

COGNIZANT D.N 4.0 JAVA FSE WEEK 4

SPRING REST

EXERCISE -1

Logging to Verify main()

```
package com.cognizant.springlearn;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
@SpringBootApplication
```

```
public class SpringLearnApplication {
```

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

```
    public static void main(String[] args) {
```

```
        LOGGER.info("Starting SpringLearnApplication...");
```

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

```
        LOGGER.info("SpringLearnApplication started successfully.");
```

```
    }
```

```
}
```

SpringLearnApplication.java

```
@SpringBootApplication
```

```
public class SpringLearnApplication {
```

```
    public static void main(String[] args) {
```

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

```
    }
```

```
}
```

pom.xml

```
<groupId>com.cognizant</groupId>  
<artifactId>spring-learn</artifactId>  
<version>0.0.1-SNAPSHOT</version>  
<packaging>jar</packaging>
```

```
<parent>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-parent</artifactId>  
  <version>3.0.0</version>  
</parent>
```

```
<dependencies>  
  <!-- Spring Web for REST controllers -->  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-web</artifactId>  
  </dependency>
```

```
  <!-- DevTools for hot reload -->  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-devtools</artifactId>  
    <scope>runtime</scope>  
  </dependency>
```

```
  <!-- Test dependencies -->  
  <dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-test</artifactId>  
    <scope>test</scope>  
  </dependency>
```

```
</dependencies>
```

```
<build>
```

```
  <plugins>
```

```
    <!-- Spring Boot Maven plugin -->
```

```
    <plugin>
```

```
      <groupId>org.springframework.boot</groupId>
```

```
      <artifactId>spring-boot-maven-plugin</artifactId>
```

```
    </plugin>
```

```
  </plugins>
```

```
</build>
```

## EXERCISE -2

SpringLearnApplication.java

```
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
```

```
import org.springframework.context.ApplicationContext;
```

```
import org.springframework.context.support.ClassPathXmlApplicationContext;
```

```
public class SpringLearnApplication {
```

```
  public static void displayDate() {
```

```
    ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
```

```
    SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);
```

```

try {
    Date date = format.parse("31/12/2018");
    System.out.println("Parsed Date: " + date);
} catch (ParseException e) {
    System.out.println("Error parsing date: " + e.getMessage());
}
}

```

```

public static void main(String[] args) {

    displayDate();
}
}

```

Port

server.port=8085

SpringLearnApplication

Parsed Date: wed july 10 00:00:00 IST 2025

### EXERCISE -3

displayDate()

```

public void displayDate() {

```

```

    LOGGER.info("START");

```

```

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

```

```

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

```

```

try {

```

```

    Date date = format.parse("31/12/2018");

```

```

    LOGGER.debug("Parsed Date: {}", date);

```

```

} catch (ParseException e) {

```

```

        LOGGER.error("Error parsing date: {}", e.getMessage());
    }

    LOGGER.info("END");
}

main()
public static void main(String[] args) {
    SpringLearnApplication app = new SpringLearnApplication();
    LOGGER.info("Application started");
    app.displayDate();
    LOGGER.info("Application ended");
}

```

#### EXERCISE -4

Country class

```
package com.cognizant.springlearn;
```

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
public class Country{
```

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

```
    private String code;
```

```
    private String name;
```

```
    public Country() {
```

```
        LOGGER.debug("Inside Country Constructor.");
```

```
}
```

```
public String getCode() {  
    LOGGER.debug("Inside getCode()");  
    return code;  
}
```

```
public void setCode(String code) {  
    LOGGER.debug("Inside setCode() with value: {}", code);  
    this.code = code;  
}
```

```
public String getName() {  
    LOGGER.debug("Inside getName()");  
    return name;  
}
```

```
public void setName(String name) {  
    LOGGER.debug("Inside setName() with value: {}", name);  
    this.name = name;  
}
```

```
@Override
```

```
public String toString() {  
    return "Country [code=" + code + ", name=" + name + "];"  
}
```

```
}
```

```
displayCountry()
```

```
import org.springframework.context.ApplicationContext;
```

```

import org.springframework.context.support.ClassPathXmlApplicationContext;

public void displayCountry() {
    LOGGER.info("START");

    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
    Country country = context.getBean("country", Country.class);

    LOGGER.debug("Country : {}", country.toString());

    LOGGER.info("END");
}

```

SpringLearnApplication.java

```

public static void main(String[] args) {
    SpringLearnApplication app = new SpringLearnApplication();
    LOGGER.info("Application started");
    app.displayCountry();
    LOGGER.info("Application ended");
}

```

## EXERCISE -5

Singleton Scope

```

public void displayCountry() {
    LOGGER.info("START");

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

    Country country = context.getBean("country", Country.class);
    LOGGER.debug("Country 1: {}", country.toString());
}

```

```

Country anotherCountry = context.getBean("country", Country.class);
LOGGER.debug("Country 2: {}", anotherCountry.toString());

LOGGER.info("END");
}

LOGGER.debug("Are both references same? {}", country == anotherCountry);

```

## EXERCISE -6

```

displayCountries()
import java.util.List;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public void displayCountries() {
    LOGGER.info("START");

    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
    @SuppressWarnings("unchecked")
    List<Country> countryList = (List<Country>) context.getBean("countryList");

    for (Country country : countryList) {
        LOGGER.debug("Country: {}", country);
    }

    LOGGER.info("END");
}

main()
public static void main(String[] args) {

```



```
SpringLearnApplication app = new SpringLearnApplication();  
LOGGER.info("Application started");  
app.displayCountries();  
LOGGER.info("Application ended");  
}
```