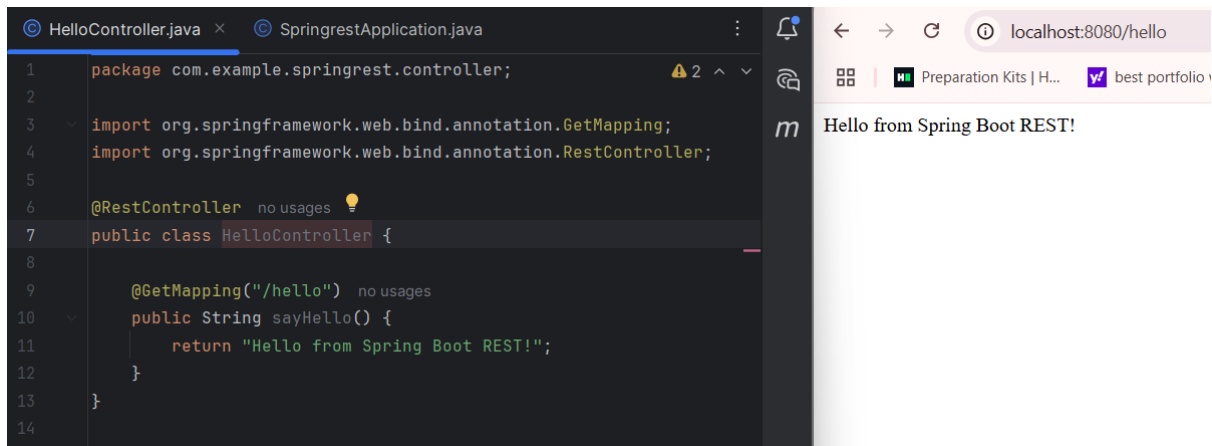


WEEK – 4

Spring REST using Spring Boot 3

Exercise 1: Create a Spring Web Project using Maven

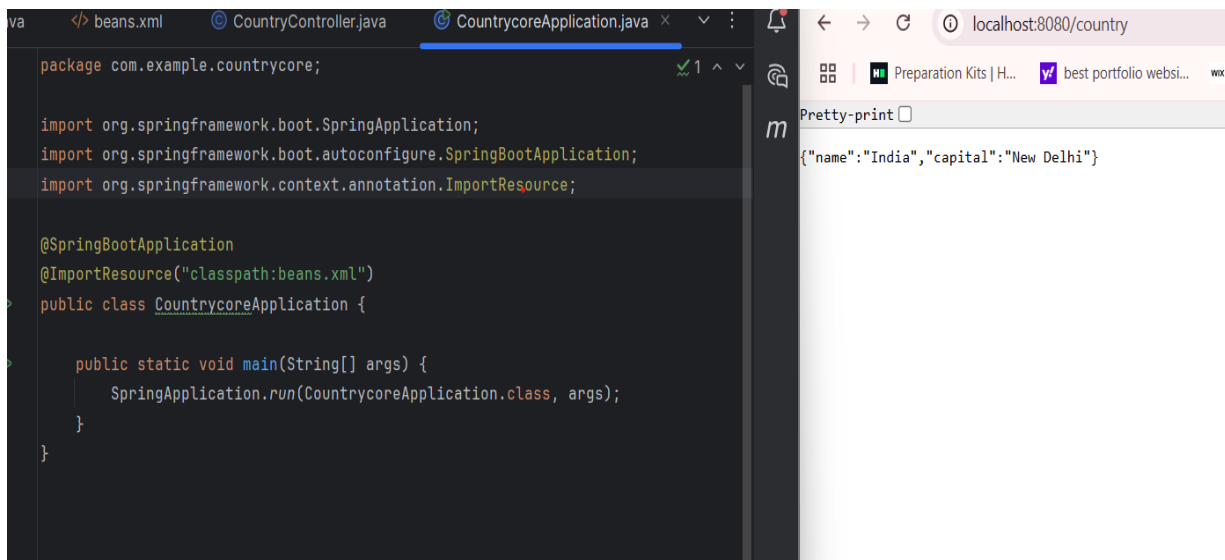


The screenshot shows an IDE with two tabs: `HelloController.java` and `SpringrestApplication.java`. The `HelloController.java` file contains the following code:

```
1 package com.example.springrest.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HelloController {
8
9     @GetMapping("/hello")
10    public String sayHello() {
11        return "Hello from Spring Boot REST!";
12    }
13 }
14
```

To the right, a web browser is open to `localhost:8080/hello`, displaying the response: `Hello from Spring Boot REST!`.

Exercise 2: Spring Core – Load Country from Spring Configuration XML



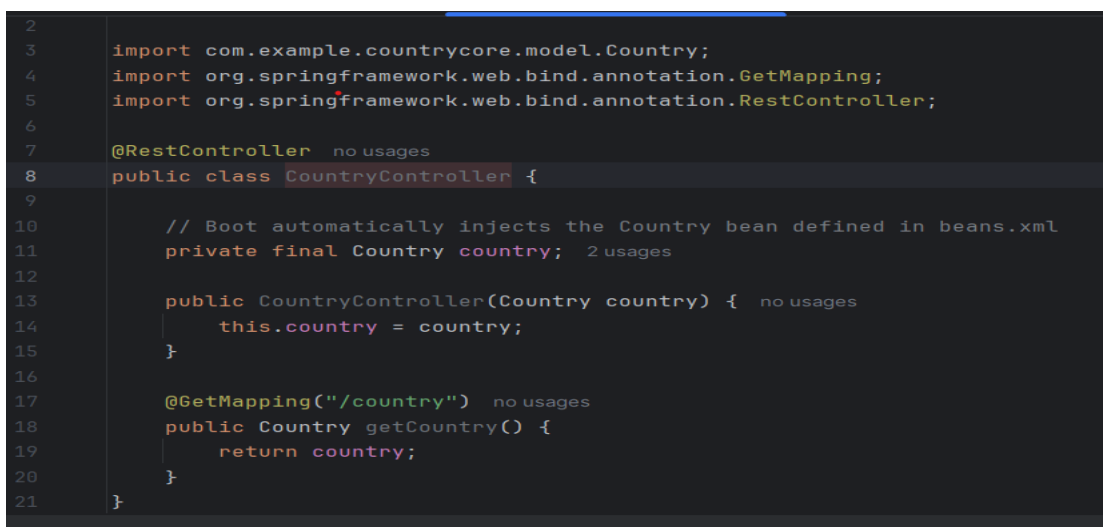
The screenshot shows an IDE with three tabs: `beans.xml`, `CountryController.java`, and `CountrycoreApplication.java`. The `CountrycoreApplication.java` file contains the following code:

```
package com.example.countrycore;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ImportResource;

@SpringBootApplication
@ImportResource("classpath:beans.xml")
public class CountrycoreApplication {

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

To the right, a web browser is open to `localhost:8080/country`, displaying the response: `{ "name": "India", "capital": "New Delhi" }`.



The screenshot shows the `CountryController.java` file in the IDE. The code is as follows:

```
2
3 import com.example.countrycore.model.Country;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController
8 public class CountryController {
9
10    // Boot automatically injects the Country bean defined in beans.xml
11    private final Country country;
12
13    public CountryController(Country country) {
14        this.country = country;
15    }
16
17    @GetMapping("/country")
18    public Country getCountry() {
19        return country;
20    }
21 }
```

```
Country.java  </> beans.xml  CountryController.java  CountrycoreApplication.java

<?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="country" class="com.example.countrycore.model.Country">
        <property name="name" value="India"/>
        <property name="capital" value="New Delhi"/>
    </bean>

</beans>
```

Exercise 3: Hello World RESTful Web Service

```
HelloController.java  HelloworldrestApplication.java

1  package com.example.helloworldrest.controller;
2
3  import org.springframework.web.bind.annotation.GetMapping;
4  import org.springframework.web.bind.annotation.RestController;
5
6  @RestController no usages
7  public class HelloController {
8
9      @GetMapping("/hello") no usages
10     public String sayHello() {
11         return "Hello World!";
12     }
13 }
14
```

localhost:8080/hello

Hello World!

Exercise 4: REST - Country Web Service

```
countryrest C:\Users\shalu\Downloads\c 1  package com.example.countryrest;
2
3  > import ...
4
5
6  @SpringBootApplication
7  public class CountryrestApplication {
8
9      public static void main(String[] args) {
10         SpringApplication.run(CountryrestApplication.class, args);
11     }
12 }
13
```

countryrest

- .idea
- countryrest
 - .mvn
 - src
 - main
 - java
 - com.example.countryrest
 - controller
 - CountryController
 - model
 - Country
 - resources

localhost:8080/country/IN

pretty-print ☐

"name":"India","capital":"New Delhi"}

```
8 public Country(int id, String name, String capital) {
9     this.id = id;
10    this.name = name;
11    this.capital = capital;
12 }
13 public int getId() { no usages
14     return id;
15 }
16 public void setId(int id) { 1 usage
17     this.id = id;
18 }
19 public String getName() { no usages
20     return name;
21 }
22 public void setName(String name) { 1 usage
23     this.name = name;
24 }
25 public String getCapital() { no usages
26     return capital;
27 }
28 public void setCapital(String capital) { 1 usage
29     this.capital = capital;
30 }
31 }
```

```
Country.java CountryController.java x CountryrestApplication.java
1 package com.example.countryrest.controller;
2
3 import com.example.countryrest.model.Country;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController no usages
8 public class CountryController {
9
10    @GetMapping("/country") no usages
11    public Country getCountry() {
12        Country country = new Country();
13        country.setId(101);
14        country.setName("India");
15        country.setCapital("New Delhi");
16        return country;
17    }
18 }
19 }
```

localhost:8080/country

Pretty-print ☐

{"id":101,"name":"India","capital":"New Delhi"}

Exercise 5: REST - Get country based on country code

```
countrycodeapi C:\Users\shalu\Downloads\countrycodeapi
> .idea
> countrycodeapi
> .mvn
> src
> main
> java
> com.example.countrycodeapi
> controller
> CountryController
> model
> Country
> CountrycodeapiApplication
> resources

1 package com.example.countrycodeapi;
2
3 > import ...
4
5 @SpringBootApplication
6 public class CountrycodeapiApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(CountrycodeapiApplication.class, args);
10    }
11
12 }
13
14 }
```

```

Country.java  CountryController.java  CountrycodeapiApplication.java
1  package com.example.countrycodeapi.controller;
2
3  import com.example.countrycodeapi.model.Country;
4  import org.springframework.web.bind.annotation.*;
5
6  import java.util.HashMap;
7  import java.util.Map;
8
9  @RestController  no usages
10 public class CountryController {
11     // Create in-memory data
12     private static final Map<String, Country> countryMap = new HashMap<>(); 5 usages
13     static {
14         countryMap.put("IN", new Country( name: "India", capital: "New Delhi"));
15         countryMap.put("US", new Country( name: "United States", capital: "Washington, D.C."));
16         countryMap.put("JP", new Country( name: "Japan", capital: "Tokyo"));
17         countryMap.put("FR", new Country( name: "France", capital: "Paris"));
18     }
19     // REST endpoint to get country by code
20     @GetMapping("/country/{code}")  no usages
21     public Country getCountryByCode(@PathVariable String code) {
22         Country country = countryMap.get(code.toUpperCase());
23         if (country != null) {
24             return country;
25         } else {
26             return new Country( name: "Not Found", capital: "N/A");
27         }
28     }
29 }

```

localhost:8080/country/JP

Preparation Kits | H... best portfolio we

Pretty-print ☐

```
{"name": "Japan", "capital": "Tokyo"}
```

localhost:8080/country/XYZ

Preparation Kits | H... best portfolio websi..

Pretty-print ☐

```
{"name": "Not Found", "capital": "N/A"}
```

localhost:8080/country/US

Preparation Kits | H... best portfolio websi... WIX WIX

Pretty-print ☐

```
{"name": "United States", "capital": "Washington, D.C."}
```

localhost:8080/country/FR

Preparation Kits | H... best portfolio we

Pretty-print ☐

```
{"name": "France", "capital": "Paris"}
```

Exercise 6: Create authentication service that returns JWT

```

Project Alt+1  AuthController.java  AuthRequest.java  AuthResponse.java  JwtauthserviceApplication.java
1  package com.example.jwtauthservice.model;
2
3  public class AuthRequest {  no usages
4      private String username; 3 usages
5      private String password; 3 usages
6
7      public AuthRequest() {}  no usages
8      public AuthRequest(String username, String password) {  no usages
9          this.username = username;
10         this.password = password;
11     }
12     public String getUsername() { return username; }  no usages
13     public void setUsername(String username) { this.username = username; }  no usages
14
15     public String getPassword() { return password; }  no usages
16     public void setPassword(String password) { this.password = password; }  no usages
17 }

```

```
1 package com.example.jwtauthservice.model;
2
3 public class AuthResponse { 3 usages
4     private String token; 3 usages
5
6     public AuthResponse() {} no usages
7     public AuthResponse(String token) { 1 usage
8         this.token = token;
9     }
10    public String getToken() { return token; } no usages
11    public void setToken(String token) { this.token = token; } no usages
12 }
```

```
1 package com.example.jwtauthservice.controller;
2
3 import com.example.jwtauthservice.model.AuthResponse;
4 import com.example.jwtauthservice.util.JwtUtil;
5 import org.springframework.web.bind.annotation.*;
6
7 @RestController no usages
8 @RequestMapping("/api")
9 public class AuthController {
10
11     private final JwtUtil jwtUtil; 2 usages
12
13     public AuthController(JwtUtil jwtUtil) { no usages
14         this.jwtUtil = jwtUtil;
15     }
16
17     @PostMapping(value = "/login", consumes = "application/x-www-form-urlencoded") no usages
18     public AuthResponse login(@RequestParam String username, @RequestParam String password) {
19         if ("admin".equals(username) && "admin123".equals(password)) {
20             String token = jwtUtil.generateToken(username);
21             return new AuthResponse(token);
22         } else {
23             throw new RuntimeException("Invalid credentials");
24         }
25     }
26 }
```

```
1 package com.example.jwtauthservice.config;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
5 import org.springframework.security.web.SecurityFilterChain;
6
7 @org.springframework.context.annotation.Configuration no usages
8 public class SecurityConfig {
9
10     @Bean no usages
11     public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
12         http
13             .csrf((CsrfConfigurer<HttpSecurity> csrf -> csrf.disable()))
14             .authorizeHttpRequests((AuthorizationManagerRequestMatcher<HttpServletRequest> auth -> auth.anyRequest().permitAll()))
15             .cors((CorsConfigurer<HttpSecurity> cors -> cors.disable());
16
17         return http.build();
18     }
19 }
```

```
1 package com.example.jwtauthservice.util;
2
3 import io.jsonwebtoken.Jwts;
4 import io.jsonwebtoken.SignatureAlgorithm;
5 import io.jsonwebtoken.security.Keys;
6 import org.springframework.stereotype.Component;
7
8 import java.security.Key;
9 import java.util.Date;
10 @Component 3 usages
11 public class JwtUtil {
12     private final Key secretKey = Keys.secretKeyFor(SignatureAlgorithm.HS256); 1 usage
13
14     public String generateToken(String username) { 1 usage
15         return Jwts.builder()
16             .setSubject(username)
17             .setIssuer("JwtAuthApp")
18             .setIssuedAt(new Date())
19             .setExpiration(new Date(System.currentTimeMillis() + 3600_000)) // 1 hour
20             .signWith(secretKey)
21             .compact();
22     }
23 }
```

JwtAuthServiceApplication.java

```
1 package com.example.jwtauthservice;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class JwtauthserviceApplication {
8     public static void main(String[] args) {
9         SpringApplication.run(JwtauthserviceApplication.class, args);
10    }
11 }
```

POST  <http://localhost:8080/api/login>

Params Authorization Headers (9) **Body** Scripts Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value	Description
✓	username	admin	
✓	password	admin123	
	Key	Value	Description

Body Cookies Headers (11) Test Results

200 OK • 1.40 s • 502

`{}` JSON Preview Visualize

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWwiOiJhZG1pb2IzImImlzcyI6Ikp3dEF1dGhBcHAiLCJpYXQiOiJlNTI0ODk5MjQsImV4cCI6MTc1MjQ5MzUyNH0.
3   bzja7V1oqAXM57R-qumvtAid-bD6qqouG8HuZs_K9fs"
4 }
```