
17
18 management.endpoints.web.exposure.include=*
19 management.endpoint.health.show-details=always
```

expose all the endpoints

# A Guide to Spring Boot Admin

## Access to Spring Boot Admin Console:

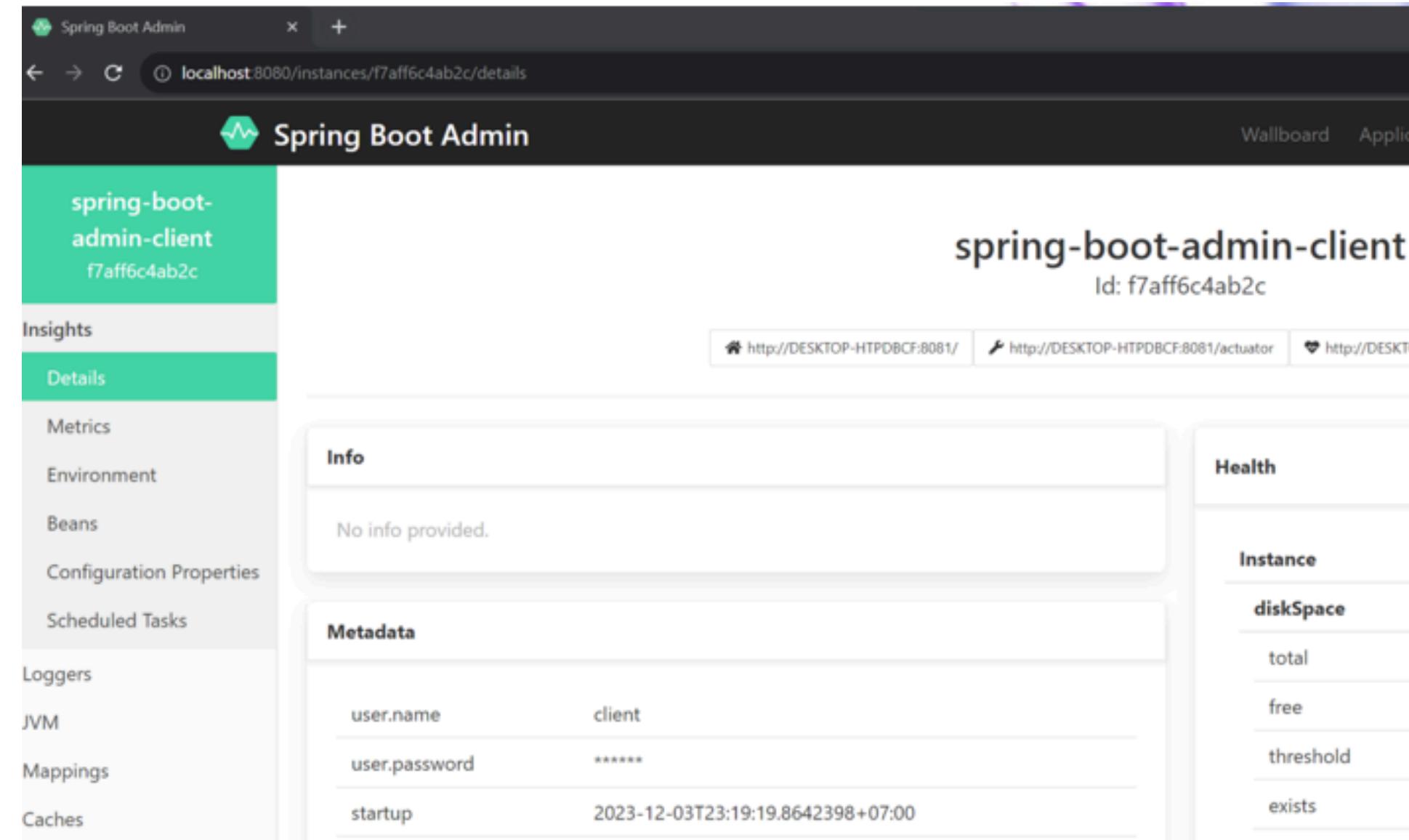


The screenshot shows the Spring Boot Admin main dashboard at [localhost:8080/applications](http://localhost:8080/applications). It displays the following information:

- APPLICATIONS:** 2
- INSTANCES:** 2
- STATUS:** all up

The list of applications includes:

- spring-boot-admin-client (2m ago, UP, <http://DESKTOP-HTPDBCF:8081/>)
- spring-boot-admin-server (4m ago, UP, <http://DESKTOP-HTPDBCF:8080/>)



The screenshot shows the Spring Boot Admin instance details page for the client application at [localhost:8080/instances/f7aff6c4ab2c/details](http://localhost:8080/instances/f7aff6c4ab2c/details). The sidebar menu is open, showing the following options:

- spring-boot-admin-client (f7aff6c4ab2c)
- Insights
- Details** (selected)
- Metrics
- Environment
- Beans
- Configuration Properties
- Scheduled Tasks
- Loggers
- JVM
- Mappings
- Caches

The main content area displays the following information for the **spring-boot-admin-client** instance:

- Info:** No info provided.
- Metadata:**

user.name	client
user.password	*****
startup	2023-12-03T23:19:19.8642398+07:00
- Health:** A sidebar showing disk space usage: total, free, threshold, exists.

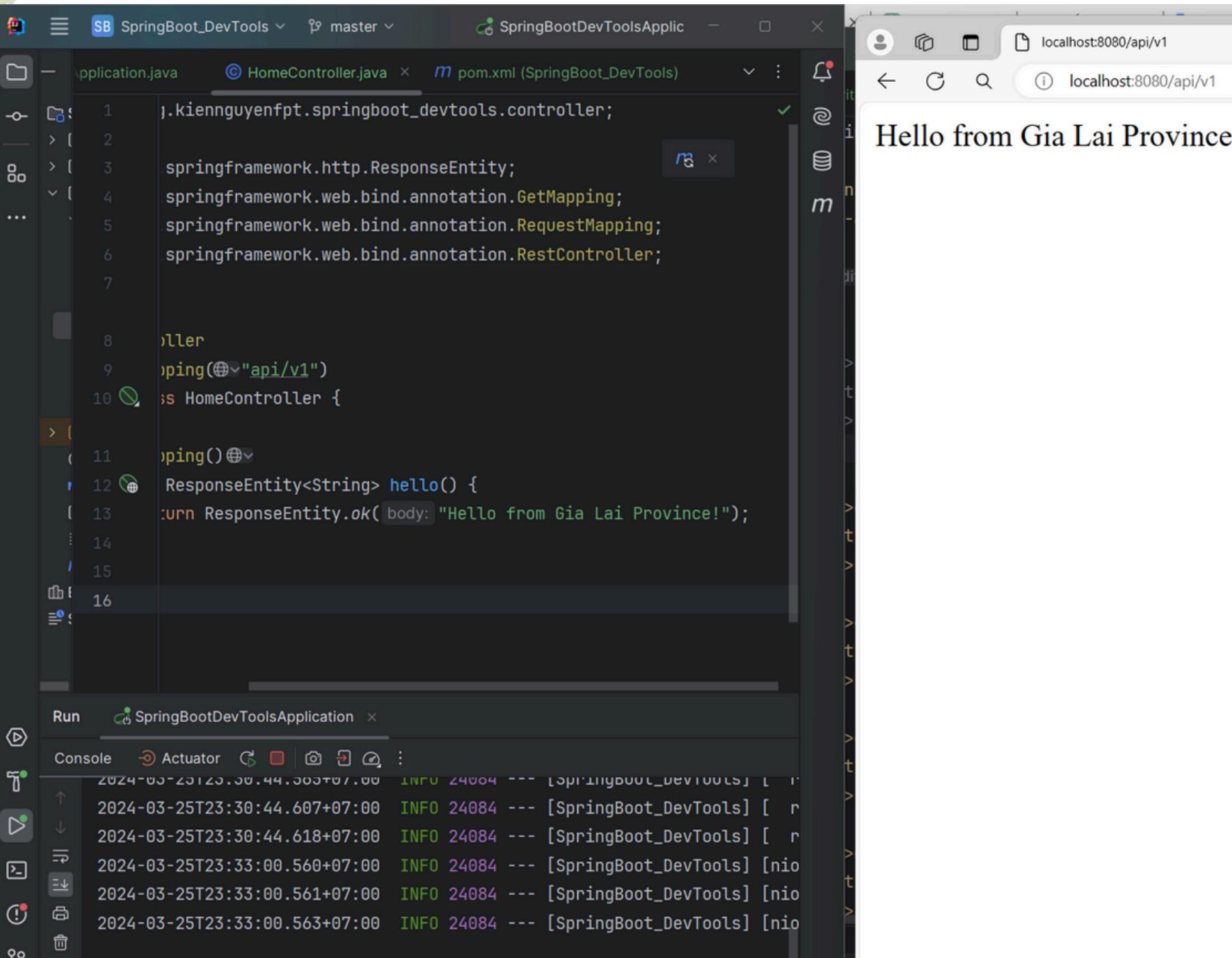
# Overview of Spring-Boot Dev Tools

- Spring Boot Dev Tools is a set of additional tools integrated into Spring Boot to make the development experience easier.
- If we make change to our source code, we need to restart application manually -> Spring Boot Dev Tools can help we to run application automatically

Add dependency:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
```

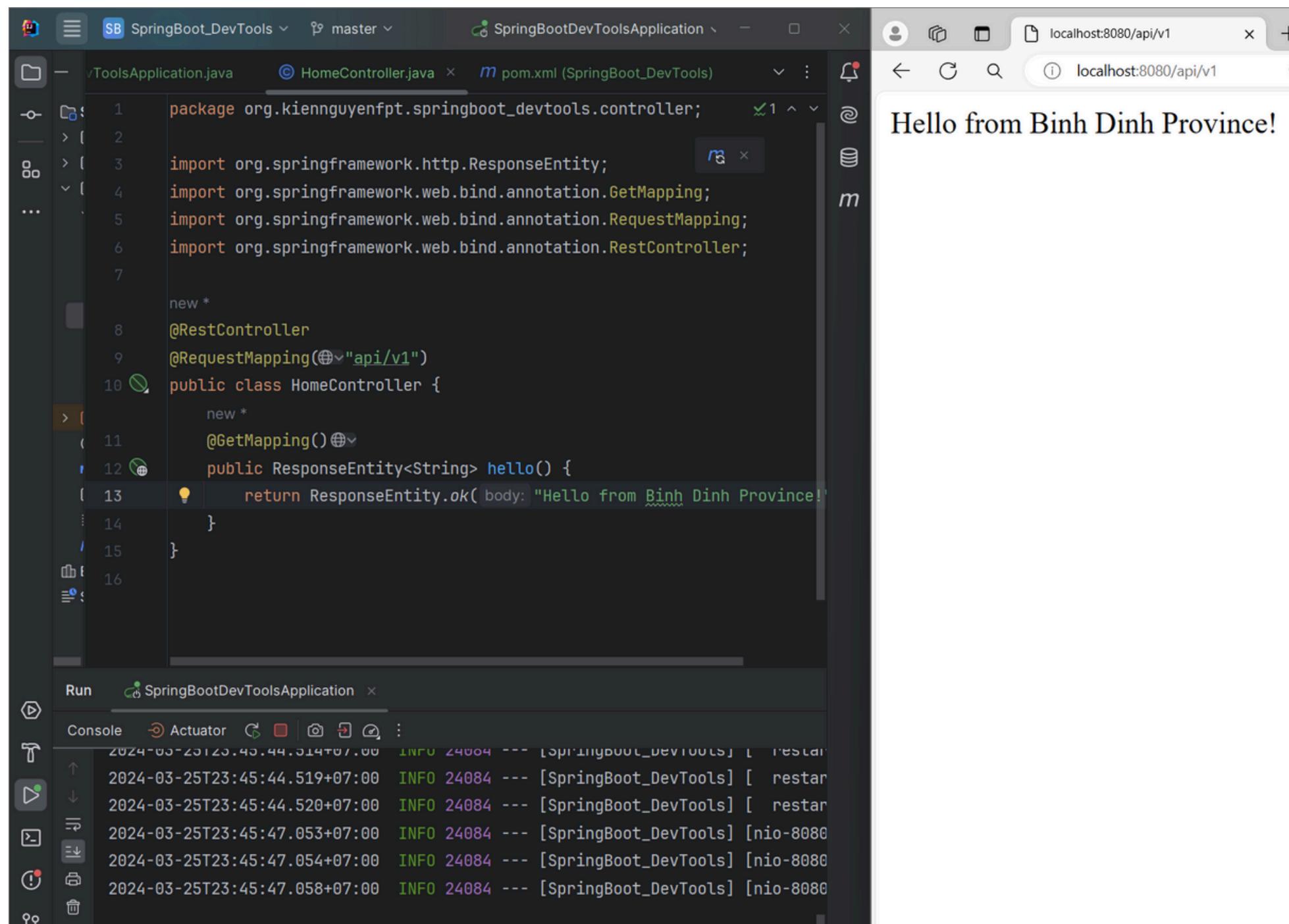
# Overview of Spring-Boot Dev Tools



```
1 package org.kiennguyenfpt.springboot_devtools.controller;
2
3 import org.springframework.http.ResponseEntity;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 @RestController
9 @RequestMapping("api/v1")
10 public class HomeController {
11
12     @GetMapping()
13     public ResponseEntity<String> hello() {
14         return ResponseEntity.ok("Hello from Gia Lai Province!");
15     }
16 }
```

localhost:8080/api/v1  
Hello from Gia Lai Province!

When we return to app, the application automatically run if we change our source code

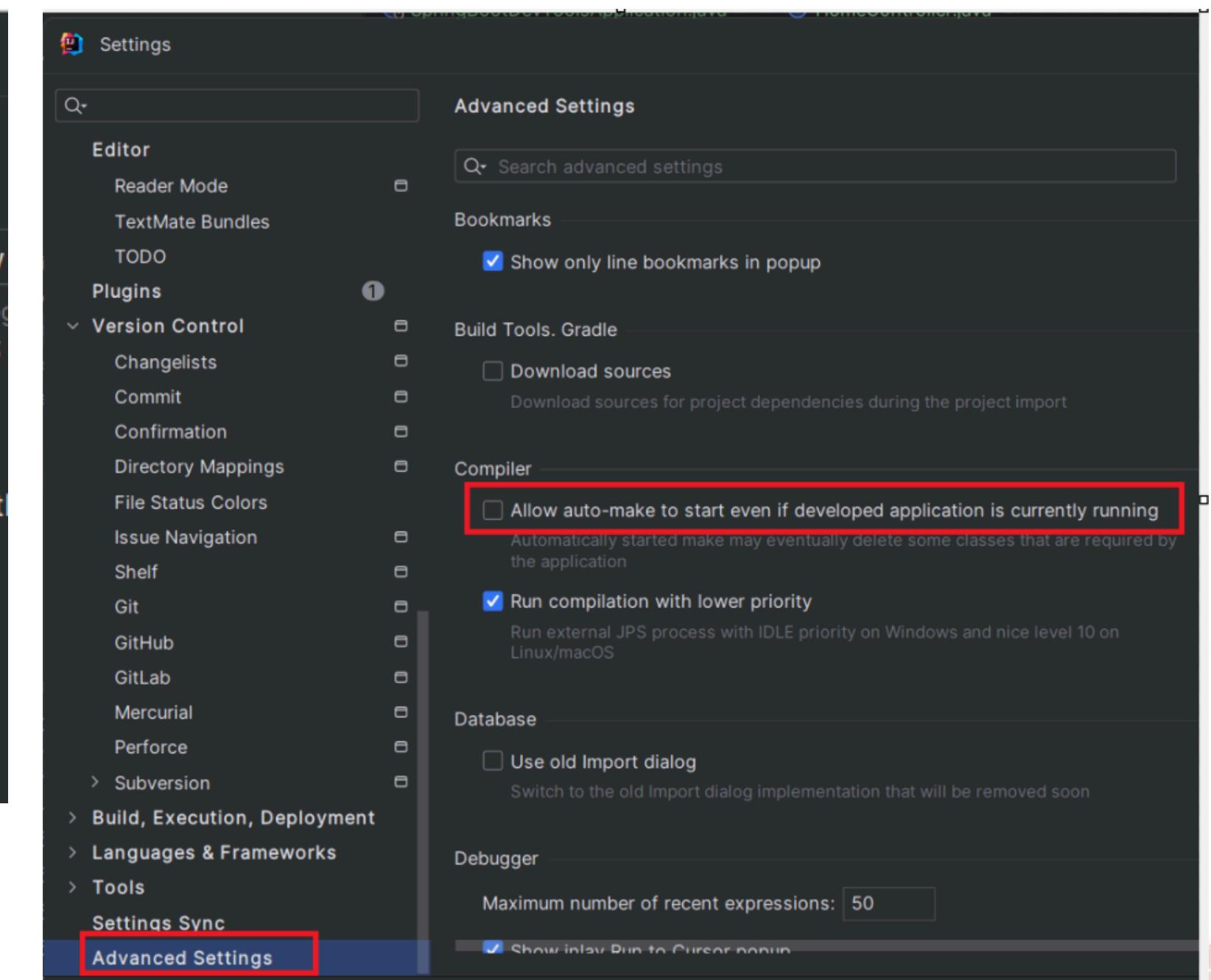
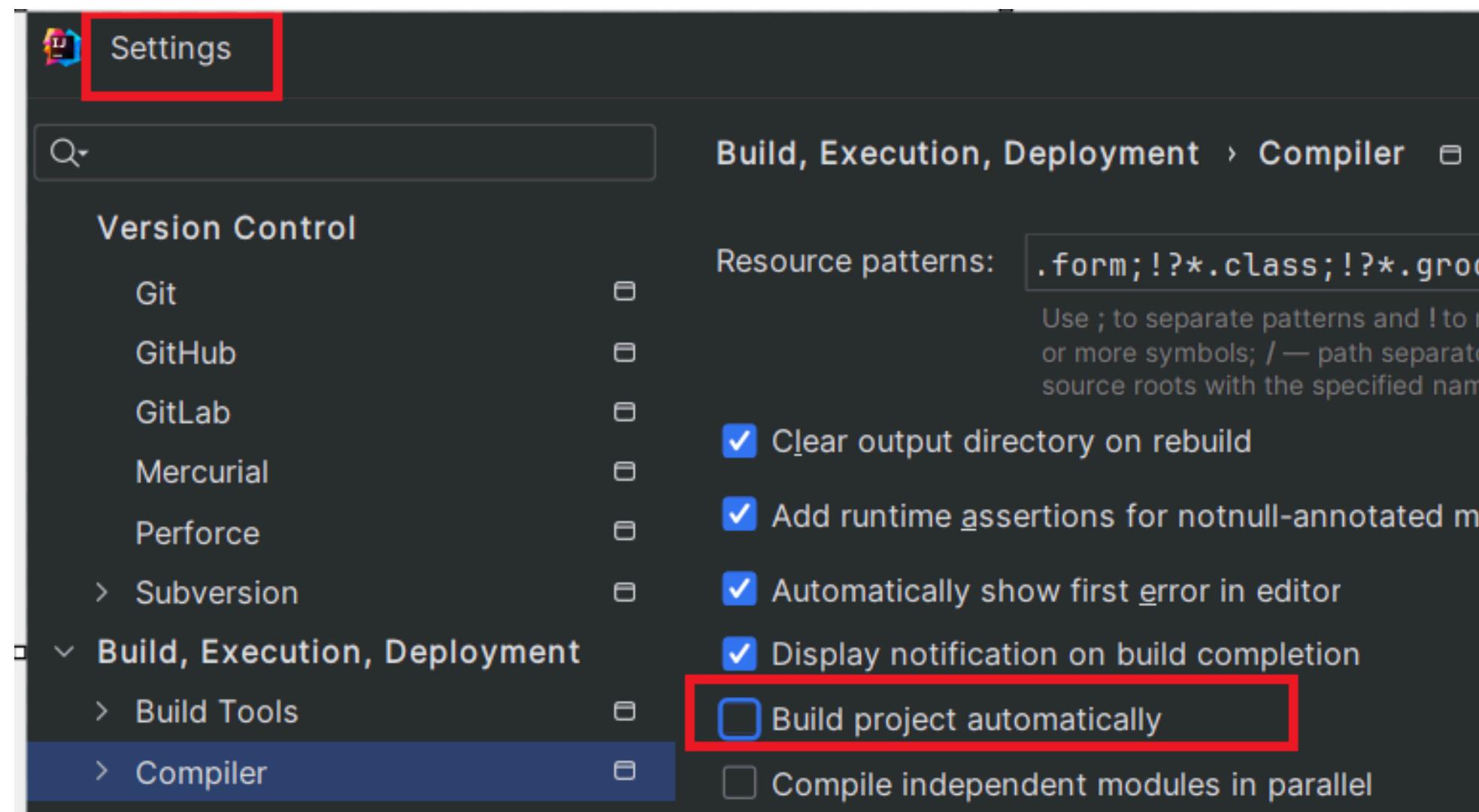


```
1 package org.kiennguyenfpt.springboot_devtools.controller;
2
3 import org.springframework.http.ResponseEntity;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RestController;
7
8 @RestController
9 @RequestMapping("api/v1")
10 public class HomeController {
11
12     @GetMapping()
13     public ResponseEntity<String> hello() {
14         return ResponseEntity.ok("Hello from Binh Dinh Province!");
15     }
16 }
```

localhost:8080/api/v1  
Hello from Binh Dinh Province!

# Overview of Spring-Boot Dev Tools

If we don't use Spring Boot Dev Tools, we can config it manually



# Overview of Spring-Boot Dev Tools

To disable live reload, we can add this configuration on file properties

```
application.properties ×  
1 server.port=8083  
2 spring.devtools.restart.enabled=false
```

```
@RestController  
@RequestMapping("api/v1")  
public class HomeController {  
    new *  
    @GetMapping()  
    public String hello() {  
        return "Hello Mentor <3";  
    }  
}
```

But the result in localhost:



# Introduction to Spring Boot CLI

- Spring Boot CLI (**Command-Line Interface**) is a command-line tool provided by Spring Boot to create, run, and test Spring Boot applications quickly and easily.
- CLI helps developers create Spring Boot applications without having to use IDEs or complicated tools.

## The way to use Spring Boot CLI:

- 1) Download SDKMAN
- 2) In SDKMAN, Install spring boot CLI:

```
$ sdk install springboot 2.7.18
springboot 2.7.18 is already installed.
```

# Introduction to Spring Boot CLI

3) Check version:

```
$ spring --version  
Spring CLI v2.7.18
```

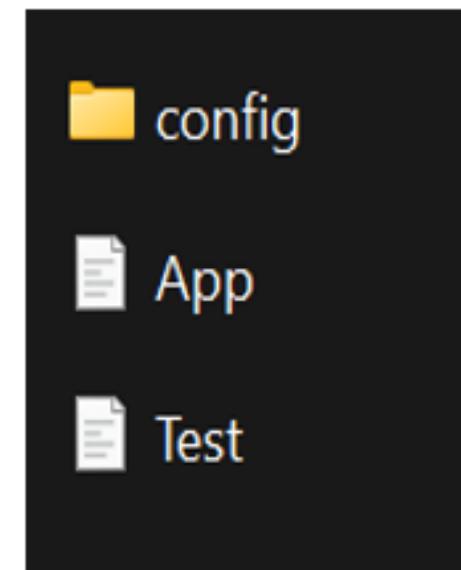
To start the embedded shell, we run: **spring shell**

From here, we can directly enter desired commands without the **spring** keyword

```
$ version  
Spring CLI v2.7.18
```

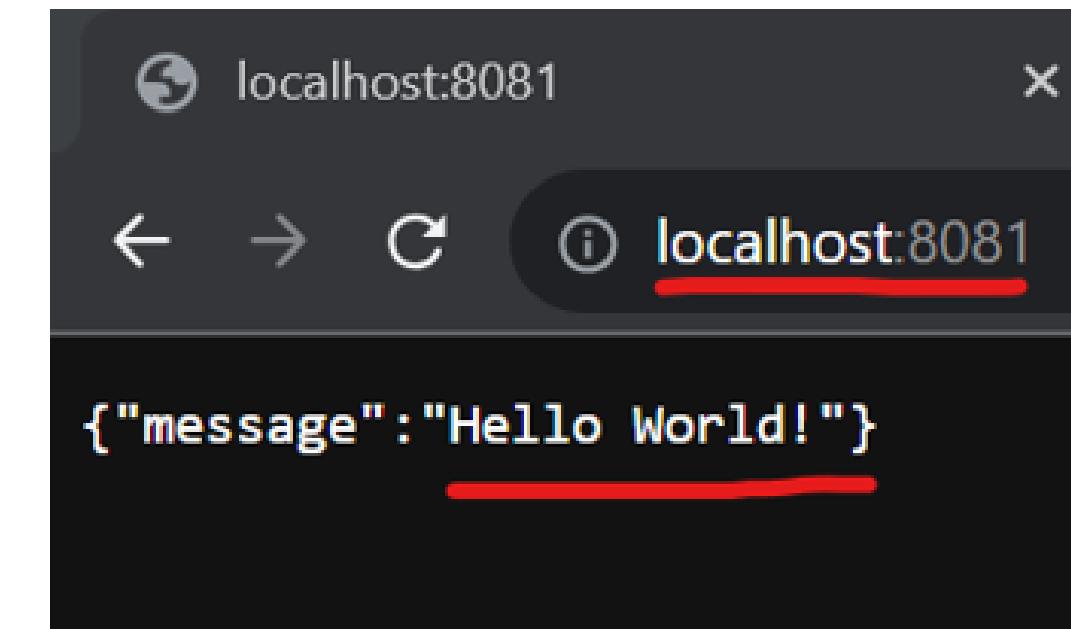
Check version again

Create files and folder in the **groovy** directory:



Change the directory to groovy directory in SDKMAN and run the application using this command:

## Result



# Spring Boot Application as a Service

- Spring Boot App as a Service refers to running a Spring Boot application continuously in the background as a service, independent of user.
- This means that the application will operate without direct interaction from the user.

# Spring Boot Application as a Service

```
@RestController  
@RequestMapping("api/v1")  
public class HomeController {  
    new *  
    @GetMapping("/home")  
    public ModelAndView hello() { return new ModelAndView( "index"); }  
}
```

```
<finalName>springboot-devops</finalName>  
<plugins>  
    <plugin>  
        <groupId>org.springframework.boot</groupId>  
        <artifactId>spring-boot-maven-plugin</artifactId>  
        <configuration>  
            <excludes>  
                <executable>true</executable>  
                <exclude>  
                    <groupId>org.projectlombok</groupId>  
                    <artifactId>lombok</artifactId>  
                </exclude>  
            </excludes>  
        </configuration>  
    </plugin>  
</plugins>
```

Config in file pom.xml

# Spring Boot Application as a Service

 [springboot-devops.jar](#)

 [WinSW.NET4.exe](#)

 [WinSW.NET4.wrapper.log](#)

 [WinSW.NET4.xml](#)

```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd ..

C:\Windows>cd ..

C:\>cd springboot-devops

C:\springboot-devops>dir
 Volume in drive C is OS
 Volume Serial Number is DA65-CCCA

 Directory of C:\springboot-devops

03/26/2024  09:39 PM    <DIR>
03/26/2024  09:26 PM      22,685,313 springboot-devops.jar
03/26/2024  09:37 PM        852,480 WinSW.NET4.exe
03/26/2024  09:44 PM        2,253 WinSW.NET4.xml
               3 File(s)     23,540,046 bytes
               1 Dir(s)   90,384,011,264 bytes free

C:\springboot-devops>WinSW.NET4.exe install
2024-03-26 21:46:28,765 INFO  - Installing service 'springboot-app (powered by WinSW) (springboot-devops)'...
2024-03-26 21:46:28,794 INFO  - Service 'springboot-app (powered by WinSW) (springboot-devops)' was installed successfully.
```

# Spring Boot Application as a Service

Services

File Action View Help

Services (Local)

Services (Local)

Name	Description	Status	Startup Type	Log On As
springboot-app (powered by WinSW)	This service is a service created from a minimal configuration	Running	Automatic	Local System
<a href="#">Start the service</a>				
Security Accounts Manager	The startup ...	Running	Automatic	Local System
Security Center	The WSCSVC...	Running	Automatic (De...	Local Service
Sensor Data Service	Delivers dat...		Manual (Trigg...	Local System
Sensor Monitoring Service	Monitors va...		Manual (Trigg...	Local Service
Sensor Service	A service for ...		Manual (Trigg...	Local System
Server	Supports file...	Running	Automatic (Tri...	Local System
Shared PC Account Manager	Manages pr...		Disabled	Local System
Shell Hardware Detection	Provides not...	Running	Automatic	Local System
Smart Card	Manages ac...		Manual (Trigg...	Local Service
Smart Card Device Enumerator	Creates soft...		Manual (Trigg...	Local System
Smart Card Removal Policy	Allows the s...		Manual	Local System
SNMP Trap	Receives tra...		Manual	Local Service
Software Protection	Enables the ...		Automatic (De...	Network Se...
Spatial Data Service	This service i...		Manual	Local Service
Spot Verifier	Verifies pote...		Manual (Trigg...	Local System
springboot-app (powered b...)	This service i...	Automatic	Local System	
SQL Full-text Filter Daemon ...	Service to la...	Manual	NT Service\...	
SQL Server (MSSQLSERVER)	Provides sto...	Automatic (De...	NT Service\...	
SQL Server (MSSQL\$SFRVFR01)	Provides sto...	Running	Automatic (De...	NT Service\...

## Result:

localhost:8080/api/v1/home

KIOT WEBSITE

Welcome To Kiot Hotel

LUXURY - MODERN - CONVENIENT

CHECK-IN  
Check-in date

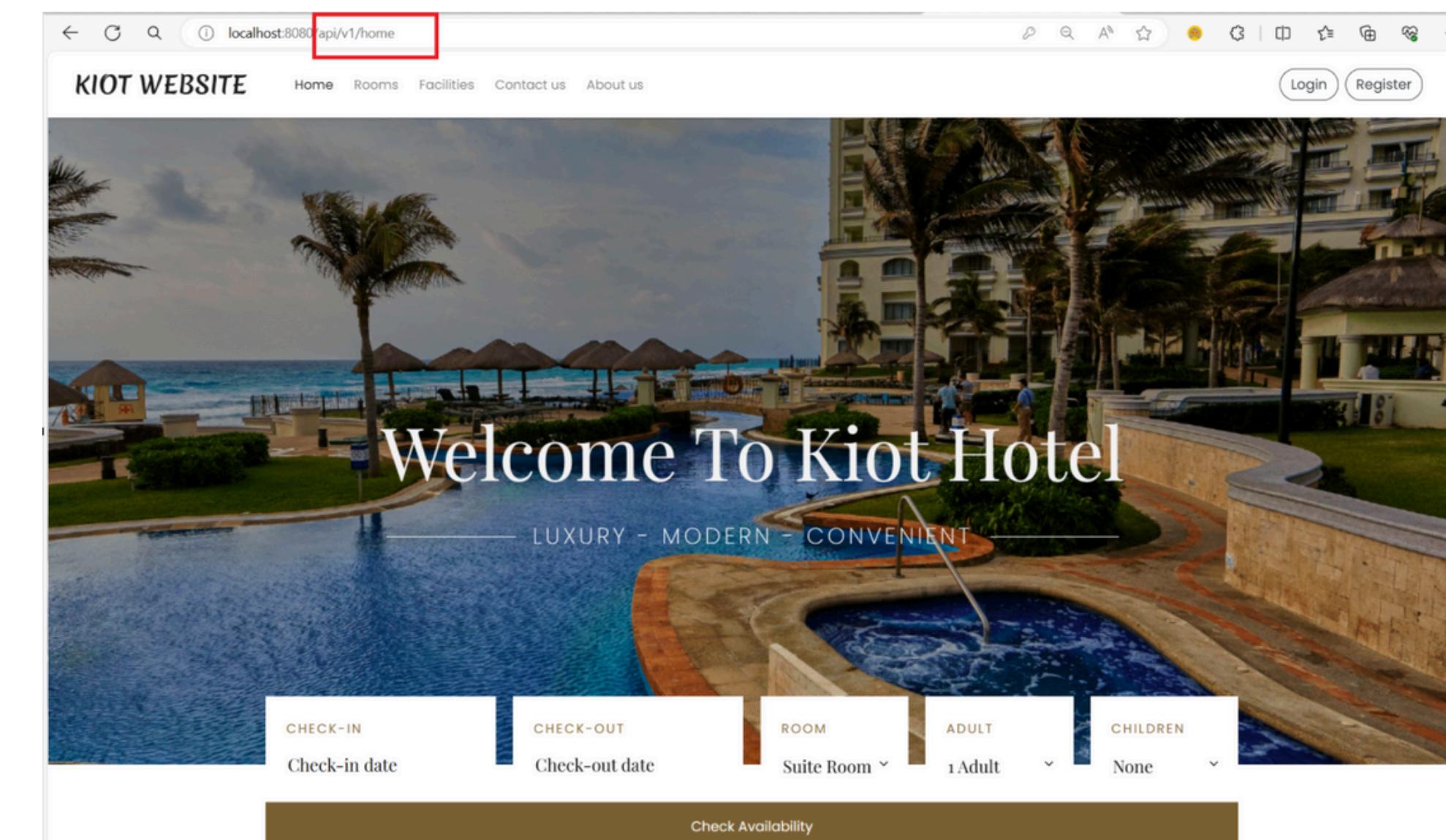
CHECK-OUT  
Check-out date

ROOM  
Suite Room

ADULT  
1 Adult

CHILDREN  
None

Check Availability



# Spring Boot Gradle Plugin

- Plugins are an extensible component that is integrated into an application or system to provide additional, customized functionality.
- In Spring Boot, plugins help us add features like dependency management, application packaging, and running Spring Boot applications from Gradle. A plugin can provide construction, configuration, and resource management tasks.

## Functions of Spring Boot Gradle Plugin:

- Package executable file (jar or war): Plugin allows you to package Spring Boot applications into executable files that can be run directly.
- Run Spring Boot application
- Dependency management: The plugin uses dependencies provided by Spring Boot dependencies to manage dependency versions.

# Spring Boot Gradle Plugin

```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.2.4'  
    id 'io.spring.dependency-management' version '1.1.4'  
}
```

```
dependencies { Edit Starters...  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    compileOnly 'org.projectlombok:lombok'  
    annotationProcessor 'org.projectlombok:lombok'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}
```

In the above example:

- + 'java' plugin is applied to build Java application,
- + 'org.springframework.boot' version '3.2.4': This is the Spring Boot version
- + id 'io.spring.dependency-management' version '1.1.4': This is Spring's dependency management plugin and we use **spring-boot-starter-web** as dependency.

# Spring Boot Gradle Plugin

## Build file .jar in Gradle project

```
PS D:\Fsoft\SpringBoot_Gradle> ./gradlew build
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

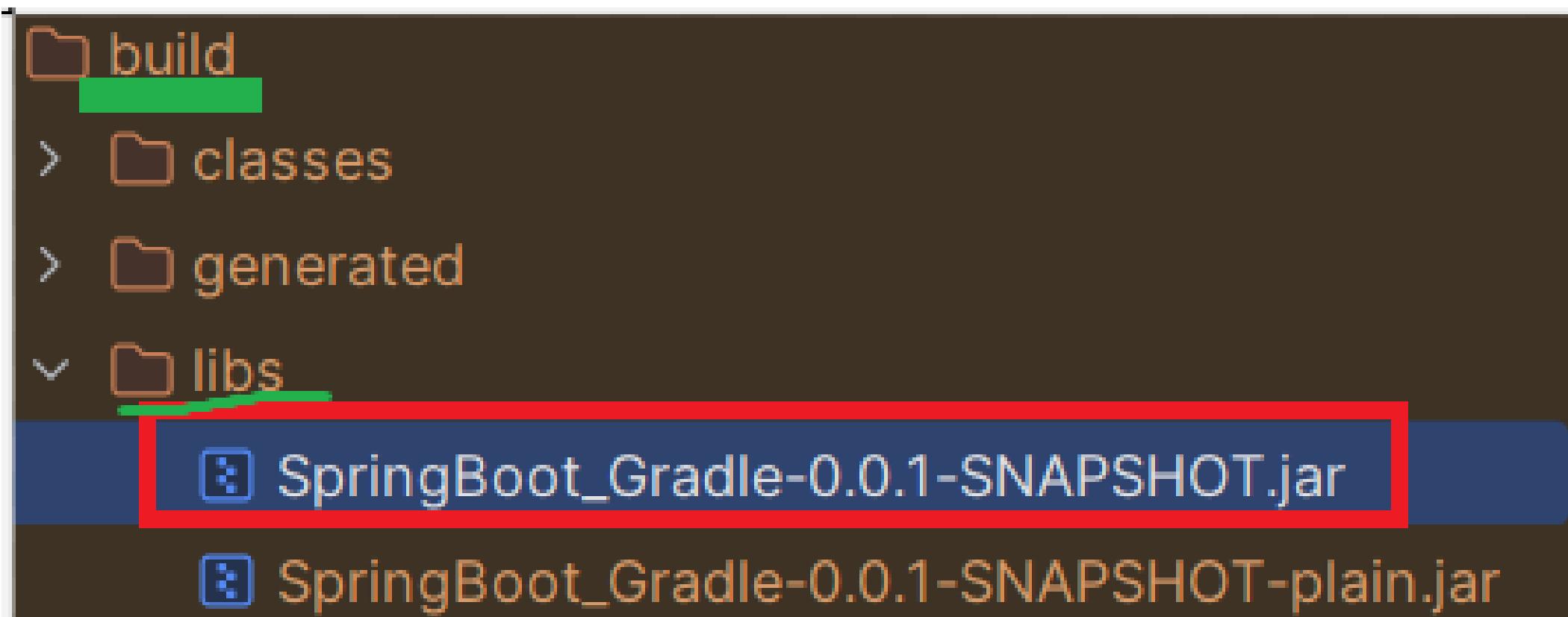
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.6/userguide/command\_line\_interface.html#sec:command\_line\_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 17s
```

After build successfully, file .jar will be located on folder **build/libs**



# Spring Boot Gradle Plugin

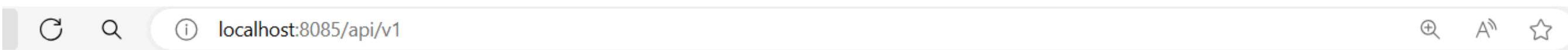
# Run file .jar

```
PS D:\Fsoft\SpringBoot_Gradle\build\libs> java -jar SpringBoot_Gradle-0.0.1-SNAPSHOT.jar

.
-----
/\ \ / ____'_- _ _ _(_)_ __ _ _ _ _ \ \ \ \
( ( )\___ | ' _ | '_| | '_ \ \_` | \ \ \ \
\ \ \ \_)| |_)| | | | | | | ( | | ) ) ) )
' |_____| .__|_|_|_|_|_\__, | / / /
=====|_|=====|_|/_=/_|/_/_
:: Spring Boot ::          (v3.2.4)

2024-03-28T15:12:27.256+07:00  INFO 30732 --- [SpringBoot_Gradle] [           main] o.k.s.SpringBootGradleApplication : Starting SpringBootGradleApplication v0.0.1-SNAPSHOT using Java 17.0.9 with PID 30732 (D:\Fsoft\SpringBoot_Gradle\build\libs\SpringBoot_Gradle-0.0.1-SNAPSHOT.jar started by TRUNG KIEN in D:\Fsoft\SpringBoot_Gradle\build\libs)
2024-03-28T15:12:27.272+07:00  INFO 30732 --- [SpringBoot_Gradle] [           main] o.k.s.SpringBootGradleApplication : No active profile set, falling back to 1 default profile: "default"
2024-03-28T15:12:29.231+07:00  INFO 30732 --- [SpringBoot_Gradle] [           (http)           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8085
2024-03-28T15:12:29.258+07:00  INFO 30732 --- [SpringBoot_Gradle] [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-03-28T15:12:29.259+07:00  INFO 30732 --- [SpringBoot_Gradle] [           main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-03-28T15:12:29.357+07:00  INFO 30732 --- [SpringBoot_Gradle] [applicationContext           main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2024-03-28T15:12:29.360+07:00  INFO 30732 --- [SpringBoot_Gradle] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialized and ready!
2024-03-28T15:12:29.983+07:00  INFO 30732 --- [SpringBoot_Gradle] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8085 (http://0.0.0.0:8085)
```

## Check result:



# Hello FSOFT from Gradle project!

# Spring Boot Gradle Plugin

Create file .war

```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.2.4'  
    id 'io.spring.dependency-management' version '1.1.4'  
    id 'war'  
}
```

```
PS D:\Fsoft\SpringBoot_Gradle> ./gradlew build
```

After build successfully:

```
SpringBoot_Gradle-0.0.1-SNAPSHOT.jar  
SpringBoot_Gradle-0.0.1-SNAPSHOT.war  
SpringBoot_Gradle-0.0.1-SNAPSHOT-plain.jar  
SpringBoot_Gradle-0.0.1-SNAPSHOT-plain.war
```

## Run file .war

```
PS D:\Fsoft\SpringBoot_Gradle\build\libs> java -jar SpringBoot_Gradle-0.0.1-SNAPSHOT.war

.   ----
/\ / _____.(_)_ -- _ \ \ \
( )\___| '_ \| '_ \| \_` | \ \ \
\_\_||_| \_|| || || (_| \_) )
' |____| .__|_|_|_|_| \_,, | / / /
=====|_|=====|_|=/_/_/_/
:: Spring Boot ::           (v3.2.4)

2024-03-28T15:32:47.697+07:00  INFO 26556 --- [SpringBoot_Gradle] [main] o.k.s.SpringBootGradleApplication : Starting SpringBootGradleApplication v0.0.1-SNAPSHOT using Java 17.0.9 with PID 26556 (D:\Fsoft\SpringBoot_Gradle\build\libs\SpringBoot_Gradle-0.0.1-SNAPSHOT.war started by TRUNG KIEN in D:\Fsoft\SpringBoot_Gradle\build\libs)
2024-03-28T15:32:47.705+07:00  INFO 26556 --- [SpringBoot_Gradle] [main] o.k.s.SpringBootGradleApplication : No active profile set, falling back to 1 default profile: "default"
2024-03-28T15:32:49.144+07:00  INFO 26556 --- [SpringBoot_Gradle] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8085
```

To run project, we use `./gradlew bootRun`

```
PS D:\Fsoft\SpringBoot_Gradle> ./gradlew bootRun

> Task :bootRun

.   ----
/\ / _____.(_)_ -- _ \ \ \
( )\___| '_ \| '_ \| \_` | \ \ \
\_\_||_| \_|| || || (_| \_) )
' |____| .__|_|_|_|_| \_,, | / / /
=====|_|=====|_|=/_/_/_/
:: Spring Boot ::           (v3.2.4)

2024-03-28T15:42:29.732+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.k.s.SpringBootGradleApplication : Starting SpringBootGradleApplication using Java 17.0.9 with PID 11000 (D:\Fsoft\SpringBoot_Gradle\build\classes\java\main started by TRUNG KIEN in D:\Fsoft\SpringBoot_Gradle)
2024-03-28T15:42:29.735+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.k.s.SpringBootGradleApplication : No active profile set, falling back to 1 default profile: "default"
2024-03-28T15:42:30.569+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8085 (http)
2024-03-28T15:42:30.583+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-03-28T15:42:30.583+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]
2024-03-28T15:42:30.646+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2024-03-28T15:42:30.647+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 861 ms
2024-03-28T15:42:30.928+07:00  INFO 11000 --- [SpringBoot_Gradle] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8085 (http)
```

# Spring Boot Gradle Plugin

```
dependencies { Edit Starters...
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compileOnly 'org.projectlombok:lombok'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}
```

```
@RestController
@RequestMapping("api/v1")
public class HomeController {
    new *
    @GetMapping()
    public String greeting() {
        return "Hello mentor FSoft from Gradle project!";
    }
}
```



localhost:8080/api/v1

Hello mentor FSoft from Gradle project!

```
dependencies { Edit Starters...
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compileOnly 'org.projectlombok:lombok'
    developmentOnly 'org.springframework.boot:spring-boot-devtools' (highlighted)
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}
```

```
@RestController
@RequestMapping("api/v1")
public class HomeController {
    new *
    @GetMapping()
    public String greeting() {
        return "Hello mentor HaiNV21 from Gradle project!";
    }
}
```



localhost:8080/api/v1

Hello mentor HaiNV21 from Gradle project!

FPT SOFTWARE QUY NHON

**Greatful Thank You for  
your listening!**

KienNCT3 - Mentor HaiNV21