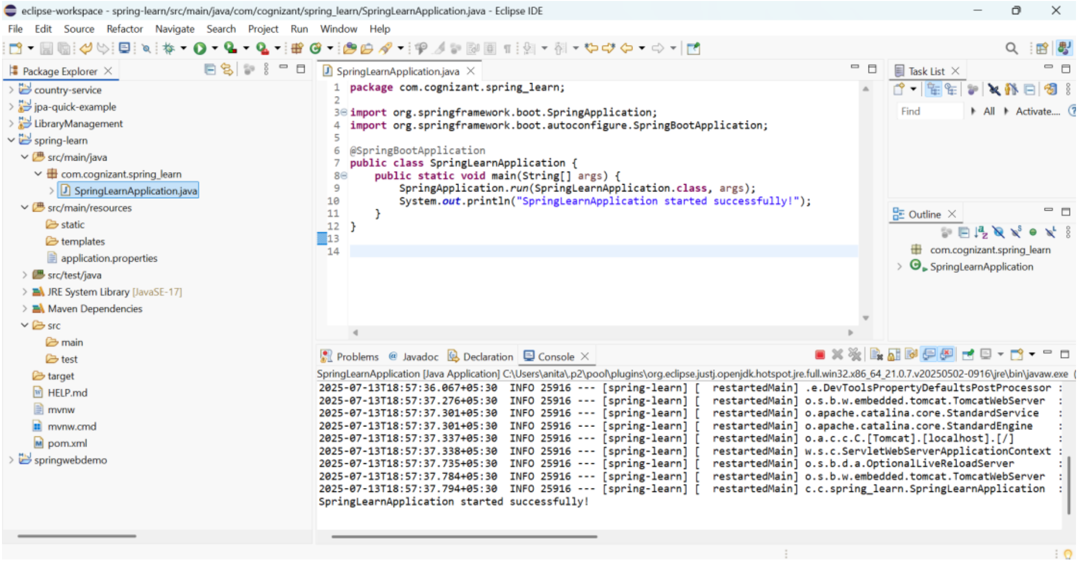
WEEK – 4

NAME : ANITA.P

SUPERSET ID : 6419735

Mandatory Hands-on

1) Create a Spring Web Project using Maven

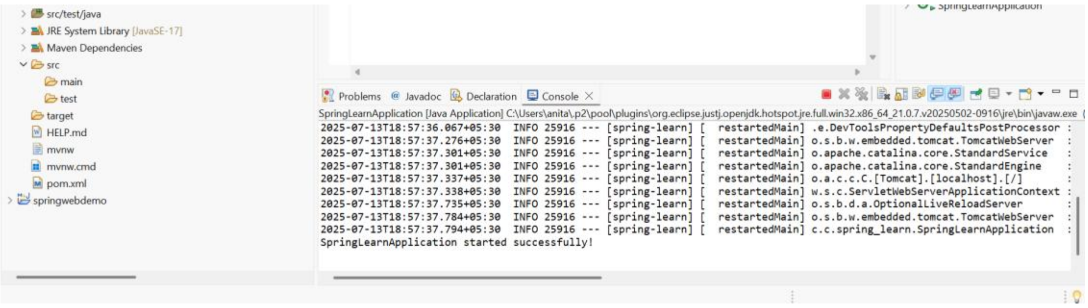


eclipse-workspace - spring-learn/src/main/java/com/cognizant/spring_learn/SpringLearnApplication.java - Eclipse IDE

```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) {
        SpringApplication.run(SpringLearnApplication.class, args);
        System.out.println("SpringLearnApplication started successfully!");
    }
}
```

SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
2025-07-13T18:57:36.067+05:30  INFO 25916 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor :
2025-07-13T18:57:37.276+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T18:57:37.301+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   :
2025-07-13T18:57:37.301+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    :
2025-07-13T18:57:37.337+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       :
2025-07-13T18:57:37.338+05:30  INFO 25916 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext :
2025-07-13T18:57:37.735+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       :
2025-07-13T18:57:37.784+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T18:57:37.794+05:30  INFO 25916 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  :
SpringLearnApplication started successfully!
```



SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
2025-07-13T18:57:36.067+05:30  INFO 25916 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor :
2025-07-13T18:57:37.276+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T18:57:37.301+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   :
2025-07-13T18:57:37.301+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    :
2025-07-13T18:57:37.337+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       :
2025-07-13T18:57:37.338+05:30  INFO 25916 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext :
2025-07-13T18:57:37.735+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       :
2025-07-13T18:57:37.784+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T18:57:37.794+05:30  INFO 25916 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  :
SpringLearnApplication started successfully!
```

2) Spring Core – Load Country from Spring Configuration XML

data-format.xml

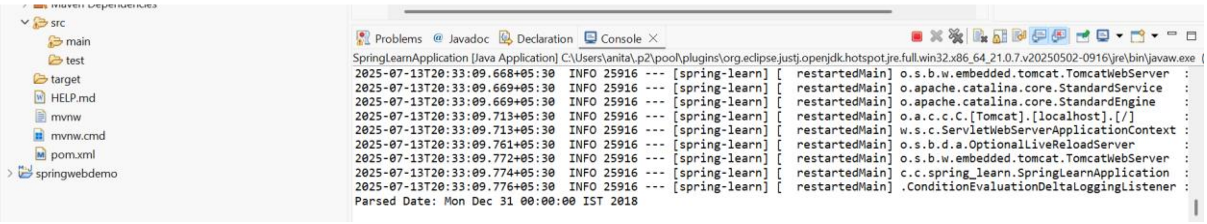
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
      https://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="dateFormat" class="java.text.SimpleDateFormat">
```

```
        <constructor-arg value="dd/MM/yyyy" />
    </bean>
</beans>
```

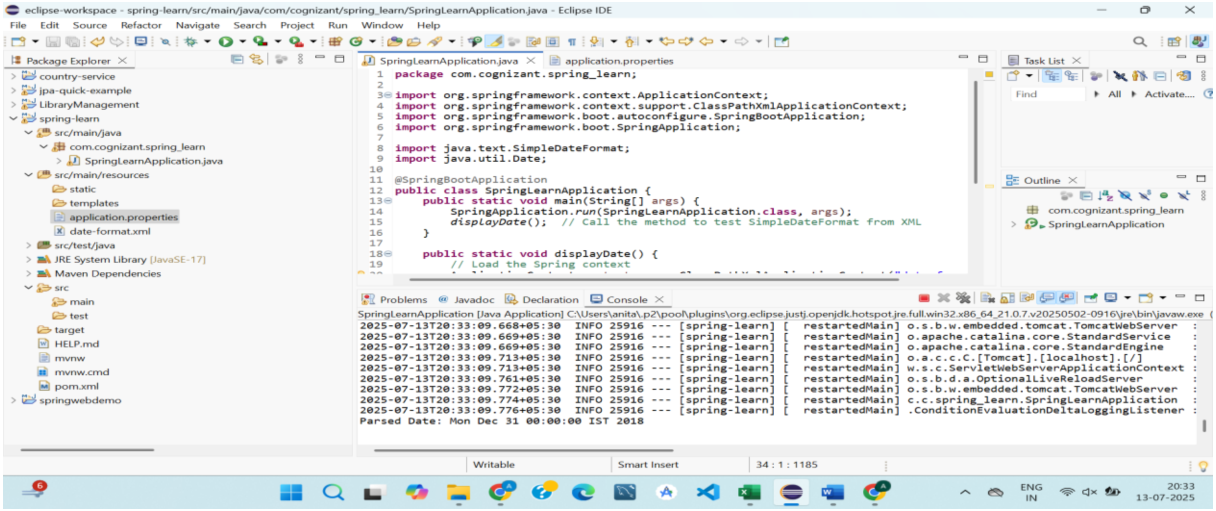
SpringLearnApplication.java

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

OUTPUT:



```
SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe
2025-07-13T20:33:09.668+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T20:33:09.669+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   :
2025-07-13T20:33:09.669+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    :
2025-07-13T20:33:09.713+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       :
2025-07-13T20:33:09.713+05:30  INFO 25916 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext :
2025-07-13T20:33:09.761+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       :
2025-07-13T20:33:09.772+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T20:33:09.774+05:30  INFO 25916 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  :
2025-07-13T20:33:09.776+05:30  INFO 25916 --- [spring-learn] [  restartedMain] .ConditionEvaluationDeltaLoggingListener :
Parsed Date: Mon Dec 31 00:00:00 IST 2018
```



```java
package com.cognizant.spring_learn;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.SpringApplication;

import java.text.SimpleDateFormat;
import java.util.Date;

@SpringBootApplication
public class SpringLearnApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
        displayDate();  // Call the method to test SimpleDateFormat from XML
    }

    public static void displayDate() {
        // Load the Spring context
```

SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
2025-07-13T20:33:09.668+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T20:33:09.669+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   :
2025-07-13T20:33:09.669+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    :
2025-07-13T20:33:09.713+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       :
2025-07-13T20:33:09.713+05:30  INFO 25916 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext :
2025-07-13T20:33:09.761+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       :
2025-07-13T20:33:09.772+05:30  INFO 25916 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T20:33:09.774+05:30  INFO 25916 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  :
2025-07-13T20:33:09.776+05:30  INFO 25916 --- [spring-learn] [  restartedMain] .ConditionEvaluationDeltaLoggingListener :
Parsed Date: Mon Dec 31 00:00:00 IST 2018
```

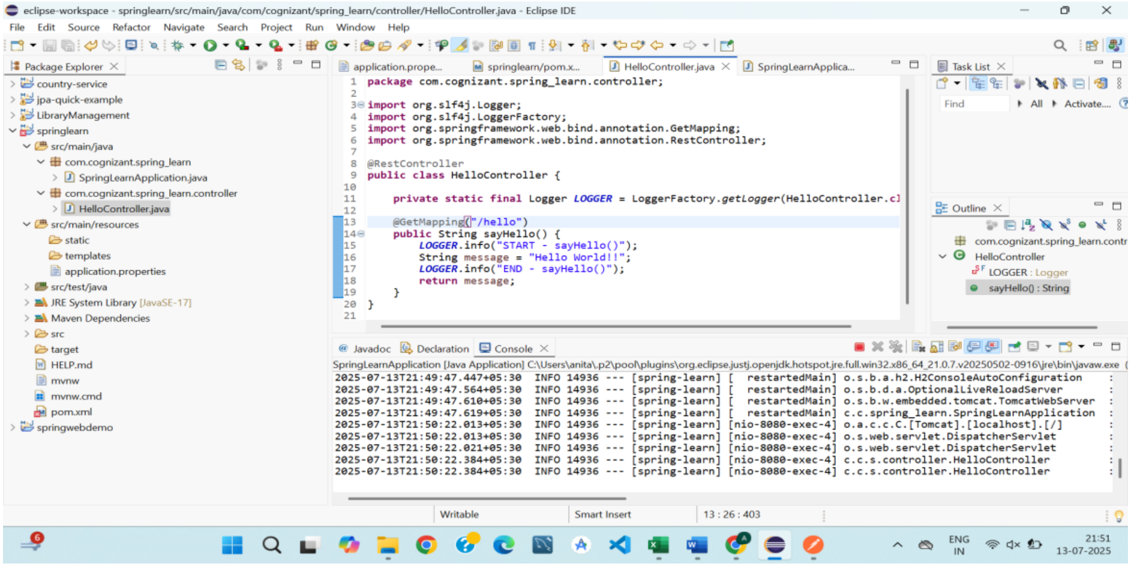
3) Hello World RESTful Web Service

HelloController.java

```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;
   }}
```

SpringLearnApplication.java

```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) {
      SpringApplication.run(SpringLearnApplication.class, args);
   }
}
```

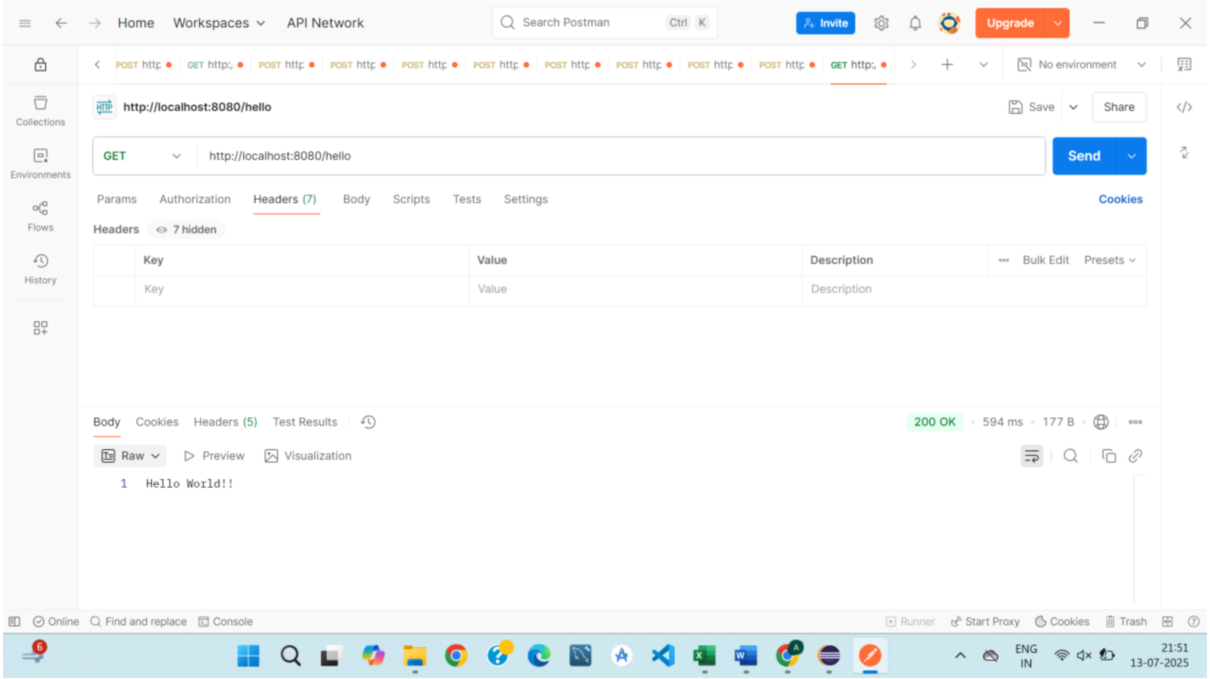


eclipse-workspace - springlearn/src/main/java/com/cognizant/spring_learn/controller/HelloController.java - Eclipse IDE

```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.cl

    @GetMapping("/hello")
    public String sayHello() {
        LOGGER.info("START - sayHello()");
        String message = "Hello World!!";
        LOGGER.info("END - sayHello()");
        return message;
    }
}
```

SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
2025-07-13T21:49:47.447+05:30  INFO 14936 --- [spring-learn] [  restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration   :
2025-07-13T21:49:47.564+05:30  INFO 14936 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer      :
2025-07-13T21:49:47.610+05:30  INFO 14936 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer :
2025-07-13T21:49:47.619+05:30  INFO 14936 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication :
2025-07-13T21:50:22.013+05:30  INFO 14936 --- [spring-learn] [nio-8080-exec-4] o.a.c.c.C.[Tomcat].[localhost].[/]      :
2025-07-13T21:50:22.013+05:30  INFO 14936 --- [spring-learn] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet       :
2025-07-13T21:50:22.021+05:30  INFO 14936 --- [spring-learn] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet       :
2025-07-13T21:50:22.384+05:30  INFO 14936 --- [spring-learn] [nio-8080-exec-4] c.c.s.controller.HelloController        :
2025-07-13T21:50:22.384+05:30  INFO 14936 --- [spring-learn] [nio-8080-exec-4] c.c.s.controller.HelloController        :
```



GET http://localhost:8080/hello

Body Cookies Headers (5) Test Results — 200 OK • 594 ms • 177 B

```
Hello World!!
```

4) REST - Country Web Service

<u>CountryController.java</u>

package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.model.Country;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController

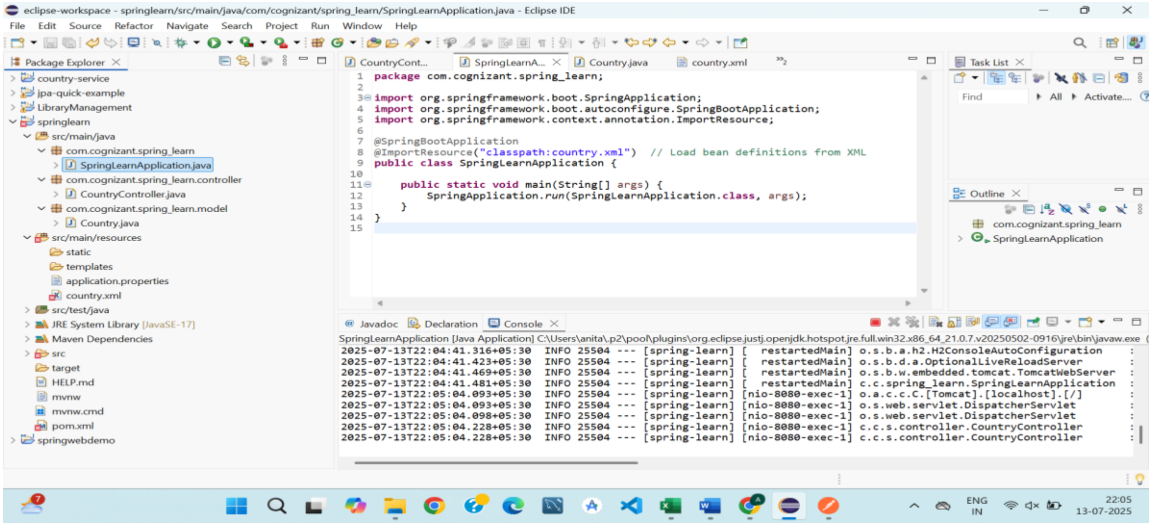
```
public class CountryController {
   private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
   @Autowired
   private ApplicationContext context;
   @RequestMapping("/country")
   public Country getCountryIndia() {
      LOGGER.info("START - getCountryIndia()");
      Country country = (Country) context.getBean("country");
      LOGGER.info("END - getCountryIndia()");
      return country;
   }
}
```

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;
   }}
```

SpringLearnApplication.java

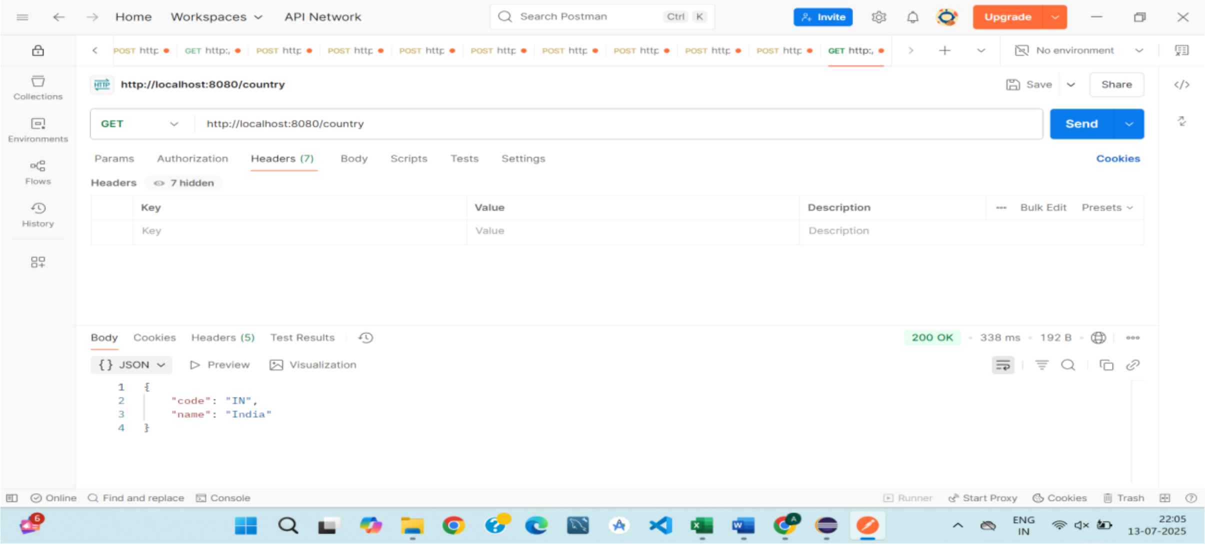
```
package com.cognizant.spring_learn;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;
@SpringBootApplication
@ImportResource("classpath:country.xml")  // Load bean definitions from XML
public class SpringLearnApplication {

   public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
    }}
```



eclipse-workspace - springlearn/src/main/java/com/cognizant/spring_learn/SpringLearnApplication.java - Eclipse IDE

```java
package com.cognizant.spring_learn;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;

@SpringBootApplication
@ImportResource("classpath:country.xml")  // Load bean definitions from XML
public class SpringLearnApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```

```
SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe
2025-07-13T22:04:41.316+05:30  INFO 25504 --- [spring-learn] [  restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration  :
2025-07-13T22:04:41.423+05:30  INFO 25504 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer     :
2025-07-13T22:04:41.469+05:30  INFO 25504 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer :
2025-07-13T22:04:41.481+05:30  INFO 25504 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication :
2025-07-13T22:05:04.093+05:30  INFO 25504 --- [spring-learn] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]     :
2025-07-13T22:05:04.093+05:30  INFO 25504 --- [spring-learn] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet      :
2025-07-13T22:05:04.098+05:30  INFO 25504 --- [spring-learn] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet      :
2025-07-13T22:05:04.228+05:30  INFO 25504 --- [spring-learn] [nio-8080-exec-1] c.c.s.controller.CountryController     :
2025-07-13T22:05:04.228+05:30  INFO 25504 --- [spring-learn] [nio-8080-exec-1] c.c.s.controller.CountryController     :
```



GET http://localhost:8080/country

Headers (7) — 7 hidden

Key	Value	Description
Key	Value	Description

Body — 200 OK · 338 ms · 192 B

```
{
    "code": "IN",
    "name": "India"
}
```

5) REST - Get country based on country code

CountryController.java

```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("/country/{code}")
  public Country getCountry(@PathVariable String code) {
    LOGGER.info("START - getCountry() with code: {}", code);
    Country country = countryService.getCountry(code);
    LOGGER.info("END - getCountry()");
    return country;
  }}
```

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;
  }}
```

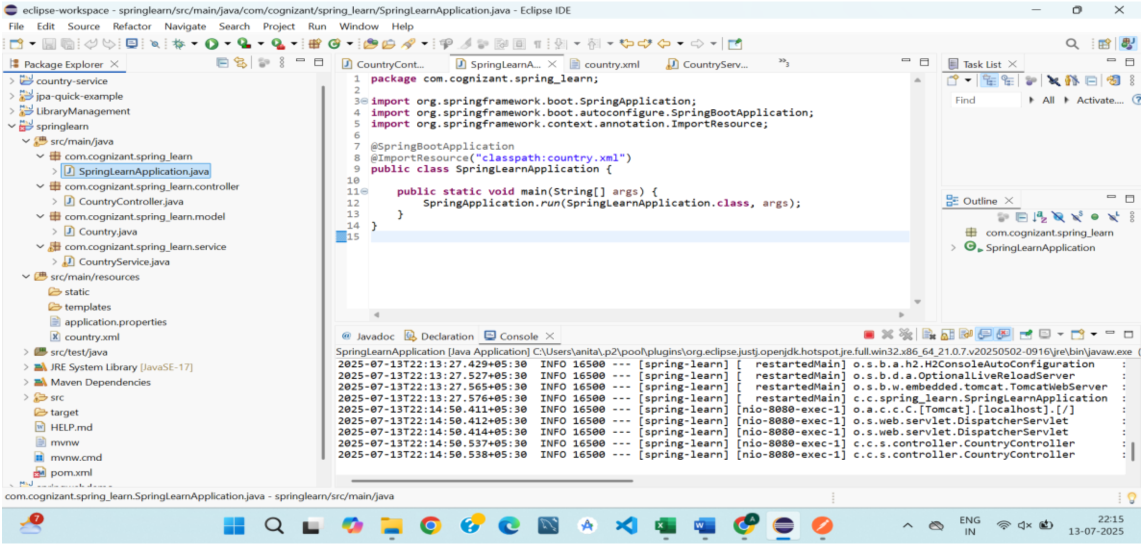
CountryService.java

```
package com.cognizant.spring_learn.service;

import com.cognizant.spring_learn.model.Country;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.stereotype.Service;
import java.util.List;
@Service
public class CountryService {
  @Autowired
  private ApplicationContext context;
  public Country getCountry(String code) {
    List<Country> countries = (List<Country>) context.getBean("countryList");
    return countries.stream()
        .filter(country -> country.getCode().equalsIgnoreCase(code))
```

```
            .findFirst()
            .orElse(null); // or throw a custom CountryNotFoundException
    }}
```

SpringLearnApplication.java

```
package com.cognizant.spring_learn;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;
@SpringBootApplication
@ImportResource("classpath:country.xml")
public class SpringLearnApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```

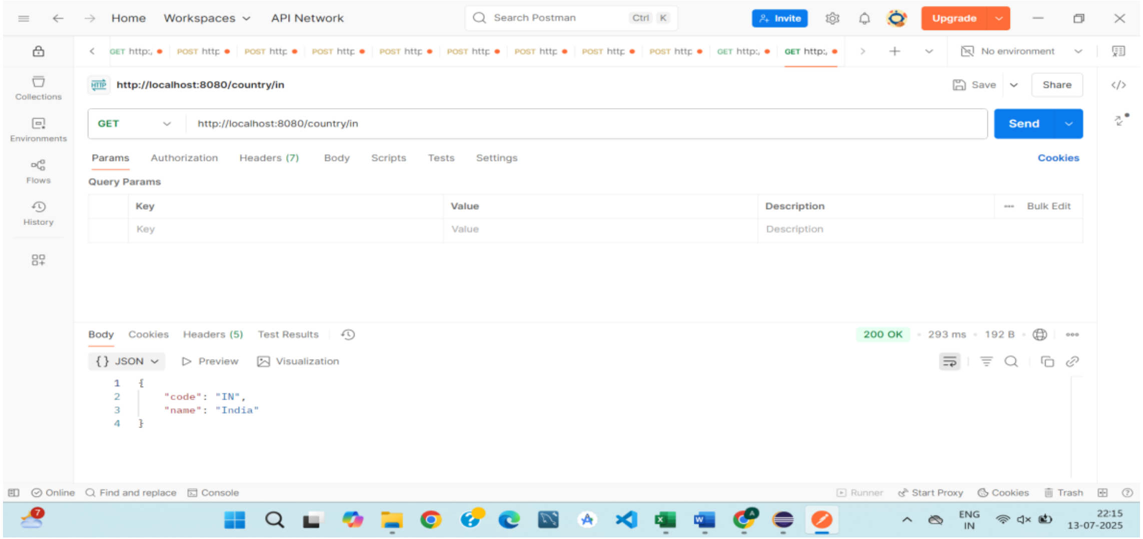


eclipse-workspace - springlearn/src/main/java/com/cognizant/spring_learn/SpringLearnApplication.java - Eclipse IDE

```
package com.cognizant.spring_learn;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;

@SpringBootApplication
@ImportResource("classpath:country.xml")
public class SpringLearnApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```

SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
2025-07-13T22:13:27.429+05:30  INFO 16500 --- [spring-learn] [  restartedMain] o.s.b.a.h2.H2ConsoleAutoConfiguration
2025-07-13T22:13:27.527+05:30  INFO 16500 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer
2025-07-13T22:13:27.565+05:30  INFO 16500 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-13T22:13:27.576+05:30  INFO 16500 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication
2025-07-13T22:14:50.411+05:30  INFO 16500 --- [spring-learn] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-13T22:14:50.412+05:30  INFO 16500 --- [spring-learn] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
2025-07-13T22:14:50.414+05:30  INFO 16500 --- [spring-learn] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
2025-07-13T22:14:50.537+05:30  INFO 16500 --- [spring-learn] [nio-8080-exec-1] c.c.s.controller.CountryController
2025-07-13T22:14:50.538+05:30  INFO 16500 --- [spring-learn] [nio-8080-exec-1] c.c.s.controller.CountryController
```



GET http://localhost:8080/country/in

200 OK · 293 ms · 192 B

```
{
    "code": "IN",
    "name": "India"
}
```

6) Create authentication service that returns JWT

SpringLearnApplication.java

```
package com.cognizant.spring_learn.config.SecurityConfig;
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.*;
import org.springframework.security.core.userdetails.*;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;
@Configuration
@EnableWebSecurity
public class SecurityConfig {
  @Bean
  public InMemoryUserDetailsManager userDetailsService() {
    UserDetails user = User.withDefaultPasswordEncoder()
        .username("user")
        .password("pwd")
        .roles("USER")
        .build();
    return new InMemoryUserDetailsManager(user);
  }
  @Bean
  public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    http.csrf().disable()
      .authorizeHttpRequests()
      .requestMatchers("/authenticate").authenticated()
      .anyRequest().permitAll()
      .and()
      .httpBasic(); // enable basic authentication
    return http.build();
  }
}
```

SecurityConfig.java

```
package com.cognizant.spring_learn.config.SecurityConfig;

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.*;
import org.springframework.security.core.userdetails.*;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;
@Configuration
@EnableWebSecurity
public class SecurityConfig {
  @Bean
  public InMemoryUserDetailsManager userDetailsService() {
    UserDetails user = User.withDefaultPasswordEncoder()
        .username("user")
        .password("pwd")
        .roles("USER")
        .build();
    return new InMemoryUserDetailsManager(user);
  }
  @Bean
  public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    http.csrf().disable()
      .authorizeHttpRequests()
      .requestMatchers("/authenticate").authenticated()
      .anyRequest().permitAll()
      .and()
      .httpBasic(); // enable basic authentication
    return http.build();  }}
```

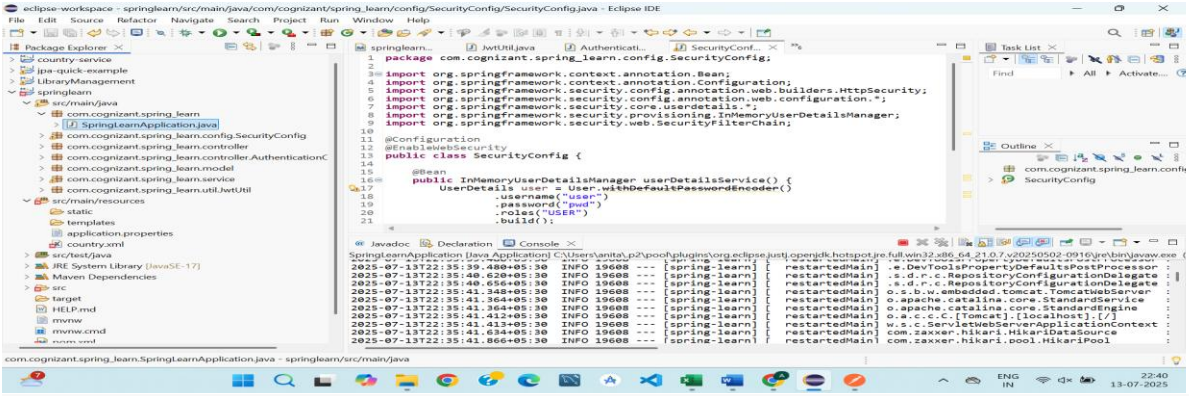
AuthenticationController.java

```
package com.cognizant.spring_learn.controller.AuthenticationController;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.*;
import com.cognizant.spring_learn.util.JwtUtil.JwtUtil;
import java.util.HashMap;
import java.util.Map;
@RestController
public class AuthenticationController {
  @Autowired
  private JwtUtil jwtUtil;
  @GetMapping("/authenticate")
  public Map<String, String> authenticate(Authentication authentication) {
    String username = authentication.getName();
    String token = jwtUtil.generateToken(username);
    Map<String, String> response = new HashMap<>();
    response.put("token", token);
    return response;
  }
}
```

JwtUtil.java

```
package com.cognizant.spring_learn.util.JwtUtil;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import org.springframework.stereotype.Component;
import java.util.Date;
@Component
public class JwtUtil {
  private final String SECRET_KEY = "secret";
  public String generateToken(String username) {
```

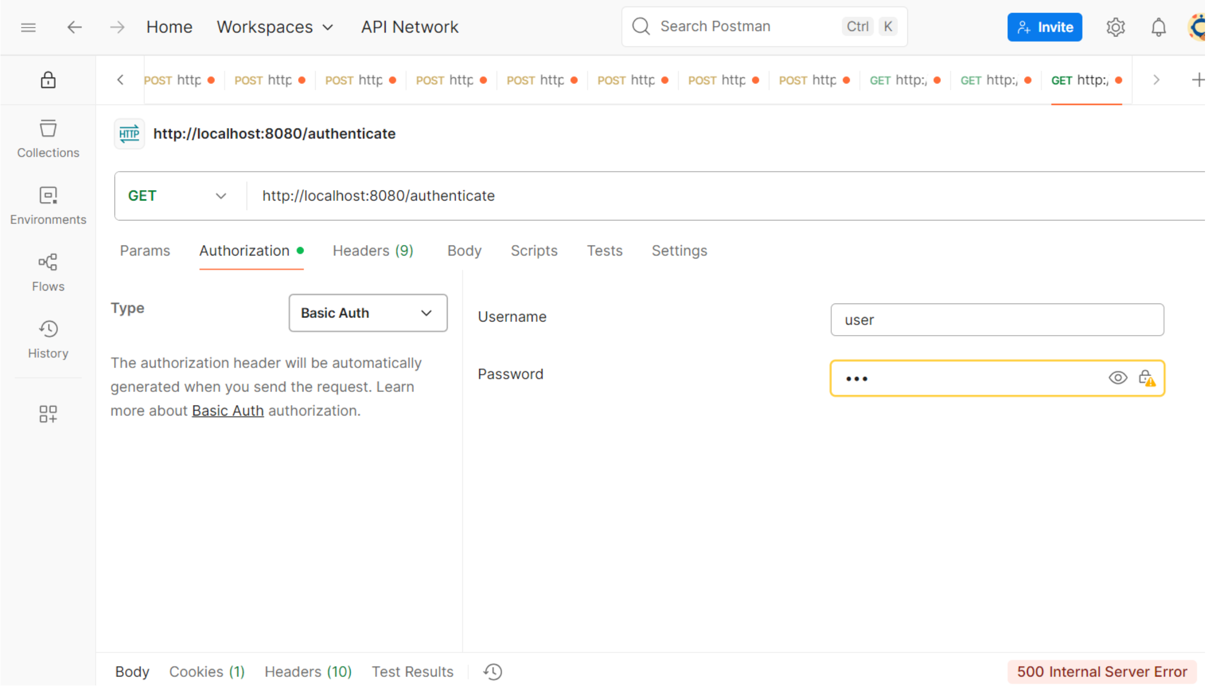
```
return Jwts.builder()
        .setSubject(username)
        .setIssuedAt(new Date(System.currentTimeMillis()))
        .setExpiration(new Date(System.currentTimeMillis() + 1000 * 60 * 60)) // 1 hour
        .signWith(SignatureAlgorithm.HS256, SECRET_KEY)
        .compact();  }}
```



```
package com.cognizant.spring_learn.config.SecurityConfig;

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.*;
import org.springframework.security.core.userdetails.*;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public InMemoryUserDetailsManager userDetailsService() {
        UserDetails user = User.withDefaultPasswordEncoder()
                .username("user")
                .password("pwd")
                .roles("USER")
                .build();
```

```
SpringLearnApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe
2025-07-13T22:35:39.480+05:30  INFO 19608 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor :
2025-07-13T22:35:40.620+05:30  INFO 19608 --- [spring-learn] [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate :
2025-07-13T22:35:40.656+05:30  INFO 19608 --- [spring-learn] [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate :
2025-07-13T22:35:41.348+05:30  INFO 19608 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  :
2025-07-13T22:35:41.364+05:30  INFO 19608 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   :
2025-07-13T22:35:41.364+05:30  INFO 19608 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    :
2025-07-13T22:35:41.412+05:30  INFO 19608 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       :
2025-07-13T22:35:41.413+05:30  INFO 19608 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext :
2025-07-13T22:35:41.634+05:30  INFO 19608 --- [spring-learn] [  restartedMain] com.zaxxer.hikari.HikariDataSource       :
2025-07-13T22:35:41.866+05:30  INFO 19608 --- [spring-learn] [  restartedMain] com.zaxxer.hikari.pool.HikariPool        :
```

eclipse-workspace - springlearn/src/main/java/com/cognizant/spring_learn/config/SecurityConfig/SecurityConfig.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
country-service
jpa-quick-example
LibraryManagement
springlearn
src/main/java
com.cognizant.spring_learn
SpringLearnApplication.java
com.cognizant.spring_learn.config.SecurityConfig
com.cognizant.spring_learn.controller
com.cognizant.spring_learn.controller.AuthenticationC
com.cognizant.spring_learn.model
com.cognizant.spring_learn.service
com.cognizant.spring_learn.util.JwtUtil
src/main/resources
static
templates
application.properties
country.xml
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
springlearn...
JwtUtil.java
Authenticati...
SecurityConf...
Task List
Outline
com.cognizant.spring_learn.config
SecurityConfig
Javadoc
Declaration
Console
com.cognizant.spring_learn.SpringLearnApplication.java - springlearn/src/main/java
ENG IN
22:40
13-07-2025



GET http://localhost:8080/authenticate

Authorization: Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about Basic Auth authorization.

Username: user

Password: •••

500 Internal Server Error

Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Collections
Environments
Flows
History
POST http
GET http:
http://localhost:8080/authenticate
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Body
Cookies (1)
Headers (10)
Test Results

ADDITIONAL EXERCISES

1) Display Employee List and Edit Employee form using RESTful Web Service

Employee.java

```
package com.cognizant.employee_rest.model;
public class Employee {
    private int id;
    private String name;
    private double salary;
    private String department;
    public Employee() {}
    public Employee(int id, String name, double salary, String 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 String getDepartment() { return department; }
    public void setDepartment(String department) { this.department = department; }
}
```

EmployeeController.java

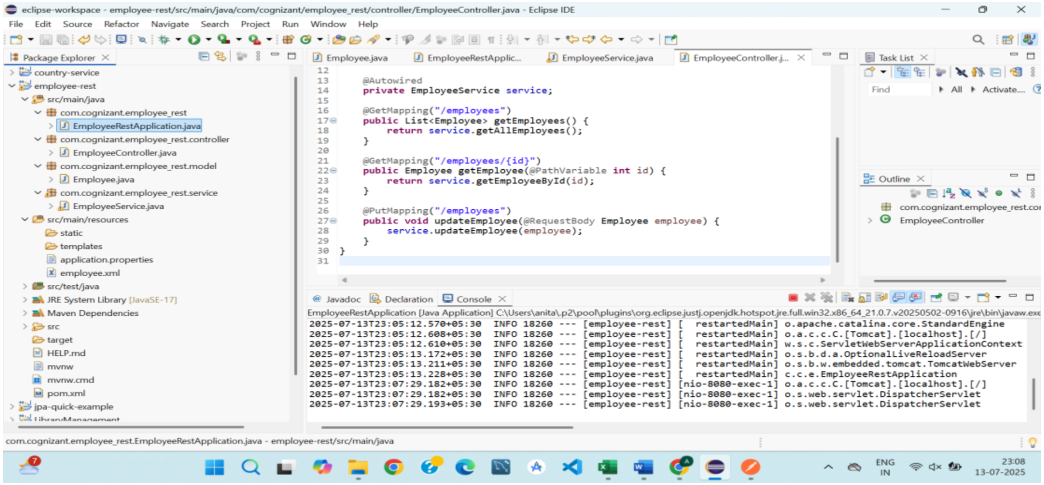
```
package com.cognizant.employee_rest.controller;
import com.cognizant.employee_rest.model.Employee;
import com.cognizant.employee_rest.service.EmployeeService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;
@RestController
public class EmployeeController {
    @Autowired
    private EmployeeService service;
    @GetMapping("/employees")
    public List<Employee> getEmployees() {
        return service.getAllEmployees();
    }
    @GetMapping("/employees/{id}")
    public Employee getEmployee(@PathVariable int id) {
        return service.getEmployeeById(id);
    }
    @PutMapping("/employees")
    public void updateEmployee(@RequestBody Employee employee) {
        service.updateEmployee(employee);
    }
}
```

EmployeeService.java

```
package com.cognizant.employee_rest.service;
import com.cognizant.employee_rest.model.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
import org.springframework.stereotype.Service;
import java.util.List;
@Service
public class EmployeeService {
  @Autowired
  private ApplicationContext context;
  public List<Employee> getAllEmployees() {
    return (List<Employee>) context.getBean("employeeList");
  }
  public Employee getEmployeeById(int id) {
    List<Employee> employees = getAllEmployees();
    return employees.stream()
          .filter(e -> e.getId() == id)
          .findFirst()
          .orElse(null);
  }
  public void updateEmployee(Employee updatedEmployee) {
    List<Employee> employees = getAllEmployees();
    for (Employee e : employees) {
      if (e.getId() == updatedEmployee.getId()) {
        e.setName(updatedEmployee.getName());
        e.setSalary(updatedEmployee.getSalary());
        e.setDepartment(updatedEmployee.getDepartment());
        break;
      }}}}
```

EmployeeRestApplication.java

```
package com.cognizant.employee_rest;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;
@SpringBootApplication
@ImportResource("classpath:employee.xml")
public class EmployeeRestApplication {
  public static void main(String[] args) {
    SpringApplication.run(EmployeeRestApplication.class, args);
  }}
```

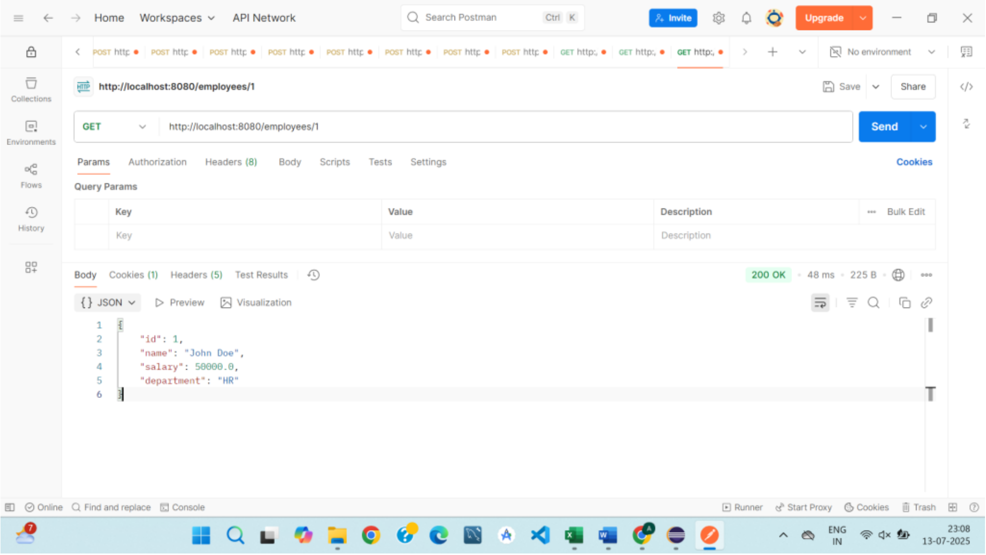


eclipse-workspace - employee-rest/src/main/java/com/cognizant/employee_rest/controller/EmployeeController.java - Eclipse IDE

```
    @Autowired
    private EmployeeService service;

    @GetMapping("/employees")
    public List<Employee> getEmployees() {
        return service.getAllEmployees();
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployee(@PathVariable int id) {
        return service.getEmployeeById(id);
    }

    @PutMapping("/employees")
    public void updateEmployee(@RequestBody Employee employee) {
        service.updateEmployee(employee);
    }
}
```

```
EmployeeRestApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe
2025-07-13T23:05:12.570+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.apache.catalina.core.StandardEngine
2025-07-13T23:05:12.608+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-13T23:05:12.610+05:30  INFO 18260 --- [employee-rest] [  restartedMain] w.s.c.ServletWebServerApplicationContext
2025-07-13T23:05:13.172+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer
2025-07-13T23:05:13.211+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-13T23:05:13.228+05:30  INFO 18260 --- [employee-rest] [  restartedMain] c.c.e.EmployeeRestApplication
2025-07-13T23:07:29.182+05:30  INFO 18260 --- [employee-rest] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-13T23:07:29.182+05:30  INFO 18260 --- [employee-rest] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
2025-07-13T23:07:29.193+05:30  INFO 18260 --- [employee-rest] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
```

com.cognizant.employee_rest.EmployeeRestApplication.java - employee-rest/src/main/java



http://localhost:8080/employees/1

GET http://localhost:8080/employees/1

200 OK · 48 ms · 225 B

```
{
    "id": 1,
    "name": "John Doe",
    "salary": 50000.0,
    "department": "HR"
}
```

2) Create static employee list data using spring xml configuration

Employee.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
         http://www.springframework.org/schema/beans/spring-beans.xsd">
    <!-- Departments -->
    <bean id="department1" class="com.cognizant.spring_learn.model.Department">
        <property name="id" value="1" />
        <property name="name" value="HR" />
    </bean>
    <bean id="department2" class="com.cognizant.spring_learn.model.Department">
        <property name="id" value="2" />
        <property name="name" value="IT" />
    </bean>
    <!-- Skills -->
    <bean id="skill1" class="com.cognizant.spring_learn.model.Skill">
        <property name="id" value="1"/>
        <property name="name" value="Java"/>
    </bean>
    <bean id="skill2" class="com.cognizant.spring_learn.model.Skill">
        <property name="id" value="2"/>
        <property name="name" value="Spring"/>
    </bean>
    <bean id="skillList1" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <ref bean="skill1"/>
                <ref bean="skill2"/>
            </list>
    </constructor-arg>
  </bean>
  <!-- Employees -->
  <bean id="employee1" class="com.cognizant.spring_learn.model.Employee">
    <property name="id" value="1"/>
    <property name="name" value="John Doe"/>
    <property name="salary" value="50000"/>
    <property name="permanent" value="true"/>
    <property name="department" ref="department1"/>
    <property name="skillList" ref="skillList1"/>
  </bean>
  <bean id="employee2" class="com.cognizant.spring_learn.model.Employee">
    <property name="id" value="2"/>
    <property name="name" value="Jane Smith"/>
    <property name="salary" value="60000"/>
    <property name="permanent" value="false"/>
    <property name="department" ref="department2"/>
    <property name="skillList" ref="skillList1"/>
  </bean>
  <bean id="employee3" class="com.cognizant.spring_learn.model.Employee">
    <property name="id" value="3"/>
    <property name="name" value="Alice"/>
    <property name="salary" value="55000"/>
    <property name="permanent" value="true"/>
    <property name="department" ref="department1"/>
    <property name="skillList" ref="skillList1"/>
  </bean>
  <bean id="employee4" class="com.cognizant.spring_learn.model.Employee">
    <property name="id" value="4"/>
    <property name="name" value="Bob"/>
    <property name="salary" value="45000"/>
    <property name="permanent" value="false"/>
    <property name="department" ref="department2"/>
    <property name="skillList" ref="skillList1"/>
  </bean>
  <!-- Employee List -->
  <bean id="employeeList" class="java.util.ArrayList">
    <constructor-arg>
      <list>
        <ref bean="employee1"/>
        <ref bean="employee2"/>
        <ref bean="employee3"/>
        <ref bean="employee4"/>
      </list>
    </constructor-arg>
  </bean>
</beans>
```

Department.java

```
package com.cognizant.spring_learn.model;
```

```
public class Department {
   private int id;
   private String name;
   public 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;
   }
}
```

Employee.java

```
package com.cognizant.spring_learn.model;
import java.util.List;
public class Employee {
   private int id;
   private String name;
   private double salary;
   private boolean permanent;
   private Department department;
   private List<Skill> skillList;
   public Employee() {}
  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 boolean isPermanent() {
      return permanent;
   }
  public void setPermanent(boolean permanent) {
    this.permanent = permanent;
  }
  public Department getDepartment() {
    return department;
  }
  public void setDepartment(Department department) {
    this.department = department;
  }
  public List<Skill> getSkillList() {
    return skillList;
  }
  public void setSkillList(List<Skill> skillList) {
    this.skillList = skillList;
  }
}
```

Skill.java

```
package com.cognizant.spring_learn.model;
public class Skill {
  private int id;
  private String name;
  public Skill() {}
   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;
  }
}
```

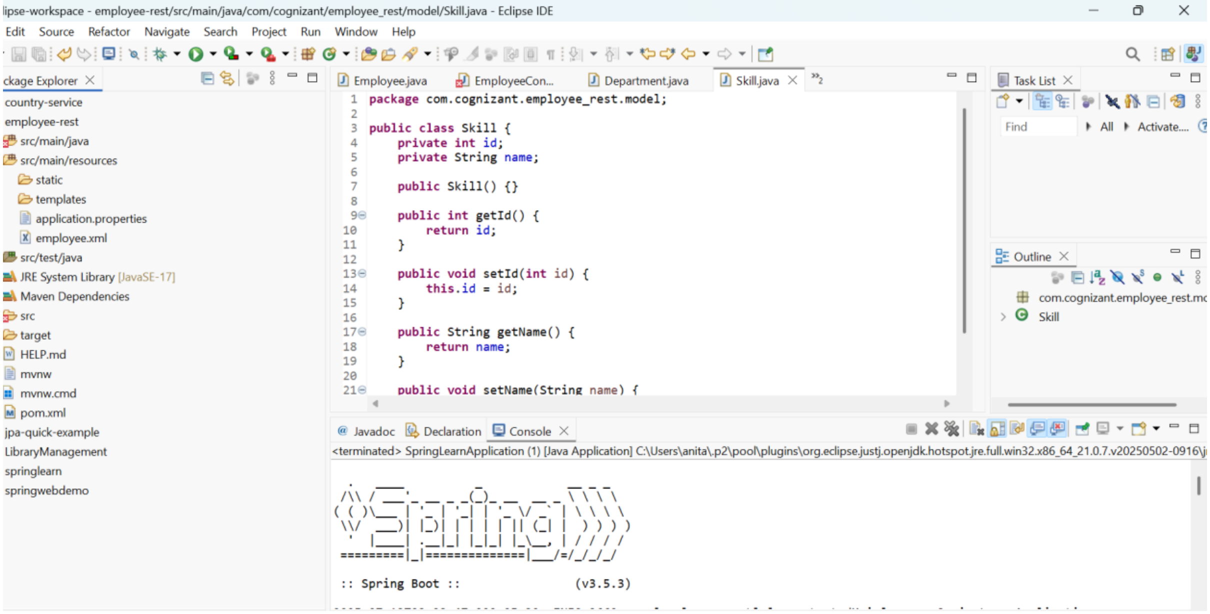
EmployeeDao.java

```
package com.cognizant.spring_learn.dao;

import com.cognizant.spring_learn.model.Employee;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;
import java.util.ArrayList;
@Repository
public class EmployeeDao {
  public static ArrayList<Employee> EMPLOYEE_LIST;
```

```
    public EmployeeDao() {
        ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");
        EMPLOYEE_LIST = (ArrayList<Employee>) context.getBean("employeeList");
    }
    public ArrayList<Employee> getAllEmployees() {
        return EMPLOYEE_LIST;
    }
}
```

SpringLearnApplication.java

```
package com.cognizant.spring_learn;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;
@SpringBootApplication
@ImportResource("classpath:employee.xml")
public class SpringLearnApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```



lipse-workspace - employee-rest/src/main/java/com/cognizant/employee_rest/model/Skill.java - Eclipse IDE

```
package com.cognizant.employee_rest.model;

public class Skill {
    private int id;
    private String name;

    public Skill() {}

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
```

<terminated> SpringLearnApplication (1) [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\j

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.3)
```

3) **Create REST service to gets all employees**

EmployeeService.java

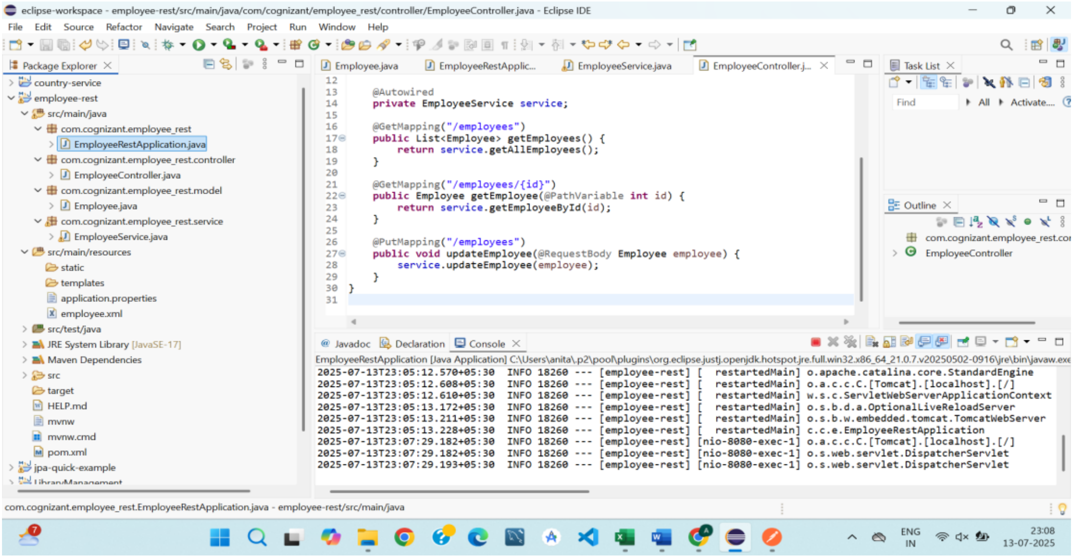
```
package com.cognizant.spring_learn.service;
import com.cognizant.spring_learn.dao.EmployeeDao;
import com.cognizant.spring_learn.model.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;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

EmployeeController.java

```
package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.model.Employee;
import com.cognizant.spring_learn.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();
  }
}
```

EmployeeDao.java

```
package com.cognizant.spring_learn.dao;
import com.cognizant.spring_learn.model.Employee;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;
import java.util.ArrayList;
@Repository
public class EmployeeDao {
  public static ArrayList<Employee> EMPLOYEE_LIST;
  public EmployeeDao() {
    ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");
    EMPLOYEE_LIST = (ArrayList<Employee>) context.getBean("employeeList");
  }
  public ArrayList<Employee> getAllEmployees() {
    return EMPLOYEE_LIST;
  }
}
```

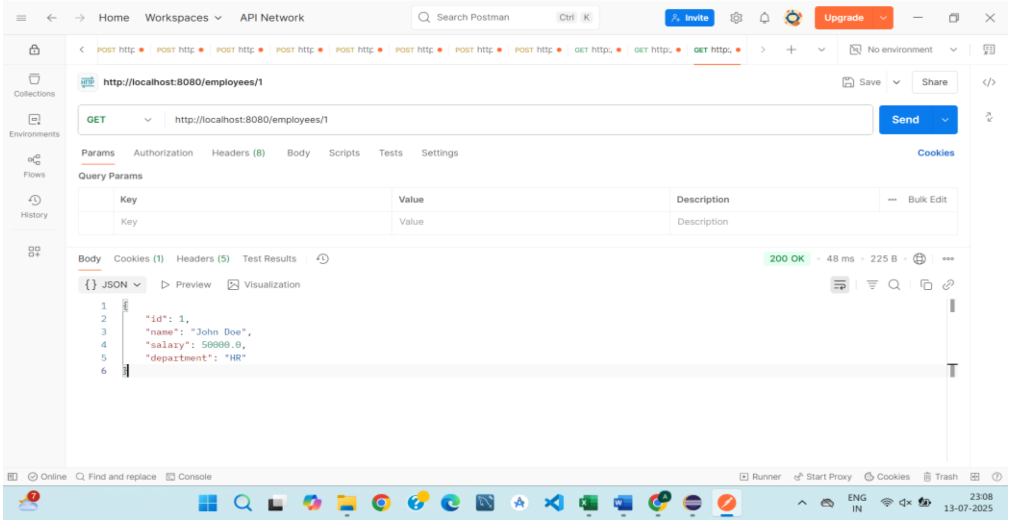


eclipse-workspace - employee-rest/src/main/java/com/cognizant/employee_rest/controller/EmployeeController.java - Eclipse IDE

```java
    @Autowired
    private EmployeeService service;

    @GetMapping("/employees")
    public List<Employee> getEmployees() {
        return service.getAllEmployees();
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployee(@PathVariable int id) {
        return service.getEmployeeById(id);
    }

    @PutMapping("/employees")
    public void updateEmployee(@RequestBody Employee employee) {
        service.updateEmployee(employee);
    }
}
```

EmployeeRestApplication [Java Application] C:\Users\anita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe

```
2025-07-13T23:05:12.570+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.apache.catalina.core.StandardEngine
2025-07-13T23:05:12.608+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-13T23:05:12.610+05:30  INFO 18260 --- [employee-rest] [  restartedMain] w.s.c.ServletWebServerApplicationContext
2025-07-13T23:05:13.172+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer
2025-07-13T23:05:13.211+05:30  INFO 18260 --- [employee-rest] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-13T23:05:13.228+05:30  INFO 18260 --- [employee-rest] [  restartedMain] c.c.e.EmployeeRestApplication
2025-07-13T23:07:29.182+05:30  INFO 18260 --- [employee-rest] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-13T23:07:29.182+05:30  INFO 18260 --- [employee-rest] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
2025-07-13T23:07:29.193+05:30  INFO 18260 --- [employee-rest] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
```



GET http://localhost:8080/employees/1

Body — 200 OK · 48 ms · 225 B

```json
{
    "id": 1,
    "name": "John Doe",
    "salary": 50000.0,
    "department": "HR"
}
```

4) Create REST service for department

Department.java

```java
package com.cognizant.spring_learn.model;
public class Department {
    private int id;
    private String name;
    public 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;
    }
}
```

Department.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
           http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="departmentList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <bean class="com.cognizant.spring_learn.model.Department">
                    <property name="id" value="1" />
                    <property name="name" value="HR" />
                </bean>
                <bean class="com.cognizant.spring_learn.model.Department">
                    <property name="id" value="2" />
                    <property name="name" value="IT" />
                </bean>
                <bean class="com.cognizant.spring_learn.model.Department">
                    <property name="id" value="3" />
                    <property name="name" value="Finance" />
                </bean>
            </list>
        </constructor-arg>
    </bean>
</beans>
```

DepartmentDao.java

```
package com.cognizant.spring_learn.dao;
import com.cognizant.spring_learn.model.Department;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;
import java.util.ArrayList;
@Repository
public class DepartmentDao {
   public static ArrayList<Department> DEPARTMENT_LIST;
    public DepartmentDao() {
        ApplicationContext context = new ClassPathXmlApplicationContext("department.xml");
        DEPARTMENT_LIST = (ArrayList<Department>) context.getBean("departmentList");
    }
    public ArrayList<Department> getAllDepartments() {
        return DEPARTMENT_LIST;
    }
}
```

<u>DepartmentService.java</u>

```
package com.cognizant.spring_learn.service;
import com.cognizant.spring_learn.dao.DepartmentDao;
import com.cognizant.spring_learn.model.Department;
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 DepartmentService {
  @Autowired
  private DepartmentDao departmentDao;
  @Transactional
  public List<Department> getAllDepartments() {
    return departmentDao.getAllDepartments(); }}
```

DepartmentController.java

```
package com.cognizant.spring_learn.controller;

import com.cognizant.spring_learn.model.Department;
import com.cognizant.spring_learn.service.DepartmentService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;
@RestController
public class DepartmentController {
  private static final Logger LOGGER = LoggerFactory.getLogger(DepartmentController.class);
  @Autowired
  private DepartmentService departmentService;
  @GetMapping("/departments")
  public List<Department> getAllDepartments() {
    LOGGER.info("START getAllDepartments()");
    List<Department> departments = departmentService.getAllDepartments();
    LOGGER.info("END getAllDepartments()");
    return departments;
  }
}
```

Response

```
[
 {
  "id": 1,
  "name": "HR"
 },
 {
    "id": 2,
    "name": "IT"
  },
  {
    "id": 3,
    "name": "Finance"
  }
]
```