**Digital Nurture 4.0**

**Week 4 Spring REST using Spring Boot 3**

# **Mandatory HandsOn**

# **File name: 1. spring-rest-handson**

**1.Create a Spring Web Project using Maven**

**Program:**

* Go to: <https://start.spring.io/>
* Configured:

**i.Group**: com.cognizant

**ii.Artifact**: spring-learn

**iii.Dependencies**: Spring Web, Spring Boot DevTools

vi.Clicked **Generate** and downloaded the ZIP.

v.Extracted the ZIP .

* Opened IntelliJ,**Open** the Selected the extracted folder.
* While opening the folder it will ask to import the maven , click import.
* The pom.xml and other packages are created and configured automatically.
* Create a file in src/main/java/com/cognizant/SpringLearnApplication.java.

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 System.*out*.println("START");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 System.*out*.println("END");  
 }  
  
}

* Run the SpringLearnApplication.java.

**Output:**

****

**2. Spring Core – Load Country from Spring**

**Configuration XMLProgram:**

* Use the same springinitializer configuration.
* Create a XML Configuration file in src/main/resources/Country.xml

Step 1: Create the country.xml file

**country.xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Individual Country Beans -->

<bean id="in" class="com.cognizant.spring\_learn.model.Country">

<property name="name" value="India"/>

</bean>

<bean id="us" class="com.cognizant.spring\_learn.model.Country">

<property name="name" value="USA"/>

</bean>

<bean id="de" class="com.cognizant.spring\_learn.model.Country">

<property name="name" value="Germany"/>

</bean>

<bean id="jp" class="com.cognizant.spring\_learn.model.Country">

<property name="name" value="Japan"/>

</bean>

<!-- List Bean -->

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="de"/>

<ref bean="jp"/>

</list>

</constructor-arg>

</bean>

</beans>

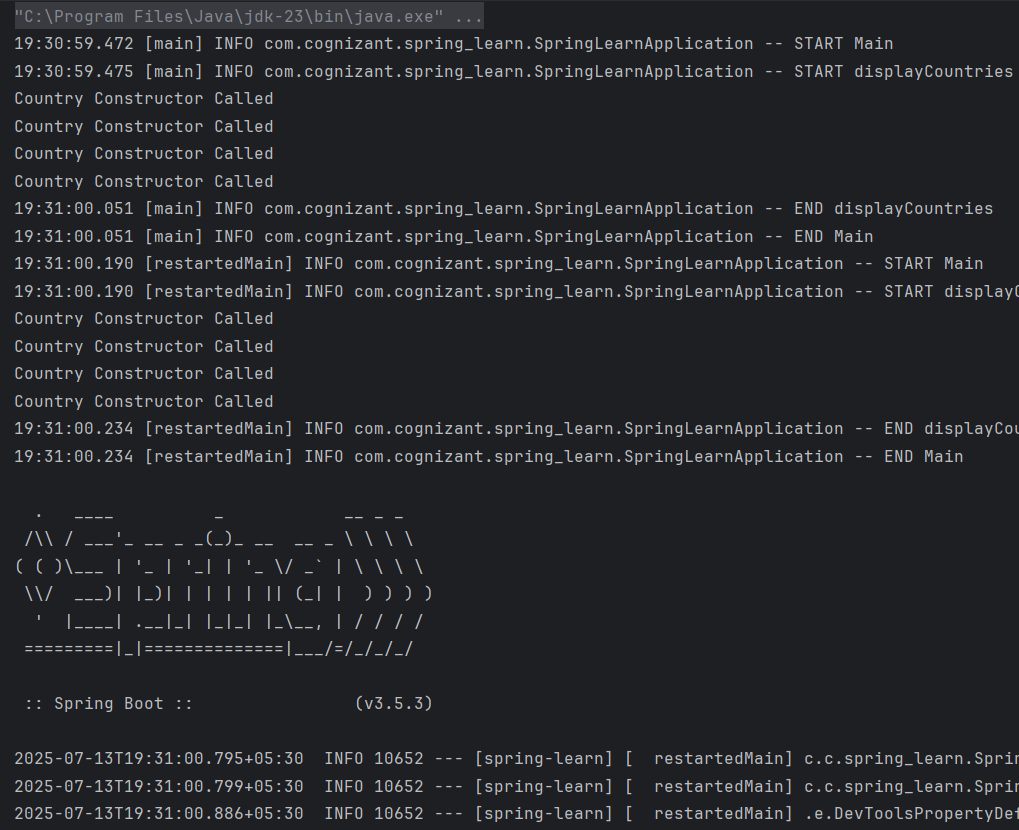
* pom.xml already have the required dependencies.
* Update the maven .
* Modify the main class in src/main/java/com/cognizant/spring\_learn/SpringLearnApplication.java

**SpringLearnApplication.java:**

package com.cognizant;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) throws Exception {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 *displayDate*();  
 }  
 public static void displayDate() throws Exception {  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
 Date parsedDate = format.parse("31/12/2018");  
 System.*out*.println(parsedDate);  
 }  
}

* Run the SpringLearnApplication.java

**Output:**

****

# **File name: 2. spring-rest-handson**

# **3.Hello World RESTful Web Service**

**Program:**

* Use the same springinitializer configuration.
* Set the server port in src/main/resources/application.properties.

**Step 1: Create Controller Class**

**HelloController.java**

package com.cognizant.spring\_learn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

@GetMapping("/hello")

public String sayHello() {

LOGGER.info("START sayHello()");

String message = "Hello World!!";

LOGGER.info("END sayHello()");

return message;

}

}

**Application.properties:**

server.port=8083  
logging.level.root=INFO  
logging.level.com.cognizant.spring\_learn=DEBUG  
spring.application.name=spring-learn

* Create a controller package in com.cognizant.spring\_learn.controller package.
* Create a file named HelloController.java in src/main/java/com/cognizant/spring\_learn/controller/HelloController.java
* Modify the SpringLearnApplication.java.

**SpringLearnApplication.java:**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

public static void main(String[] args) {

LOGGER.info("START - SpringLearnApplication");

SpringApplication.run(SpringLearnApplication.class, args);

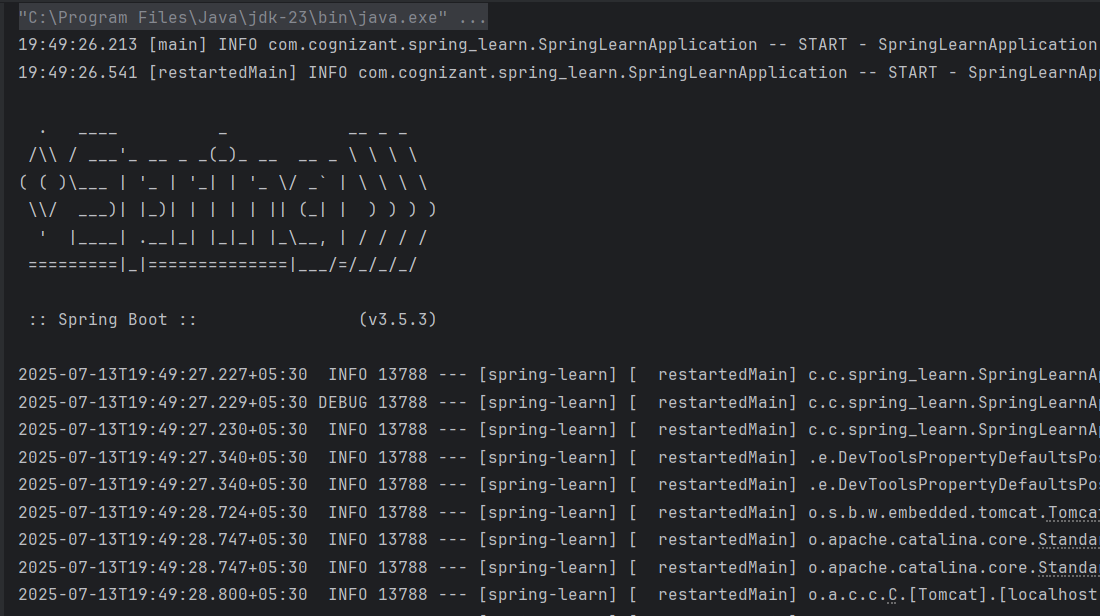
LOGGER.info("END - SpringLearnApplication");

}

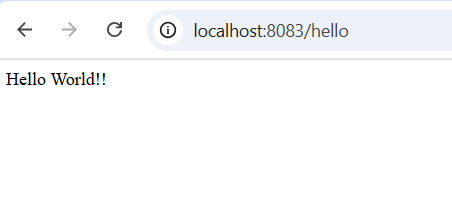
}

* No need to modify the pom.xml.
* Run the SpringLearnApplication.java and run the url <http://localhost:8083/hello> in the browser.

**Output:**

****

**In Browser:**

****

# **4.REST - Country Web Service**

**Program:**

* Use the same springinitializer configuration.
* Create the XML Configuration file in src/main/resources/ applicationContext.xml

**applicationContext.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- India Bean -->

<bean id="in" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

* </beans>Create the Country model class in src/main/java/com/cognizant/spring\_learn/model/Country.java

**Country.java:**

package com.cognizant.spring\_learn.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 // Constructors  
 public Country() {}  
  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = name;  
 }  
  
 // Getters and Setters  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 // toString()  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

* Create CountryController.java in src/main/java/com/cognizant/spring\_learn/controller/CountryController.java

**CountryController.java:**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.model.Country;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
@RestController  
public class CountryController {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryController.class);  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 *LOGGER*.info("START - getCountryIndia()");  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 Country country = (Country) context.getBean("in");  
  
 *LOGGER*.info("END - getCountryIndia()");  
 return country;  
 }  
}

* No need to make changes in the SpringLearnApplication.java,application.properties.
* Run the SpringLearnApplication.java and the url <http://localhost:8083/country> in the browser.

**Output:**

**A screen shot of a computer

AI-generated content may be incorrect.**

**In browser:**

**A screenshot of a computer

AI-generated content may be incorrect.**

# **5. REST - Get country based on country code**

**Program:**

* Use the same springinitializer configuration.
* Update the applicationContext.xml

**applicationContext.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Define countries -->

<bean id="in" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="IN"/>

<property name="name" value="India"/>

</bean>

<bean id="us" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

<bean id="jp" class="com.cognizant.spring\_learn.model.Country">

<property name="code" value="JP"/>

<property name="name" value="Japan"/>

</bean>

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="jp"/>

</list>

</constructor-arg>

</bean>

</beans>

* Create new package named service and service class named CountryService.java in src/main/java/com/cognizant/spring\_learn/service/CountryService.java

**CountryService.java:**

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.model.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

List<Country> countries = (List<Country>) context.getBean("countryList");

return countries.stream()

.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null); // Return null if not found

}

}

* Update the CountryCountroller.java

**CountryController.java:**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.model.Country;

import com.cognizant.spring\_learn.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

@RestController

public class CountryController {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

@Autowired

private CountryService countryService;

@GetMapping("/countries/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START - getCountry(): {}", code);

Country country = countryService.getCountry(code);

LOGGER.info("END - getCountry(): {}", country);

return country;

}

}

No need to make changes in the application.properties,SpringLearnApplication.java and pom.xml.

**Country.java:**

package com.cognizant.spring\_learn.model;

public class Country {

private String code;

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

* Run the SpringLearnApplication and the urls <http://localhost:8083/countries/in>

**Output:**

**A screen shot of a computer

AI-generated content may be incorrect.**

**In browser:**

A screenshot of a computer

AI-generated content may be incorrect.

# **File name: 5. JWT-handson**

# **6. Create authentication service that returns JWT**

**Program:**

* Spring Initializer setup.
* **Project**: Maven
* **Language**: Java
* **Spring Boot**: 3.1+ or 3.5.3
* **Group**: com.cognizant
* **Artifact**: jwt-auth
* **Name**: jwt-auth
* **Package name**: com.cognizant.jwt\_auth
* **Packaging**: Jar
* **Java Version**: 17 or 21
* Update the pom.xml

**Pom.xml:**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.5.3</version>  
 <relativePath/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>jwt-auth</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>jwt-auth</name>  
 <description>JWT Auth Project</description>  
  
 <properties>  
 <java.version>24</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web and Security -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
  
 <!-- JSON Web Token - JJWT -->  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-api</artifactId>  
 <version>0.11.5</version>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-impl</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-jackson</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!-- Servlet API -->  
 <dependency>  
 <groupId>jakarta.servlet</groupId>  
 <artifactId>jakarta.servlet-api</artifactId>  
 <version>6.0.0</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!-- Development and Testing -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

* Create package named jwt\_auth and config,controller
* Path src/main/java/com/cognizant/jwt\_auth/controller, src/main/java/com/cognizant/jwt\_auth/config.
* Then create a class named AuthenticationController.java in src/main/java/com/cognizant/jwt\_auth/controller/AuthenticationController.java.

**AuthenticationController.java:**

package com.cognizant.jwt\_auth.controller;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import io.jsonwebtoken.security.Keys;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
import jakarta.servlet.http.HttpServletRequest;  
import java.security.Key;  
import java.util.Base64;  
import java.util.Date;  
import java.util.HashMap;  
import java.util.Map;  
  
@RestController  
public class AuthenticationController {  
  
 private final Key secretKey = Keys.*hmacShaKeyFor*("ThisIsASecretKeyThatIsAtLeast32Chars!".getBytes());  
  
 @GetMapping("/authenticate")  
 public Map<String, String> authenticate(HttpServletRequest request) {  
 String authHeader = request.getHeader("Authorization");  
  
 if (authHeader == null || !authHeader.startsWith("Basic ")) {  
 throw new RuntimeException("Missing or invalid Authorization header");  
 }  
  
 String base64Credentials = authHeader.substring("Basic ".length());  
 String credentials = new String(Base64.*getDecoder*().decode(base64Credentials));  
 String[] values = credentials.split(":", 2);  
  
 String username = values[0];  
 String password = values[1];  
  
 if (!username.equals("user") || !password.equals("pwd")) {  
 throw new RuntimeException("Invalid credentials");  
 }  
  
 String token = Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.*currentTimeMillis*() + 600000)) // 10 minutes  
 .signWith(secretKey, SignatureAlgorithm.*HS256*)  
 .compact();  
  
 Map<String, String> response = new HashMap<>();  
 response.put("token", token);  
 return response;  
 }  
}

* Security configuration named SecurityConfig.java in src/main/java/com/cognizant/jwt\_auth/config/SecurityConfig.java

**SecurityConfig.java:**

package com.cognizant.jwt\_auth.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetails;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
@EnableWebSecurity  
public class SecurityConfig {  
  
 @Bean  
 public UserDetailsService userDetailsService() {  
 UserDetails user = User  
 .*withUsername*("user")  
 .password("{noop}pwd") // {noop} means no encoding  
 .roles("USER")  
 .build();  
 return new InMemoryUserDetailsManager(user);  
 }  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").authenticated()  
 .anyRequest().permitAll()  
 )  
 .httpBasic(); // enables basic auth  
 return http.build();  
 }  
}

* Add the port number in application.properties.

**Application.properties:**

spring.application.name=jwt-auth  
server.port=8080

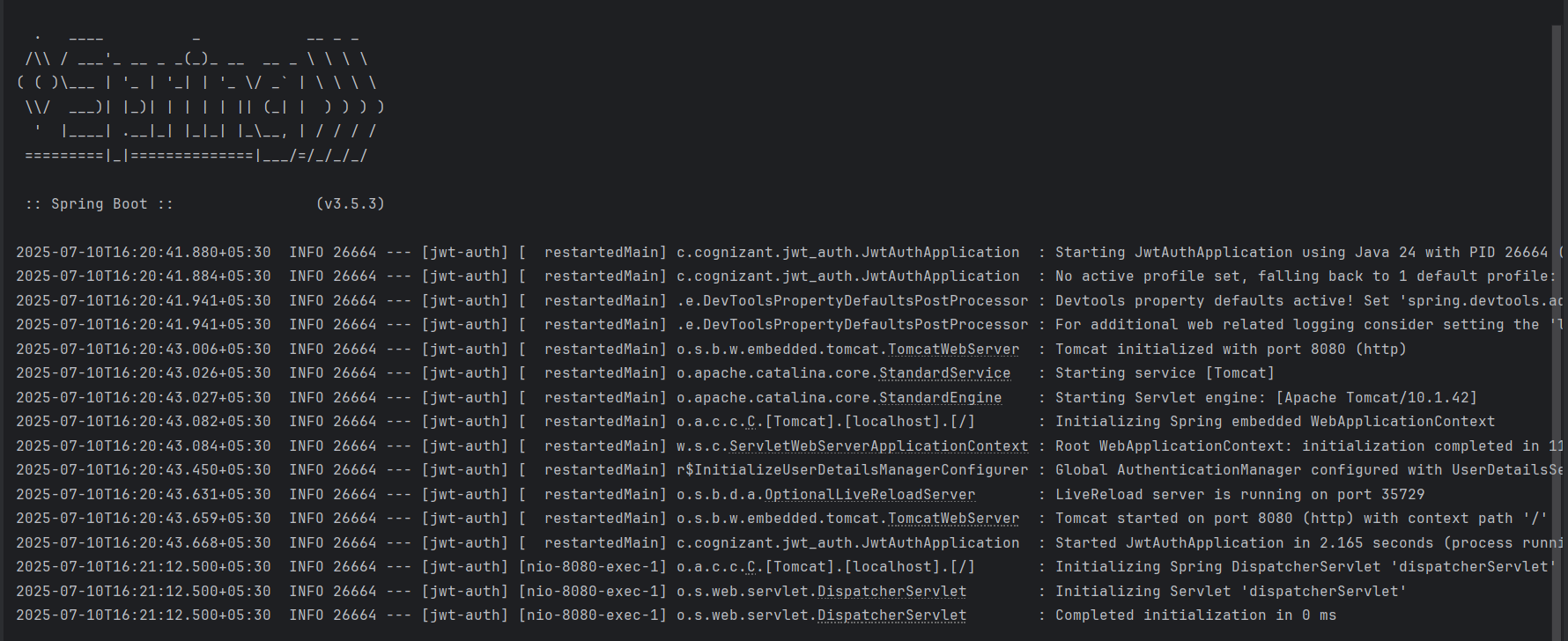
* Create the JwtAuthApplication.java in src/main/java/com/cognizant/jwt\_auth/JwtAuthApplication.java.

**JwtAuthApplication.java:**

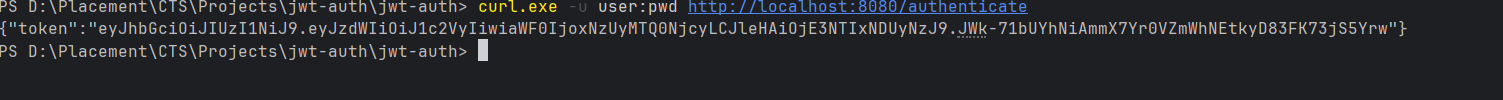
package com.cognizant.jwt\_auth;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class JwtAuthApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(JwtAuthApplication.class, args);  
 }  
}

* Run the JwtAuthApplication.java

**Output:**

****

**The terminal token generation:**

****

**Additional important hands-on**

# **File name: 3. spring-rest-handson**

# **7.Demonstrate integration of RESTful Web Service of type GET and test the service using postman.**

**Program:**

* Configure the application.properties.

**Application.properties:**

spring.application.name=jwt-auth  
server.port=8080  
spring.security.user.name=  
spring.security.user.password=  
spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration

* Update the pom.xml

**Pom.xml:**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.5.3</version>  
 <relativePath/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>jwt-auth</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>jwt-auth</name>  
 <description>JWT Auth Project</description>  
  
 <properties>  
 <java.version>24</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web and Security -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
  
 <!-- JSON Web Token - JJWT -->  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-api</artifactId>  
 <version>0.11.5</version>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-impl</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-jackson</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!-- Servlet API -->  
 <dependency>  
 <groupId>jakarta.servlet</groupId>  
 <artifactId>jakarta.servlet-api</artifactId>  
 <version>6.0.0</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!-- Development and Testing -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-tx</artifactId>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

* Create the XML file named employee.xml in src/main/resources/employee.xml.

**Employee.xml:**

package com.cognizant.springrest.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private double salary;  
 private Department department;  
  
 public Employee() {}  
  
 public Employee(int id, String name, double salary, Department department) {  
 this.id = id;  
 this.name = name;  
 this.salary = salary;  
 this.department = department;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
  
 public Department getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(Department department) {  
 this.department = department;  
 }  
}

* Create a class named Employee.java in src/main/java/com/cognizant/springrest/model

**Employee.java:**

package com.cognizant.springrest.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private double salary;  
 private Department department;  
  
 public Employee() {}  
  
 public Employee(int id, String name, double salary, Department department) {  
 this.id = id;  
 this.name = name;  
 this.salary = salary;  
 this.department = department;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
  
 public Department getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(Department department) {  
 this.department = department;  
 }  
}

* Create a class named Department.java in src/main/java/com/cognizant/springrest/model

**Department.java:**

package com.cognizant.springrest.model;  
  
public class Department {  
 private int id;  
 private String name;  
  
 public Department() {}  
  
 public Department(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

* Create a class named Skill.java in src/main/java/com/cognizant/springrest/model

**Skill.java:**

package com.cognizant.springrest.model;  
  
public class Skill {  
 private int id;  
 private String name;  
  
 public Skill() {}  
  
 public Skill(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

* Create a package named controller in src/main/java/com/cognizant/springrest/controller.
* Create a class named EmployeeController.java in src/main/java/com/cognizant/springrest/controller

**EmployeeController.java:**

package com.cognizant.springrest.controller;  
  
import com.cognizant.springrest.model.Employee;  
import com.cognizant.springrest.service.EmployeeService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 @GetMapping("/employees")  
 public List<Employee> getAllEmployees() {  
 return employeeService.getAllEmployees();  
 }  
}

* Create a class named DepartmentController.java in src/main/java/com/cognizant/springrest/controller

**DepartmentCountroller.java:**

package com.cognizant.springrest.controller;  
  
import com.cognizant.springrest.model.Department;  
import com.cognizant.springrest.service.DepartmentService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
@RestController  
public class DepartmentController {  
  
 @Autowired  
 private DepartmentService departmentService;  
  
 @GetMapping("/departments")  
 public List<Department> getAllDepartments() {  
 return departmentService.getAllDepartments();  
 }  
}

* Create a package named dao in src/main/java/com/cognizant/springrest/dao.
* Create a class named Employeedao.java in src/main/java/com/cognizant/springrest/dao.

**Employeedao.java:**

package com.cognizant.springrest.dao;  
  
import com.cognizant.springrest.model.Employee;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Repository;  
  
import java.util.Arrays;  
import java.util.List;  
  
@Repository  
public class EmployeeDao {  
  
 private static List<Employee> *EMPLOYEE\_LIST*;  
  
 public EmployeeDao() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");  
 Employee emp1 = context.getBean("employee1", Employee.class);  
 Employee emp2 = context.getBean("employee2", Employee.class);  
 *EMPLOYEE\_LIST* = Arrays.*asList*(emp1, emp2);  
 }  
  
 public List<Employee> getAllEmployees() {  
 return *EMPLOYEE\_LIST*;  
 }  
}

* Create a class named Departmentdao.java in src/main/java/com/cognizant/springrest/dao.

**Departmentdao.java:**

package com.cognizant.springrest.dao;  
  
import com.cognizant.springrest.model.Department;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Repository;  
  
import java.util.Arrays;  
import java.util.List;  
  
@Repository  
public class DepartmentDao {  
  
 private static List<Department> *DEPARTMENT\_LIST*;  
  
 public DepartmentDao() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");  
 Department dept1 = context.getBean("dept1", Department.class);  
 Department dept2 = context.getBean("dept2", Department.class);  
 *DEPARTMENT\_LIST* = Arrays.*asList*(dept1, dept2);  
 }  
  
 public List<Department> getAllDepartments() {  
 return *DEPARTMENT\_LIST*;  
 }  
}

* Create a package named service in src/main/java/com/cognizant/springrest/service.
* Create a class named Employeeservice.java in src/main/java/com/cognizant/springrest/service.

**Employeeservice.java:**

package com.cognizant.springrest.service;  
  
import com.cognizant.springrest.dao.EmployeeDao;  
import com.cognizant.springrest.model.Employee;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
import java.util.List;  
  
@Service  
public class EmployeeService {  
  
 @Autowired  
 private EmployeeDao employeeDao;  
  
 @Transactional  
 public List<Employee> getAllEmployees() {  
 return employeeDao.getAllEmployees();  
 }  
}

* Create a class named Departmentservice.java in src/main/java/com/cognizant/springrest/service.

**Departmentservice.java:**

package com.cognizant.springrest.service;  
  
import com.cognizant.springrest.dao.DepartmentDao;  
import com.cognizant.springrest.model.Department;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import java.util.List;  
  
@Service  
public class DepartmentService {  
  
 @Autowired  
 private DepartmentDao departmentDao;  
  
 public List<Department> getAllDepartments() {  
 return departmentDao.getAllDepartments();  
 }  
}

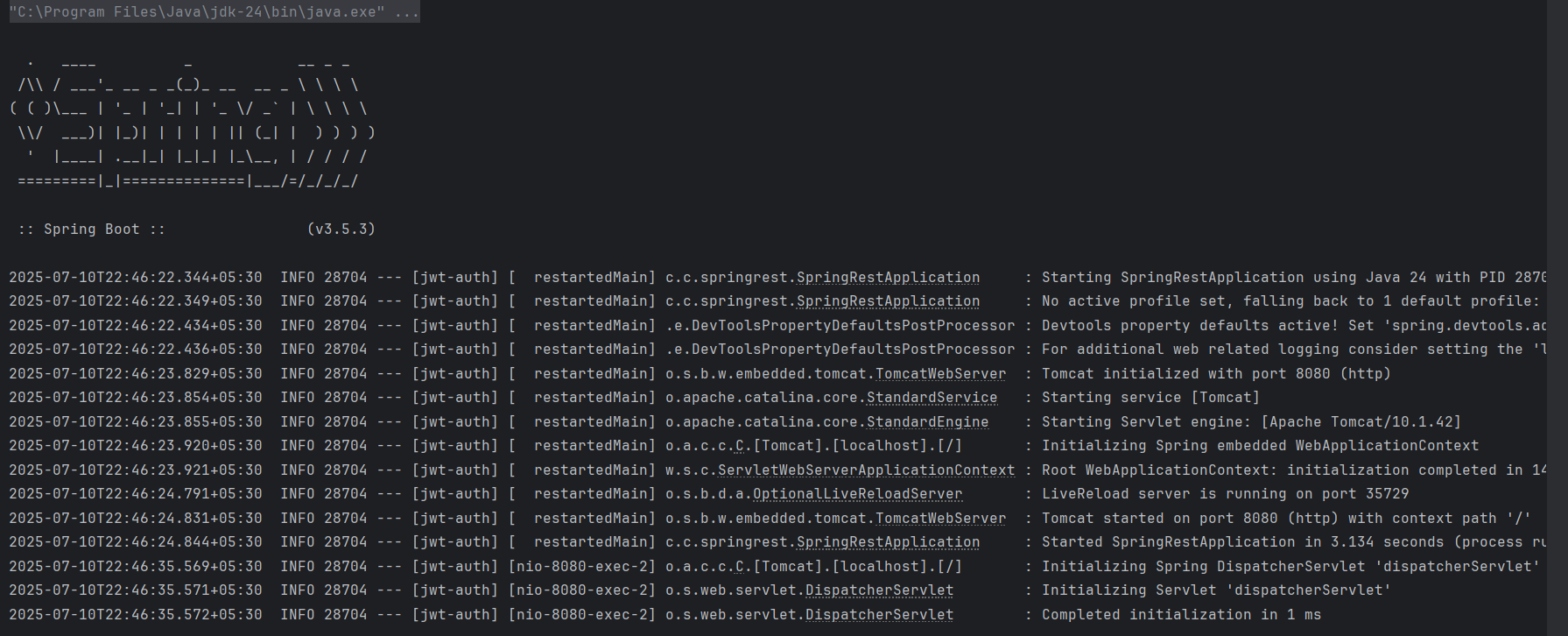
* Create the main class SpringRestApplication.java in src/main/java/com/cognizant/springrest.

**SpringRestApplication.java:**

package com.cognizant.springrest;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
@SpringBootApplication  
public class SpringRestApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringRestApplication.class, args);  
 }  
}

* Run the SpringRestApplication.java and Open the postman for checking
* GET all employees: <http://localhost:8080/employees>
* GET all departments: <http://localhost:8080/departments>

**Output:**

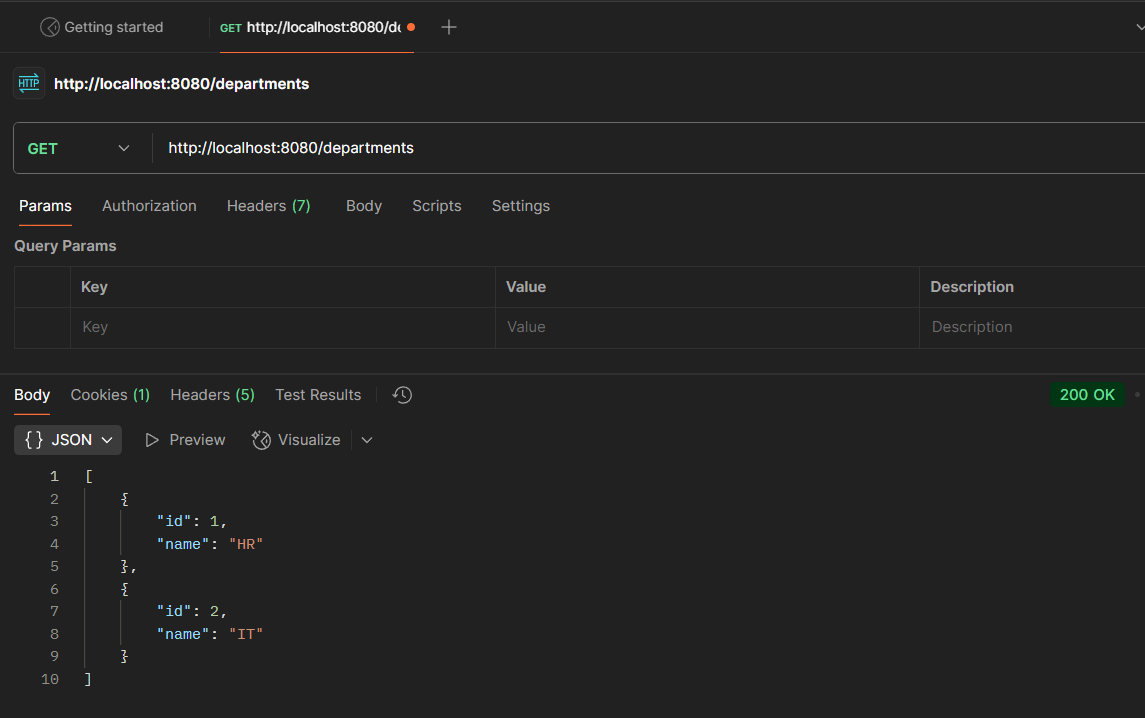
****

**In postman -to get all employees:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**To get all departments:**

****