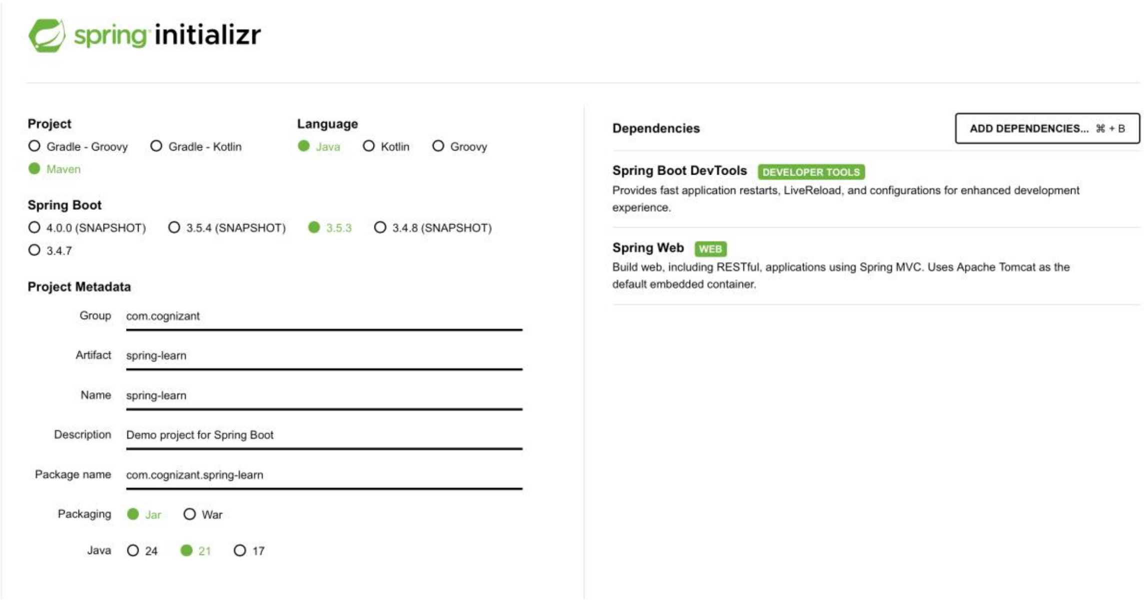
HANDSON EXERCISES - WEEK 4

Skill : Spring REST using Spring Boot 3

File : 1. spring-rest-handson.docx

Exercise 1 : Create a Spring Web Project using Maven

Spring Initializr Configuration for spring-learn Project



spring initializr

Project
○ Gradle - Groovy ○ Gradle - Kotlin
● Maven

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 4.0.0 (SNAPSHOT) ○ 3.5.4 (SNAPSHOT) ● 3.5.3 ○ 3.4.8 (SNAPSHOT)
○ 3.4.7

Project Metadata

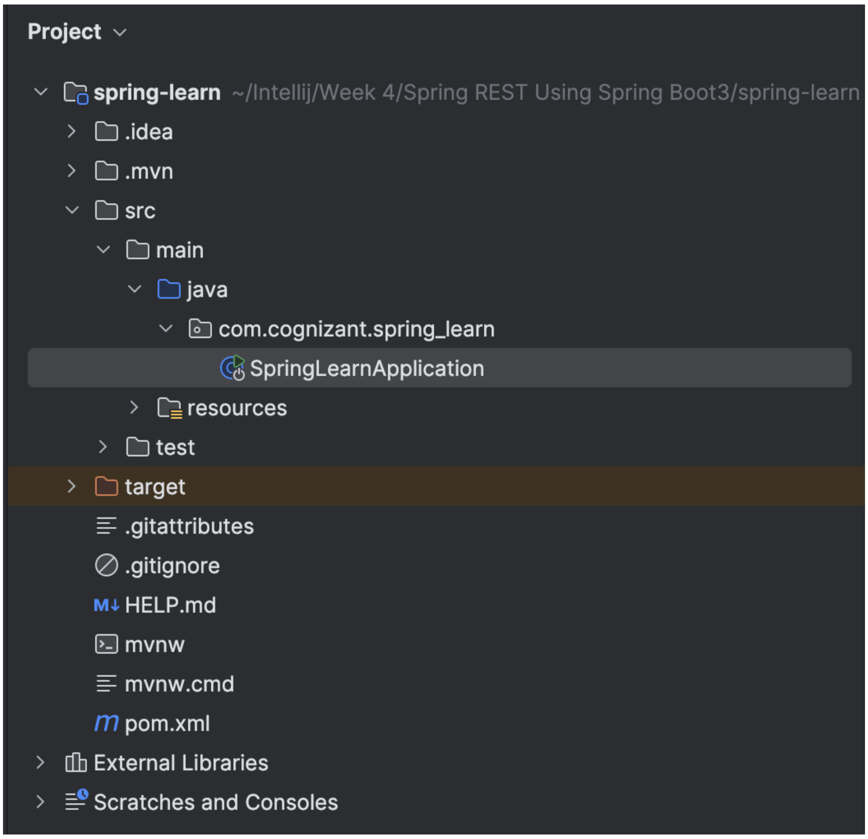
Group	com.cognizant
Artifact	spring-learn
Name	spring-learn
Description	Demo project for Spring Boot
Package name	com.cognizant.spring-learn
Packaging	● Jar ○ War
Java	○ 24 ● 21 ○ 17

Dependencies ADD DEPENDENCIES... ⌘ + B

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Project Structure :



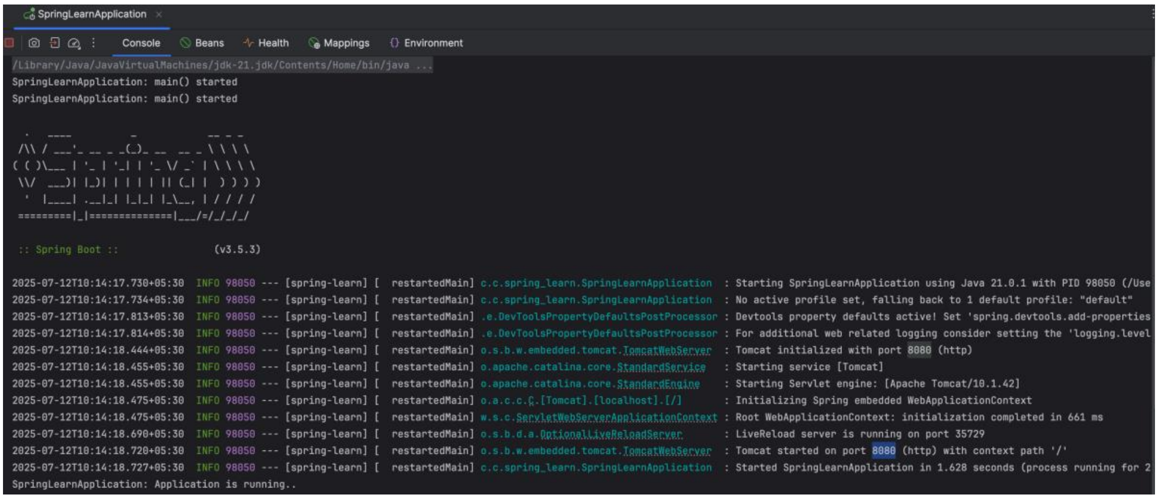
```
Project
  spring-learn ~/Intellij/Week 4/Spring REST Using Spring Boot3/spring-learn
    .idea
    .mvn
    src
      main
        java
          com.cognizant.spring_learn
            SpringLearnApplication
        resources
      test
    target
    .gitattributes
    .gitignore
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
  External Libraries
  Scratches and Consoles
```

CODE :

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("SpringLearnApplication: main() started");
        SpringApplication.run(SpringLearnApplication.class, args);
        System.out.println("SpringLearnApplication: Application is running..");
    }
}
```

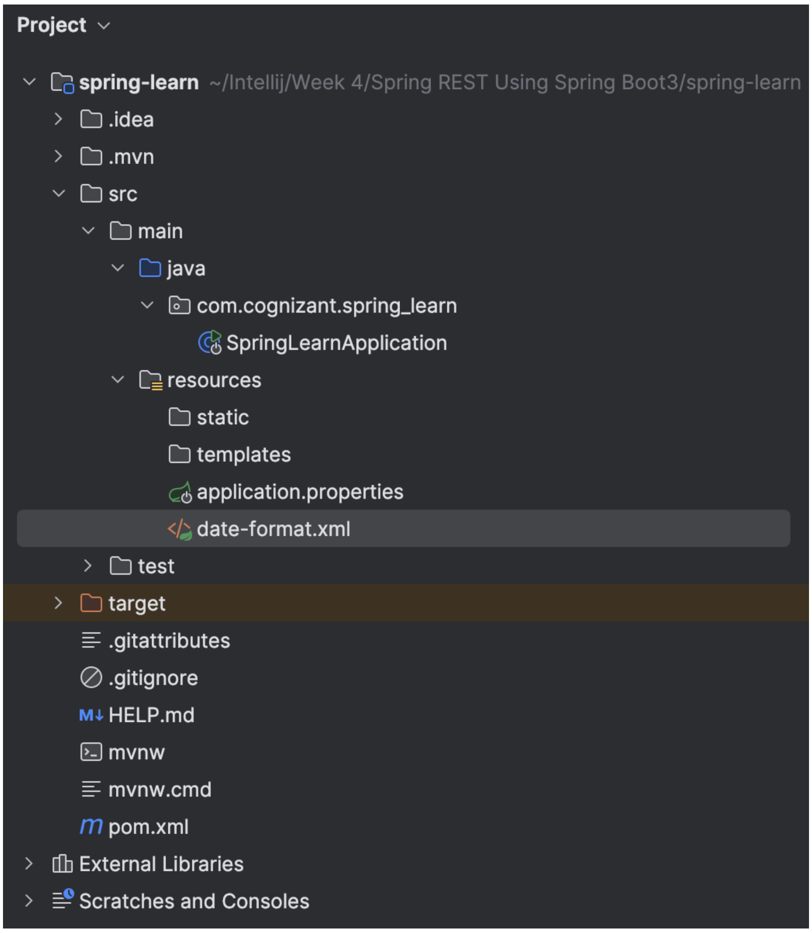
OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...
SpringLearnApplication: main() started
SpringLearnApplication: main() started

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

 :: Spring Boot ::                (v3.5.3)

2025-07-12T10:14:17.730+05:30  INFO 98050 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  : Starting SpringLearnApplication using Java 21.0.1 with PID 98050 (/Use
2025-07-12T10:14:17.734+05:30  INFO 98050 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  : No active profile set, falling back to 1 default profile: "default"
2025-07-12T10:14:17.813+05:30  INFO 98050 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties
2025-07-12T10:14:17.814+05:30  INFO 98050 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level
2025-07-12T10:14:18.444+05:30  INFO 98050 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2025-07-12T10:14:18.455+05:30  INFO 98050 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-07-12T10:14:18.455+05:30  INFO 98050 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-12T10:14:18.475+05:30  INFO 98050 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-07-12T10:14:18.475+05:30  INFO 98050 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 661 ms
2025-07-12T10:14:18.690+05:30  INFO 98050 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-12T10:14:18.720+05:30  INFO 98050 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2025-07-12T10:14:18.727+05:30  INFO 98050 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  : Started SpringLearnApplication in 1.628 seconds (process running for 2
SpringLearnApplication: Application is running..
```

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

Project Structure :



```
Project ⌄

⌄ spring-learn  ~/Intellij/Week 4/Spring REST Using Spring Boot3/spring-learn
    > .idea
    > .mvn
    ⌄ src
        ⌄ main
            ⌄ java
                ⌄ com.cognizant.spring_learn
                        SpringLearnApplication
            ⌄ resources
                    static
                    templates
                    application.properties
                    date-format.xml
        > test
    > target
      .gitattributes
      .gitignore
      HELP.md
      mvnw
      mvnw.cmd
      pom.xml
> External Libraries
> Scratches and Consoles
```

CODE :

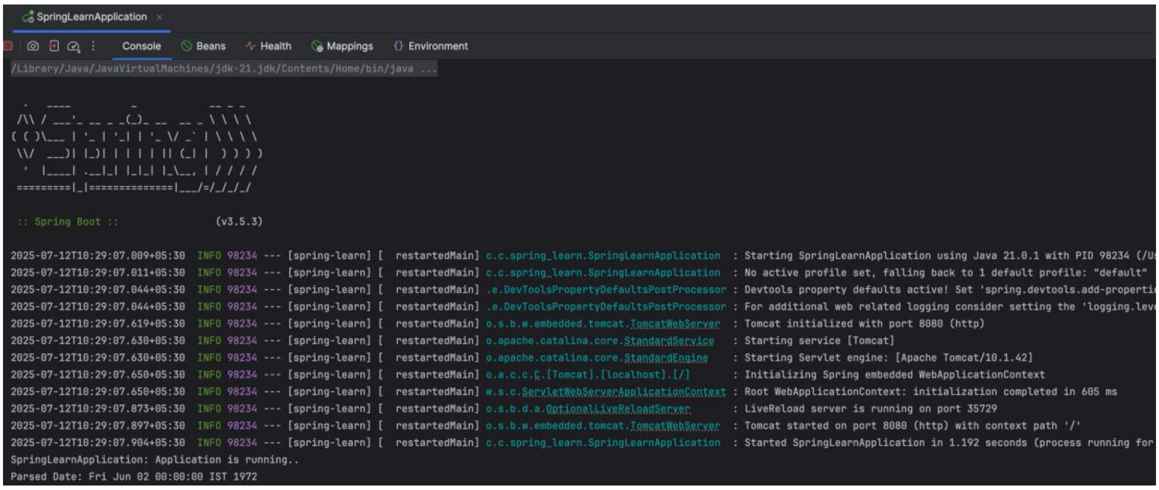
src/resources/date-fromat.xml :

```
<?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.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

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);
        System.out.println("SpringLearnApplication: Application is running..");

        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("Parsed Date: " + parsedDate);
    }
}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...

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

 :: Spring Boot ::                (v3.5.3)

2025-07-12T10:29:07.009+05:30  INFO 98234 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  : Starting SpringLearnApplication using Java 21.0.1 with PID 98234 (/U
2025-07-12T10:29:07.011+05:30  INFO 98234 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  : No active profile set, falling back to 1 default profile: "default"
2025-07-12T10:29:07.044+05:30  INFO 98234 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properti
2025-07-12T10:29:07.044+05:30  INFO 98234 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.lev
2025-07-12T10:29:07.619+05:30  INFO 98234 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2025-07-12T10:29:07.630+05:30  INFO 98234 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-07-12T10:29:07.630+05:30  INFO 98234 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-12T10:29:07.650+05:30  INFO 98234 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-07-12T10:29:07.650+05:30  INFO 98234 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 605 ms
2025-07-12T10:29:07.873+05:30  INFO 98234 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-12T10:29:07.897+05:30  INFO 98234 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2025-07-12T10:29:07.904+05:30  INFO 98234 --- [spring-learn] [  restartedMain] c.c.spring_learn.SpringLearnApplication  : Started SpringLearnApplication in 1.192 seconds (process running for
SpringLearnApplication: Application is running..
Parsed Date: Fri Jun 02 00:00:00 IST 1972
```

Exercise 3 : Spring Core - Incorporate Logging

CODE :

resources/application.properties :

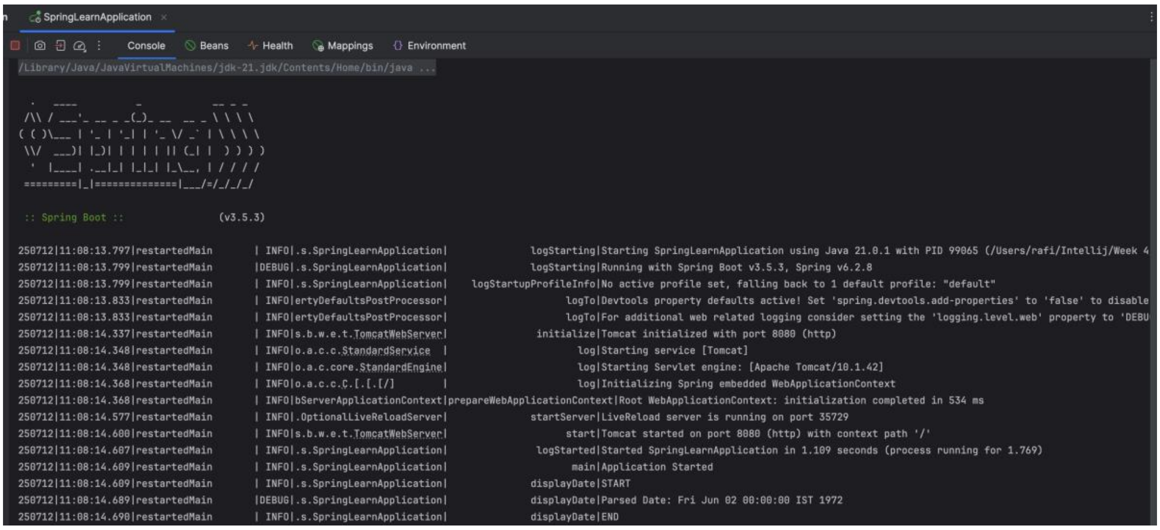
```
spring.application.name=spring-learn

logging.level.org.springframework=info
logging.level.com.cognizant.spring_learn=debug
logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n
```

SpringLearnApplication.java :

```
package com.cognizant.spring_learn;

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

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

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) throws Exception {
      SpringApplication.run(SpringLearnApplication.class, args);
      LOGGER.info("Application Started");

      displayDate();
   }

   public static void displayDate() throws Exception {
      LOGGER.info("START");

      ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
      SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

      Date parsedDate = format.parse("02/06/1972");

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

```
    LOGGER.info("END");
  }
}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...

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

 :: Spring Boot ::                (v3.5.3)

250712|11:08:13.797|restartedMain      | INFO|.s.SpringLearnApplication|           logStarting|Starting SpringLearnApplication using Java 21.0.1 with PID 99065 (/Users/rafi/Intellij/Week 4
250712|11:08:13.799|restartedMain      |DEBUG|.s.SpringLearnApplication|           logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
250712|11:08:13.799|restartedMain      | INFO|.s.SpringLearnApplication|    logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
250712|11:08:13.833|restartedMain      | INFO|ertyDefaultsPostProcessor|                 logTo|Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
250712|11:08:13.833|restartedMain      | INFO|ertyDefaultsPostProcessor|                 logTo|For additional web related logging consider setting the 'logging.level.web' property to 'DEBU
250712|11:08:14.337|restartedMain      | INFO|s.b.w.e.t.TomcatWebServer|            initialize|Tomcat initialized with port 8080 (http)
250712|11:08:14.348|restartedMain      | INFO|o.a.c.c.StandardService  |                   log|Starting service [Tomcat]
250712|11:08:14.348|restartedMain      | INFO|o.a.c.core.StandardEngine|                   log|Starting Servlet engine: [Apache Tomcat/10.1.42]
250712|11:08:14.368|restartedMain      | INFO|o.a.c.c.C.[.[.[/]        |                   log|Initializing Spring embedded WebApplicationContext
250712|11:08:14.368|restartedMain      | INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 534 ms
250712|11:08:14.577|restartedMain      | INFO|.OptionalLiveReloadServer|           startServer|LiveReload server is running on port 35729
250712|11:08:14.600|restartedMain      | INFO|s.b.w.e.t.TomcatWebServer|                 start|Tomcat started on port 8080 (http) with context path '/'
250712|11:08:14.607|restartedMain      | INFO|.s.SpringLearnApplication|            logStarted|Started SpringLearnApplication in 1.109 seconds (process running for 1.769)
250712|11:08:14.609|restartedMain      | INFO|.s.SpringLearnApplication|                  main|Application Started
250712|11:08:14.609|restartedMain      | INFO|.s.SpringLearnApplication|           displayDate|START
250712|11:08:14.689|restartedMain      |DEBUG|.s.SpringLearnApplication|           displayDate|Parsed Date: Fri Jun 02 00:00:00 IST 1972
250712|11:08:14.690|restartedMain      | INFO|.s.SpringLearnApplication|           displayDate|END
```

Exercise 4 : Spring Core – Load Country from Spring Configuration XML

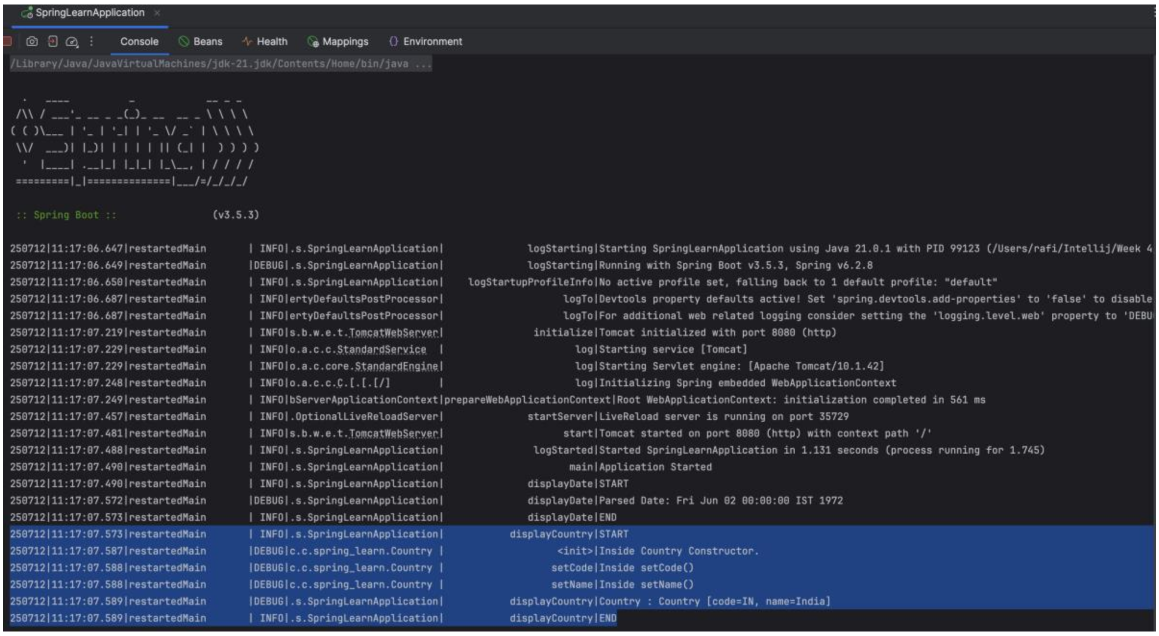
resources/country.xml :

```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="country" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>

</beans>
```

SpringLearnApplication.java :

```java
package com.cognizant.spring_learn;

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

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import java.text.SimpleDateFormat;
import java.util.Date;

@SpringBootApplication
public class SpringLearnApplication {

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

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

      LOGGER.info("Application Started");

      displayDate();

      displayCountry();

   }

   public static void displayDate() throws Exception {
      LOGGER.info("START");

      ApplicationContext context = new ClassPathXmlApplicationContext("date-
format.xml");
      SimpleDateFormat format = context.getBean("dateFormat",
SimpleDateFormat.class);

      Date parsedDate = format.parse("02/06/1972");
      LOGGER.debug("Parsed Date: {}", parsedDate);

      LOGGER.info("END");

   }

   public static 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");

   }

}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...

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

 :: Spring Boot ::                (v3.5.3)

250712|11:17:06.647|restartedMain        | INFO|.s.SpringLearnApplication|           logStarting|Starting SpringLearnApplication using Java 21.0.1 with PID 99123 (/Users/rafi/Intellij/Week 4
250712|11:17:06.649|restartedMain        |DEBUG|.s.SpringLearnApplication|           logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
250712|11:17:06.650|restartedMain        | INFO|.s.SpringLearnApplication|  logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
250712|11:17:06.687|restartedMain        | INFO|ertyDefaultsPostProcessor|                 logTo|Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
250712|11:17:06.687|restartedMain        | INFO|ertyDefaultsPostProcessor|                 logTo|For additional web related logging consider setting the 'logging.level.web' property to 'DEBU
250712|11:17:07.219|restartedMain        | INFO|s.b.w.e.t.TomcatWebServer|            initialize|Tomcat initialized with port 8080 (http)
250712|11:17:07.229|restartedMain        | INFO|o.a.c.c.StandardService  |                   log|Starting service [Tomcat]
250712|11:17:07.229|restartedMain        | INFO|o.a.c.core.StandardEngine|                   log|Starting Servlet engine: [Apache Tomcat/10.1.42]
250712|11:17:07.248|restartedMain        | INFO|o.a.c.c.C.[.[.[/]        |                   log|Initializing Spring embedded WebApplicationContext
250712|11:17:07.249|restartedMain        | INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 561 ms
250712|11:17:07.457|restartedMain        | INFO|.OptionalLiveReloadServer|           startServer|LiveReload server is running on port 35729
250712|11:17:07.481|restartedMain        | INFO|s.b.w.e.t.TomcatWebServer|                 start|Tomcat started on port 8080 (http) with context path '/'
250712|11:17:07.488|restartedMain        | INFO|.s.SpringLearnApplication|            logStarted|Started SpringLearnApplication in 1.131 seconds (process running for 1.745)
250712|11:17:07.490|restartedMain        | INFO|.s.SpringLearnApplication|                  main|Application Started
250712|11:17:07.490|restartedMain        | INFO|.s.SpringLearnApplication|           displayDate|START
250712|11:17:07.572|restartedMain        |DEBUG|.s.SpringLearnApplication|           displayDate|Parsed Date: Fri Jun 02 00:00:00 IST 1972
250712|11:17:07.573|restartedMain        | INFO|.s.SpringLearnApplication|           displayDate|END
250712|11:17:07.573|restartedMain        | INFO|.s.SpringLearnApplication|        displayCountry|START
250712|11:17:07.587|restartedMain        |DEBUG|c.c.spring_learn.Country |                <init>|Inside Country Constructor.
250712|11:17:07.588|restartedMain        |DEBUG|c.c.spring_learn.Country |               setCode|Inside setCode()
250712|11:17:07.588|restartedMain        |DEBUG|c.c.spring_learn.Country |               setName|Inside setName()
250712|11:17:07.589|restartedMain        |DEBUG|.s.SpringLearnApplication|        displayCountry|Country : Country [code=IN, name=India]
250712|11:17:07.589|restartedMain        | INFO|.s.SpringLearnApplication|        displayCountry|END
```

Exercise 5 : Spring Core – Demonstration of Singleton Scope and Prototype Scope

Singleton Scope :

resources/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">

    <bean id="country" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>

</beans>
```

SpringLearnApplication.java :

```
package com.cognizant.spring_learn;

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

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

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

@SpringBootApplication
public class SpringLearnApplication {

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

    public static void main(String[] args) throws Exception {
        SpringApplication.run(SpringLearnApplication.class, args);
        LOGGER.info("Application Started");

        displayDate();
        displayCountry();
    }

    public static void displayDate() throws Exception {
        LOGGER.info("START");

        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

        Date parsedDate = format.parse("02/06/1972");
        LOGGER.debug("Parsed Date: {}", parsedDate);

        LOGGER.info("END");
    }

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

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

        Country country1 = context.getBean("country", Country.class);
        Country country2 = context.getBean("country", Country.class);

        LOGGER.debug("Country 1 : {}", country1);
        LOGGER.debug("Country 2 : {}", country2);

        LOGGER.info("END");
    }
}
```

Prototype Scope -

country.xml :

```
<bean id="country" class="com.cognizant.spring_learn.Country" scope="prototype">
    <property name="code" value="IN"/>
    <property name="name" value="India"/>
</bean>
```

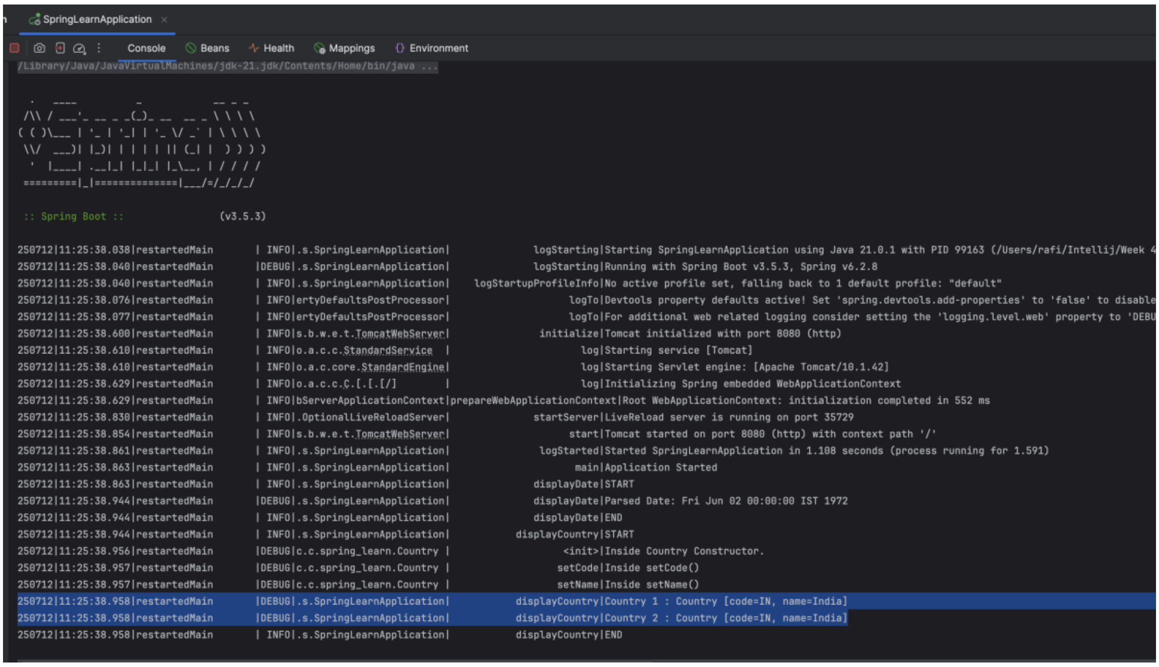
SpringLearnApplication.java :

The same code , here we’ll be checking whether the objects are same objects or not.

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

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

    Country country1 = context.getBean("country", Country.class);
    Country country2 = context.getBean("country", Country.class);

    LOGGER.debug("Country 1 : {}", country1);
    LOGGER.debug("Country 2 : {}", country2);
    LOGGER.debug("Are both objects same? {}", country1 == country2);

    LOGGER.info("END");
}
```

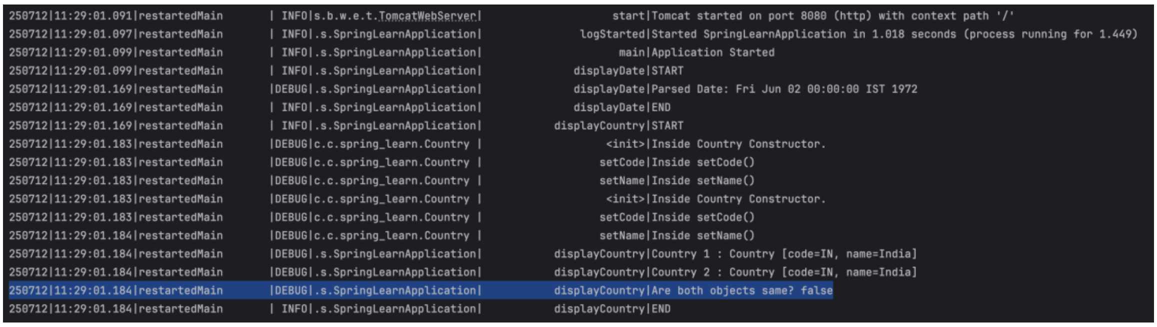
OUTPUT :

Singleton Scope -



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...

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

 :: Spring Boot ::                (v3.5.3)

250712|11:25:38.038|restartedMain       | INFO|.s.SpringLearnApplication|           logStarting|Starting SpringLearnApplication using Java 21.0.1 with PID 99163 (/Users/rafi/Intellij/Week 4
250712|11:25:38.040|restartedMain       |DEBUG|.s.SpringLearnApplication|           logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
250712|11:25:38.040|restartedMain       | INFO|.s.SpringLearnApplication|    logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
250712|11:25:38.076|restartedMain       | INFO|ertyDefaultsPostProcessor|                 logTo|Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
250712|11:25:38.077|restartedMain       | INFO|ertyDefaultsPostProcessor|                 logTo|For additional web related logging consider setting the 'logging.level.web' property to 'DEBU
250712|11:25:38.600|restartedMain       | INFO|s.b.w.e.t.TomcatWebServer|            initialize|Tomcat initialized with port 8080 (http)
250712|11:25:38.610|restartedMain       | INFO|o.a.c.c.StandardService  |                   log|Starting service [Tomcat]
250712|11:25:38.610|restartedMain       | INFO|o.a.c.core.StandardEngine|                   log|Starting Servlet engine: [Apache Tomcat/10.1.42]
250712|11:25:38.629|restartedMain       | INFO|o.a.c.c.C.[.[.[/]        |                   log|Initializing Spring embedded WebApplicationContext
250712|11:25:38.629|restartedMain       | INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 552 ms
250712|11:25:38.830|restartedMain       | INFO|.OptionalLiveReloadServer|           startServer|LiveReload server is running on port 35729
250712|11:25:38.854|restartedMain       | INFO|s.b.w.e.t.TomcatWebServer|                 start|Tomcat started on port 8080 (http) with context path '/'
250712|11:25:38.861|restartedMain       | INFO|.s.SpringLearnApplication|            logStarted|Started SpringLearnApplication in 1.108 seconds (process running for 1.591)
250712|11:25:38.863|restartedMain       | INFO|.s.SpringLearnApplication|                  main|Application Started
250712|11:25:38.863|restartedMain       | INFO|.s.SpringLearnApplication|           displayDate|START
250712|11:25:38.944|restartedMain       |DEBUG|.s.SpringLearnApplication|           displayDate|Parsed Date: Fri Jun 02 00:00:00 IST 1972
250712|11:25:38.944|restartedMain       | INFO|.s.SpringLearnApplication|           displayDate|END
250712|11:25:38.944|restartedMain       | INFO|.s.SpringLearnApplication|        displayCountry|START
250712|11:25:38.956|restartedMain       |DEBUG|c.c.spring_learn.Country |                <init>|Inside Country Constructor.
250712|11:25:38.957|restartedMain       |DEBUG|c.c.spring_learn.Country |               setCode|Inside setCode()
250712|11:25:38.957|restartedMain       |DEBUG|c.c.spring_learn.Country |               setName|Inside setName()
250712|11:25:38.958|restartedMain       |DEBUG|.s.SpringLearnApplication|        displayCountry|Country 1 : Country [code=IN, name=India]
250712|11:25:38.958|restartedMain       |DEBUG|.s.SpringLearnApplication|        displayCountry|Country 2 : Country [code=IN, name=India]
250712|11:25:38.958|restartedMain       | INFO|.s.SpringLearnApplication|        displayCountry|END
```

Prototype Scope -



```
250712|11:29:01.091|restartedMain       | INFO|s.b.w.e.t.TomcatWebServer|                 start|Tomcat started on port 8080 (http) with context path '/'
250712|11:29:01.097|restartedMain       | INFO|.s.SpringLearnApplication|            logStarted|Started SpringLearnApplication in 1.018 seconds (process running for 1.449)
250712|11:29:01.099|restartedMain       | INFO|.s.SpringLearnApplication|                  main|Application Started
250712|11:29:01.099|restartedMain       | INFO|.s.SpringLearnApplication|           displayDate|START
250712|11:29:01.169|restartedMain       |DEBUG|.s.SpringLearnApplication|           displayDate|Parsed Date: Fri Jun 02 00:00:00 IST 1972
250712|11:29:01.169|restartedMain       | INFO|.s.SpringLearnApplication|           displayDate|END
250712|11:29:01.169|restartedMain       | INFO|.s.SpringLearnApplication|        displayCountry|START
250712|11:29:01.183|restartedMain       |DEBUG|c.c.spring_learn.Country |                <init>|Inside Country Constructor.
250712|11:29:01.183|restartedMain       |DEBUG|c.c.spring_learn.Country |               setCode|Inside setCode()
250712|11:29:01.183|restartedMain       |DEBUG|c.c.spring_learn.Country |               setName|Inside setName()
250712|11:29:01.183|restartedMain       |DEBUG|c.c.spring_learn.Country |                <init>|Inside Country Constructor.
250712|11:29:01.183|restartedMain       |DEBUG|c.c.spring_learn.Country |               setCode|Inside setCode()
250712|11:29:01.184|restartedMain       |DEBUG|c.c.spring_learn.Country |               setName|Inside setName()
250712|11:29:01.184|restartedMain       |DEBUG|.s.SpringLearnApplication|        displayCountry|Country 1 : Country [code=IN, name=India]
250712|11:29:01.184|restartedMain       |DEBUG|.s.SpringLearnApplication|        displayCountry|Country 2 : Country [code=IN, name=India]
250712|11:29:01.184|restartedMain       |DEBUG|.s.SpringLearnApplication|        displayCountry|Are both objects same? false
250712|11:29:01.184|restartedMain       | INFO|.s.SpringLearnApplication|        displayCountry|END
```

Exercise 6 : Spring Core – Load list of countries from Spring Configuration XML

CODE:

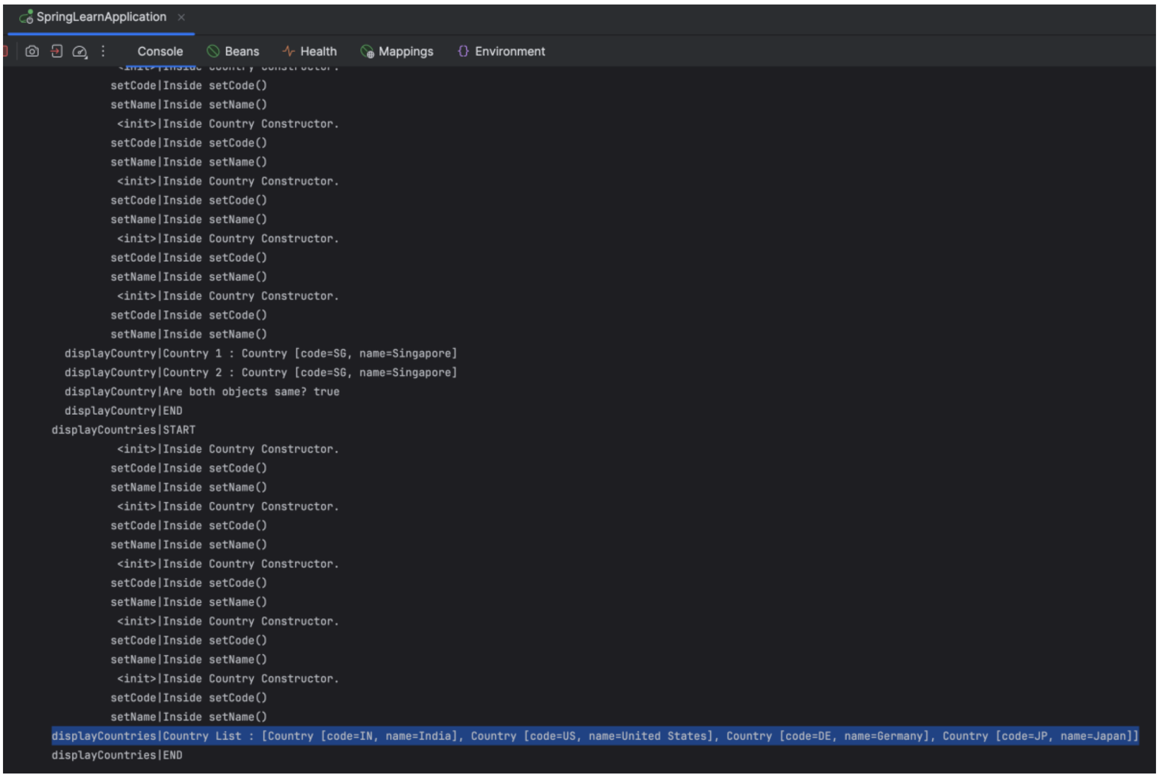
resources/country.xml :

```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="country" class="com.cognizant.spring_learn.Country" scope="singleton">
        <property name="code" value="SG"/>
        <property name="name" value="Singapore"/>
    </bean>

    <bean id="in" class="com.cognizant.spring_learn.Country">
        <property name="code" value="IN"/>
        <property name="name" value="India"/>
    </bean>

    <bean id="us" class="com.cognizant.spring_learn.Country">
        <property name="code" value="US"/>
        <property name="name" value="United States"/>
    </bean>

    <bean id="de" class="com.cognizant.spring_learn.Country">
        <property name="code" value="DE"/>
        <property name="name" value="Germany"/>
    </bean>

    <bean id="jp" class="com.cognizant.spring_learn.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="de"/>
                <ref bean="jp"/>
            </list>
        </constructor-arg>
    </bean>

</beans>
```

SpringLearnApplication.java :

```java
package com.cognizant.spring_learn;

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

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

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

@SpringBootApplication
public class SpringLearnApplication {

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

    public static void main(String[] args) throws Exception {
        SpringApplication.run(SpringLearnApplication.class, args);
        LOGGER.info("Application Started");

        displayDate();
        displayCountry();
        displayCountries();
    }

    public static void displayDate() throws Exception {
        LOGGER.info("START");

        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

        Date parsedDate = format.parse("02/06/1972");
        LOGGER.debug("Parsed Date: {}", parsedDate);

        LOGGER.info("END");
    }

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

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

        Country country1 = context.getBean("country", Country.class);
        Country country2 = context.getBean("country", Country.class);
        LOGGER.debug("Country 1 : {}", country1);
        LOGGER.debug("Country 2 : {}", country2);
        LOGGER.debug("Are both objects same? {}", country1 == country2);

        LOGGER.info("END");
    }

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

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

        List<Country> countryList = context.getBean("countryList", List.class);
        LOGGER.debug("Country List : {}", countryList);

        LOGGER.info("END");
    }
}
```

OUTPUT:

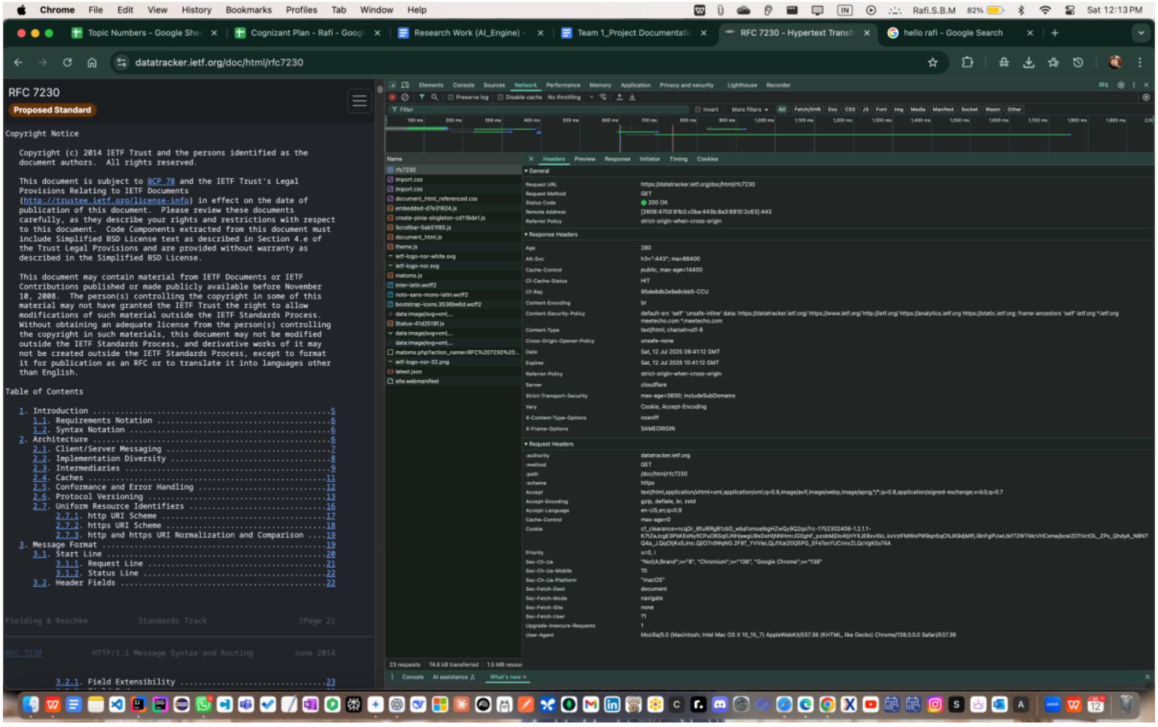


```
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
     displayCountry|Country 1 : Country [code=SG, name=Singapore]
     displayCountry|Country 2 : Country [code=SG, name=Singapore]
     displayCountry|Are both objects same? true
     displayCountry|END
   displayCountries|START
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
             <init>|Inside Country Constructor.
            setCode|Inside setCode()
            setName|Inside setName()
   displayCountries|Country List : [Country [code=IN, name=India], Country [code=US, name=United States], Country [code=DE, name=Germany], Country [code=JP, name=Japan]]
   displayCountries|END
```

File : 2. spring-rest-handson.docx

Exercise 1 : HTTP Request Response

OUTPUT : General , Response & Request Headers.



datatracker.ietf.org/doc/html/rfc7230

RFC 7230
Proposed Standard

```
Copyright Notice

   Copyright (c) 2014 IETF Trust and the persons identified as the
   document authors.  All rights reserved.

   This document is subject to BCP 78 and the IETF Trust's Legal
   Provisions Relating to IETF Documents
   (http://trustee.ietf.org/license-info) in effect on the date of
   publication of this document.  Please review these documents
   carefully, as they describe your rights and restrictions with respect
   to this document.  Code Components extracted from this document must
   include Simplified BSD License text as described in Section 4.e of
   the Trust Legal Provisions and are provided without warranty as
   described in the Simplified BSD License.

   This document may contain material from IETF Documents or IETF
   Contributions published or made publicly available before November
   10, 2008.  The person(s) controlling the copyright in some of this
   material may not have granted the IETF Trust the right to allow
   modifications of such material outside the IETF Standards Process.
   Without obtaining an adequate license from the person(s) controlling
   the copyright in such materials, this document may not be modified
   outside the IETF Standards Process, and derivative works of it may
   not be created outside the IETF Standards Process, except to format
   it for publication as an RFC or to translate it into languages other
   than English.

Table of Contents

   1. Introduction ....................................................5
     1.1. Requirements Notation .......................................6
     1.2. Syntax Notation .............................................6
   2. Architecture ....................................................6
     2.1. Client/Server Messaging .....................................7
     2.2. Implementation Diversity ....................................8
     2.3. Intermediaries ..............................................9
     2.4. Caches .....................................................11
     2.5. Conformance and Error Handling .............................12
     2.6. Protocol Versioning ........................................13
     2.7. Uniform Resource Identifiers ...............................16
          2.7.1. http URI Scheme .....................................17
          2.7.2. https URI Scheme ....................................18
          2.7.3. http and https URI Normalization and Comparison ....19
   3. Message Format .................................................19
     3.1. Start Line .................................................20
          3.1.1. Request Line ........................................21
          3.1.2. Status Line .........................................22
     3.2. Header Fields ..............................................22


Fielding & Reschke           Standards Track                    [Page 2]

RFC 7230           HTTP/1.1 Message Syntax and Routing         June 2014


          3.2.1. Field Extensibility .................................23
```

Name
rfc7230
import.css
import.css
document_html_referenced.css
embedded-d7e31924.js
create-pinia-singleton-cd118de1.js
Scrollbar-6ab51f85.js
document_html.js
theme.js
ietf-logo-nor-white.svg
ietf-logo-nor.svg
matomo.js
inter-latin.woff2
noto-sans-mono-latin.woff2
bootstrap-icons.3536be6d.woff2
data:image/svg+xml,...
Status-41d2518f.js
data:image/svg+xml,...
data:image/svg+xml,...
matomo.php?action_name=RFC%207230%20...
ietf-logo-nor-32.png
latest.json
site.webmanifest

23 requests | 74.6 kB transferred | 1.5 MB resour

Headers

▾ General

Request URL	https://datatracker.ietf.org/doc/html/rfc7230
Request Method	GET
Status Code	● 200 OK
Remote Address	[2606:4700:91b2:c0ba:443b:8a3:6810:2c63]:443
Referrer Policy	strict-origin-when-cross-origin

▾ Response Headers

Age	280
Alt-Svc	h3=":443"; ma=86400
Cache-Control	public, max-age=14400
Cf-Cache-Status	HIT
Cf-Ray	95de8db2e9a9cbb5-CCU
Content-Encoding	br
Content-Security-Policy	default-src 'self' 'unsafe-inline' data: https://datatracker.ietf.org/ https://www.ietf.org/ http://ietf.org/ https://analytics.ietf.org https://static.ietf.org; frame-ancestors 'self' ietf.org *.ietf.org meetecho.com *.meetecho.com
Content-Type	text/html; charset=utf-8
Cross-Origin-Opener-Policy	unsafe-none
Date	Sat, 12 Jul 2025 06:41:12 GMT
Expires	Sat, 12 Jul 2025 10:41:12 GMT
Referrer-Policy	strict-origin-when-cross-origin
Server	cloudflare
Strict-Transport-Security	max-age=3600; includeSubDomains
Vary	Cookie, Accept-Encoding
X-Content-Type-Options	nosniff
X-Frame-Options	SAMEORIGIN

▾ Request Headers

:authority	datatracker.ietf.org
:method	GET
:path	/doc/html/rfc7230
:scheme	https
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	en-US,en;q=0.9
Cache-Control	max-age=0
Cookie	cf_clearance=ncqDr_6fui6RgB1zbD_wba1omoelkgHZwQy9Q2qo7rc-1752302408-1.2.1.1-K7tZeJcgE3PbKEsNyfICPuO6Sq0JNHjeegU9sDsHljNNHmrJG0ghF_pzobMjDo4tjHYXJEBxvXkL.kxVzlFMWxPW9qn5qCNJK9djMPj.lBnFgPUwLtkf72WTMcVHCsmejbcelZO1VctOL_ZPx_QhdyA_NBNTQ4a_J.QqOtjKx5Jmc.QjIO7rdWqNG.2F9T_YVVIeLQjJfXal20Q5PG_EFeTexYUCnnxZLQcVgK0o74A
Priority	u=0, i
Sec-Ch-Ua	"Not)A;Brand";v="8", "Chromium";v="138", "Google Chrome";v="138"
Sec-Ch-Ua-Mobile	?0
Sec-Ch-Ua-Platform	"macOS"
Sec-Fetch-Dest	document
Sec-Fetch-Mode	navigate
Sec-Fetch-Site	none
Sec-Fetch-User	?1
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat 12:13 PM
Topic Numbers - Google Shee
Cognizant Plan - Rafi - Goog
Research Work (AI_Engine)
Team 1_Project Documentati
RFC 7230 - Hypertext Transf
hello rafi - Google Search
Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse Recorder
Preserve log Disable cache No throttling
Filter Invert More filters All Fetch/XHR Doc CSS JS Font Img Media Manifest Socket Wasm Other
Headers Preview Response Initiator Timing Cookies
Console AI assistance What's new

Exercise 2 : Hello World RESTful Web Service

CODE:

application.properties :

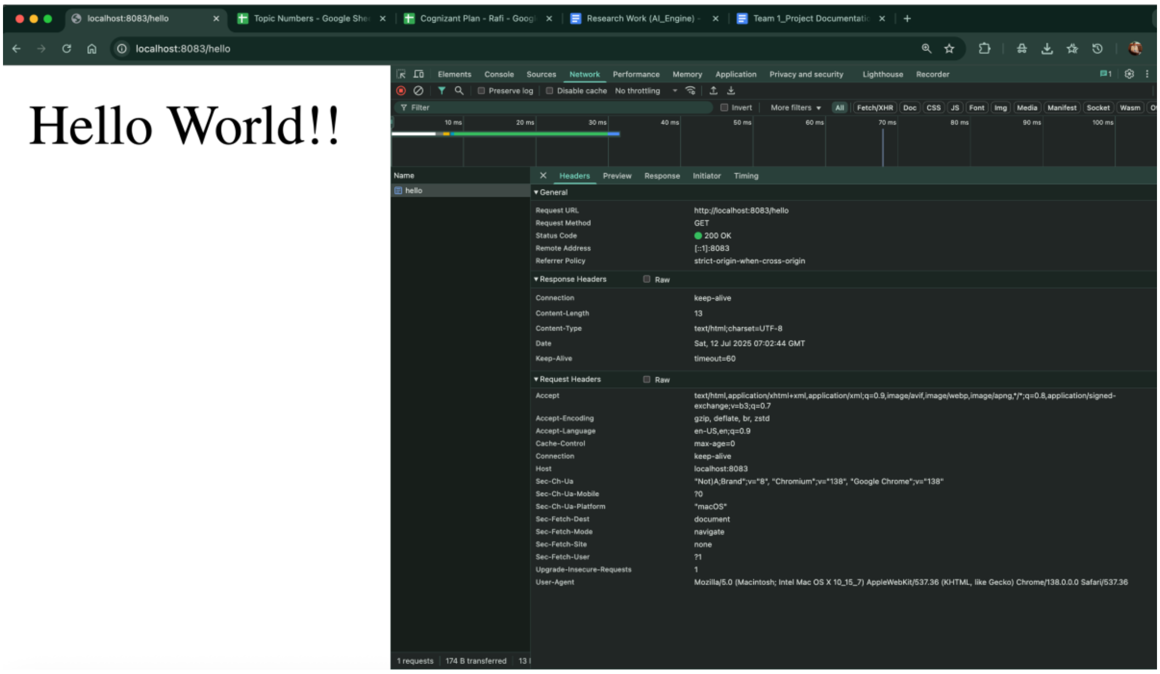
```
spring.application.name=spring-learn

logging.level.org.springframework=info
logging.level.com.cognizant.spring_learn=debug
logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n

server.port=8083
```

com.cognizant.spring_learn.controller/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 {
```

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

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

OUTPUT : DevTools



localhost:8083/hello

Hello World!!

Headers | Preview | Response | Initiator | Timing

▾ General

```
Request URL              http://localhost:8083/hello
Request Method           GET
Status Code              ● 200 OK
Remote Address           [::1]:8083
Referrer Policy          strict-origin-when-cross-origin
```

▾ Response Headers □ Raw

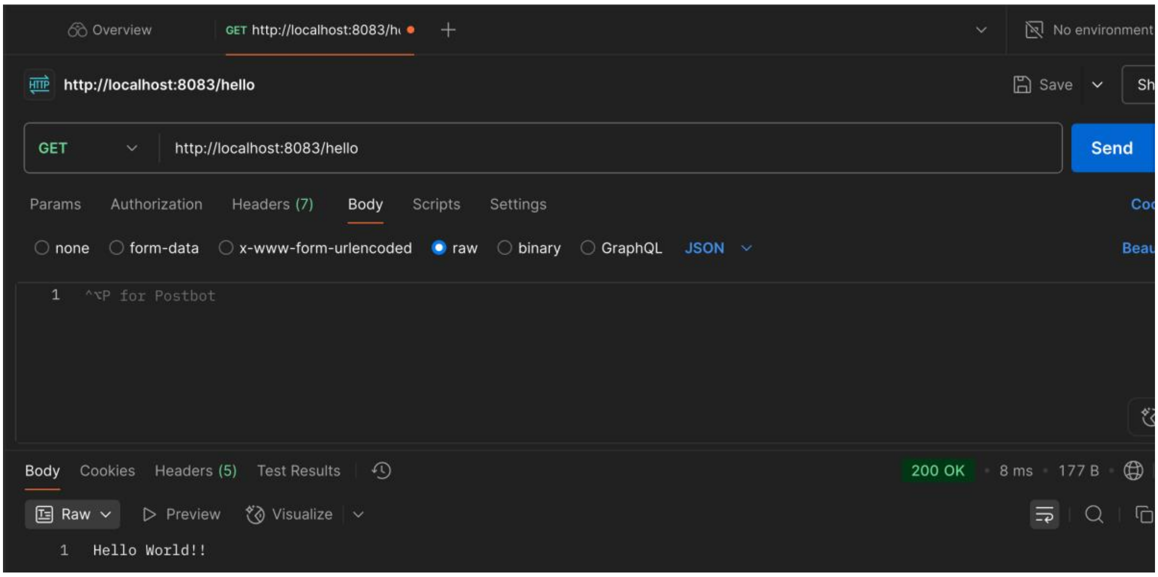
```
Connection               keep-alive
Content-Length           13
Content-Type             text/html;charset=UTF-8
Date                     Sat, 12 Jul 2025 07:02:44 GMT
Keep-Alive               timeout=60
```

▾ Request Headers □ Raw

```
Accept                     text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
                           exchange;v=b3;q=0.7
Accept-Encoding            gzip, deflate, br, zstd
Accept-Language            en-US,en;q=0.9
Cache-Control              max-age=0
Connection                 keep-alive
Host                       localhost:8083
Sec-Ch-Ua                  "Not)A;Brand";v="8", "Chromium";v="138", "Google Chrome";v="138"
Sec-Ch-Ua-Mobile           ?0
Sec-Ch-Ua-Platform         "macOS"
Sec-Fetch-Dest             document
Sec-Fetch-Mode             navigate
Sec-Fetch-Site             none
Sec-Fetch-User             ?1
Upgrade-Insecure-Requests  1
User-Agent                 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
```

1 requests | 174 B transferred

Postman -



GET http://localhost:8083/hello

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
1  ^⌥P for Postbot
```

Response: 200 OK · 8 ms · 177 B

```
1  Hello World!!
```

Exercise 3 : REST - Country Web Service

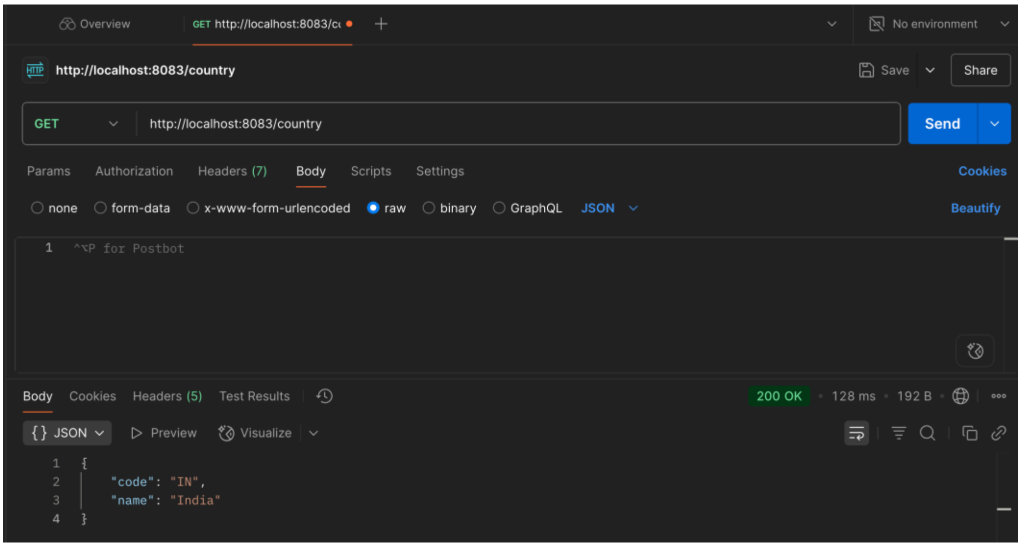
com.cognizant.spring_learn.controller/CountryController.java :

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

import com.cognizant.spring_learn.Country;
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;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@RestController
public class CountryController {

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

    @RequestMapping("/country")
    public Country getCountryIndia() {
        LOGGER.info("START");

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

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

        LOGGER.info("END");
        return india;
    }
}
```

resources/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">

<bean id="country" class="com.cognizant.spring_learn.Country">
    <property name="code" value="IN"/>
    <property name="name" value="India"/>
</bean>

</beans
```

OUTPUT:



GET http://localhost:8083/country

```
^⌥P for Postbot
```

Body — 200 OK · 128 ms · 192 B

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

Exercise 4 : REST - Get all countries

CODE:

resources/country.xml :

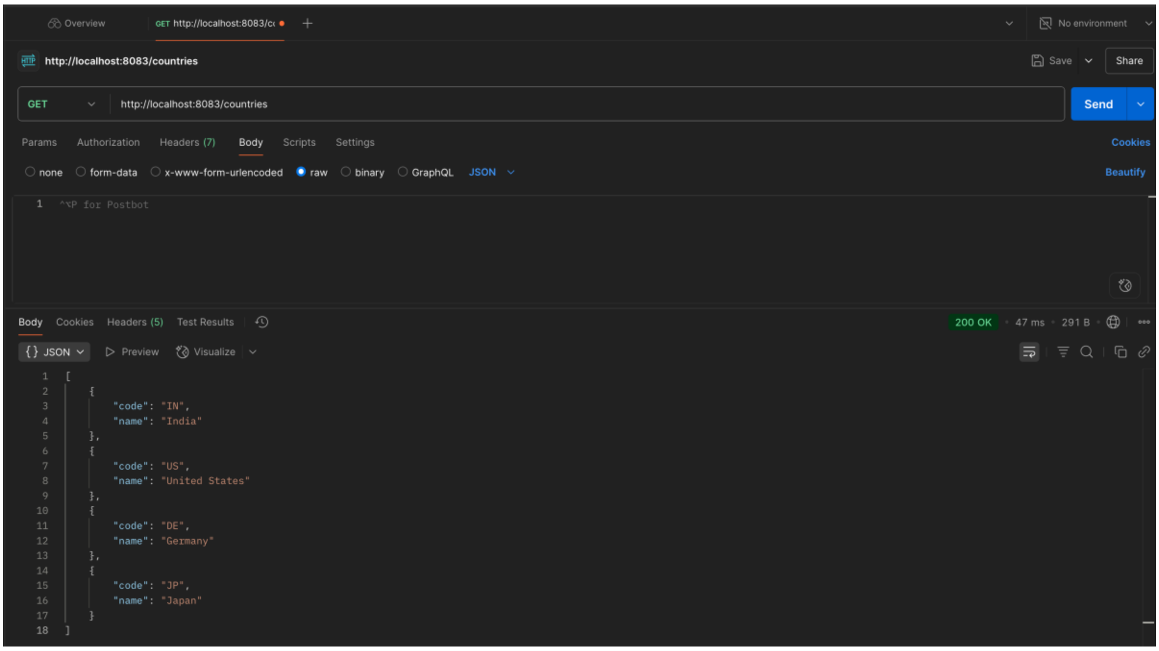
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="country" class="com.cognizant.spring_learn.Country">
      <property name="code" value="IN"/>
      <property name="name" value="India"/>
  </bean>

  <bean id="us" class="com.cognizant.spring_learn.Country">
      <property name="code" value="US"/>
      <property name="name" value="United States"/>
  </bean>

  <bean id="de" class="com.cognizant.spring_learn.Country">
      <property name="code" value="DE"/>
      <property name="name" value="Germany"/>
  </bean>

  <bean id="jp" class="com.cognizant.spring_learn.Country">
      <property name="code" value="JP"/>
      <property name="name" value="Japan"/>
  </bean>
```

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

        <constructor-arg>
            <list>
                <ref bean="country"/>
                <ref bean="us"/>
                <ref bean="de"/>
                <ref bean="jp"/>
            </list>
        </constructor-arg>

    </bean>

</beans>
```

CountryController.java :

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

import com.cognizant.spring_learn.Country;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
public class CountryController {

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

    @GetMapping("/country")
    public Country getCountryIndia() {
        LOGGER.info("START");

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

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

        return country;
    }

    @GetMapping("/countries")
    public List<Country> getAllCountries() {
        LOGGER.info("START");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        List<Country> countryList = context.getBean("countryList", List.class);

        LOGGER.debug("Country List: {}", countryList);
        LOGGER.info("END");

        return countryList;
    }
}
```

OUTPUT:



GET http://localhost:8083/countries

http://localhost:8083/countries

200 OK · 47 ms · 291 B

```
[
    {
        "code": "IN",
        "name": "India"
    },
    {
        "code": "US",
        "name": "United States"
    },
    {
        "code": "DE",
        "name": "Germany"
    },
    {
        "code": "JP",
        "name": "Japan"
    }
]
```

Exercise 5 : REST - Get country based on country code

CODE:

CountryController.java :

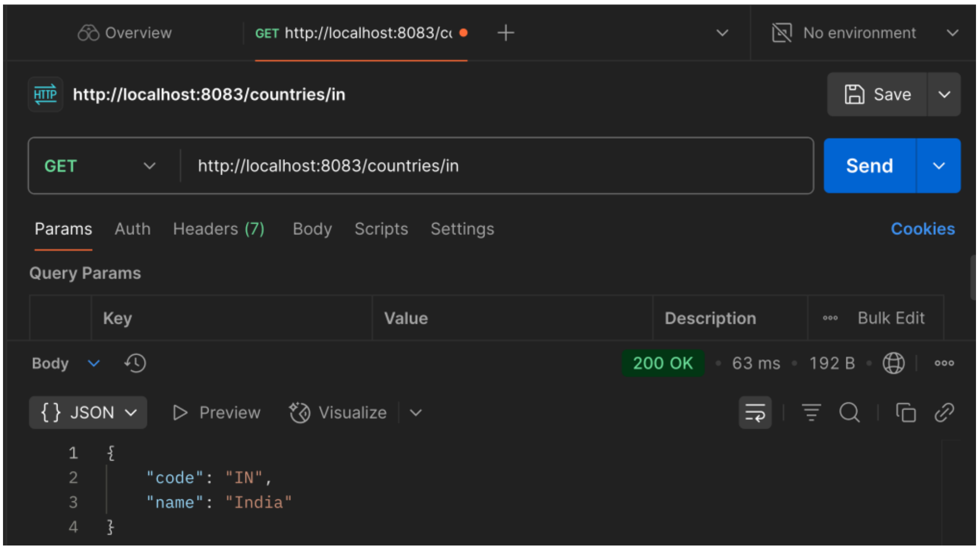
Added a method -

```
    @GetMapping("/countries/{code}")
    public Country getCountry(@PathVariable String code) {
        LOGGER.info("START");
        Country country = countryService.getCountry(code);
        LOGGER.debug("Fetched Country for code {}: {}", code, country);
        LOGGER.info("END");
        return country;
    }
```

service/CountryService.java :

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

import com.cognizant.spring_learn.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("country.xml");
        List<Country> countries = context.getBean("countryList", List.class);

        return countries.stream()
                .filter(c -> c.getCode().equalsIgnoreCase(code))
                .findFirst()
                .orElse(null);
    }
}
```

OUTPUT :

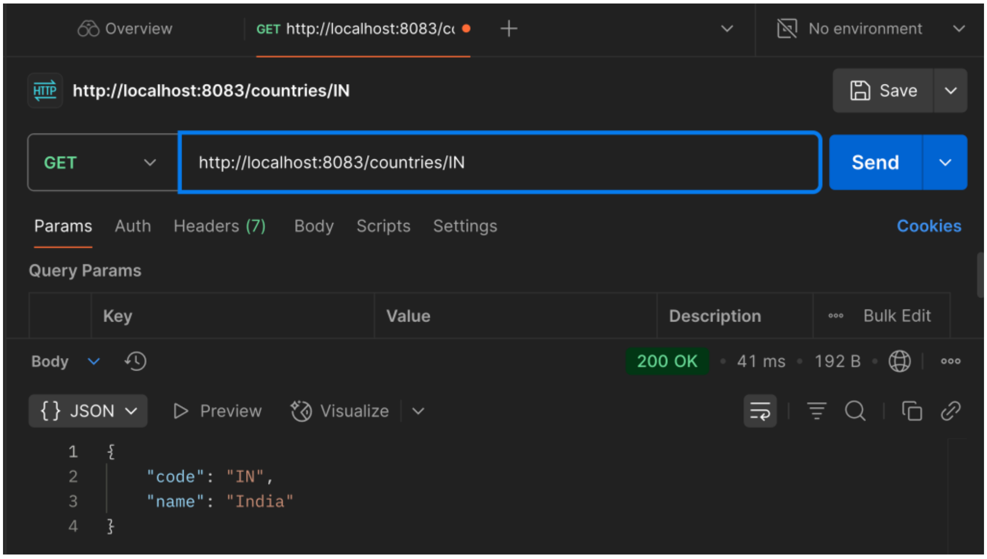


http://localhost:8083/countries/in

GET http://localhost:8083/countries/in Send

200 OK • 63 ms • 192 B

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

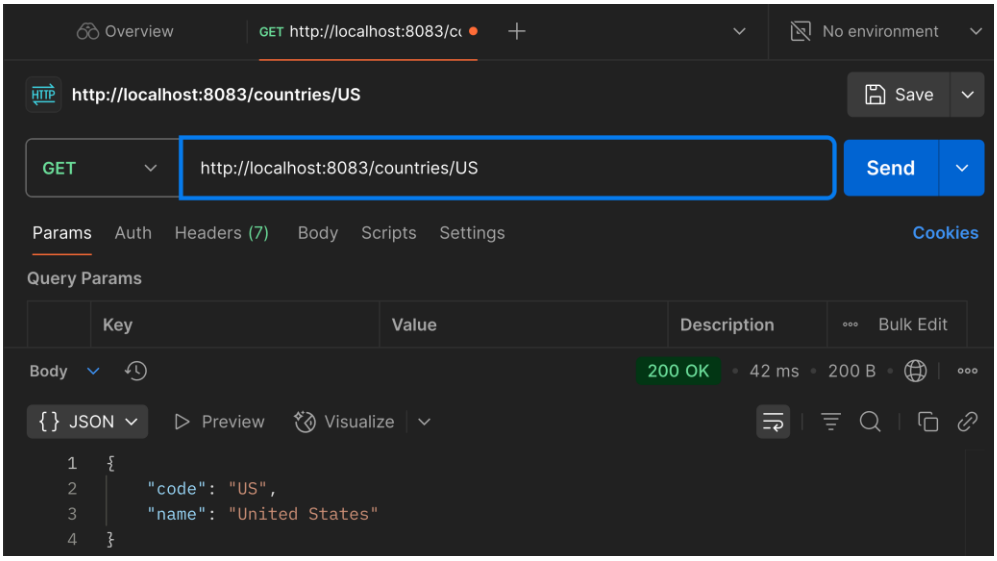


http://localhost:8083/countries/IN

GET http://localhost:8083/countries/IN Send

200 OK • 41 ms • 192 B

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



http://localhost:8083/countries/US

GET http://localhost:8083/countries/US

200 OK • 42 ms • 200 B

```
{
    "code": "US",
    "name": "United States"
}
```

File : 5. JWT-handson

com.cognizant.spring_learn.controller/AuthController.java :

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

import com.cognizant.spring_learn.security.JwtUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.*;

import java.util.*;

@RestController
public class AuthController {

    @Autowired
    private JwtUtil jwtUtil;

    @GetMapping("/authenticate")
    public ResponseEntity<Map<String, String>> authenticate(Authentication authentication) {
        String username = authentication.getName();
        String token = jwtUtil.generateToken(username);

        Map<String, String> result = new HashMap<>();
        result.put("token", token);

        return ResponseEntity.ok(result);
    }
}
```

JwtUtil.java :

```java
package com.cognizant.spring_learn.security;

import io.jsonwebtoken.*;
import io.jsonwebtoken.security.Keys;
import org.springframework.stereotype.Component;

import java.security.Key;
import java.util.Date;

@Component
public class JwtUtil {

    private final String SECRET_KEY = "ThisIsASecretKeyForJwtGeneration1234567890!";
    private final long EXPIRATION_TIME = 1000 * 60 * 60;
    private Key getSigningKey() {
        return Keys.hmacShaKeyFor(SECRET_KEY.getBytes());
    }

    public String generateToken(String username) {
        return Jwts.builder()
                .setSubject(username)
                .setIssuedAt(new Date(System.currentTimeMillis()))
                .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
                .signWith(getSigningKey(), SignatureAlgorithm.HS256)
                .compact();
    }

    public String extractUsername(String token) {
        return Jwts.parserBuilder()
                .setSigningKey(getSigningKey())
                .build()
                .parseClaimsJws(token)
                .getBody()
                .getSubject();
    }

    public boolean validateToken(String token, String username) {
        String extractedUsername = extractUsername(token);
        return extractedUsername.equals(username) && !isTokenExpired(token);
    }

    private boolean isTokenExpired(String token) {
        Date expiration = Jwts.parserBuilder()
                .setSigningKey(getSigningKey())
                .build()
                .parseClaimsJws(token)
                .getBody()
                .getExpiration();
        return expiration.before(new Date());
    }
}
```

SecurityConfig.java :

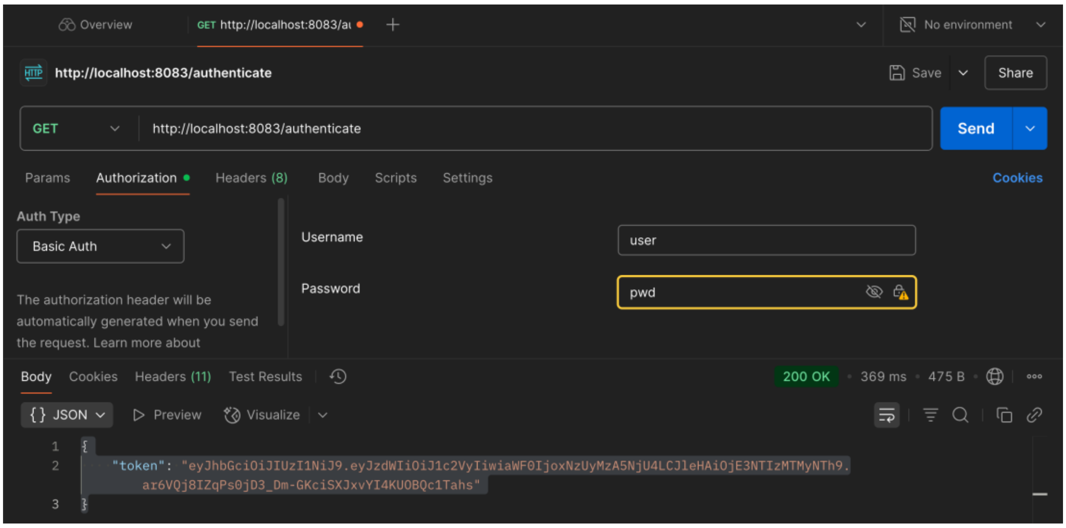
```java
package com.cognizant.spring_learn.security;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.authentication.AuthenticationManagerResolver;
import org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;
import
```

```java
org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;

@Configuration
@EnableWebSecurity
@EnableMethodSecurity
public class SecurityConfig {

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

    @Bean
    public PasswordEncoder passwordEncoder() {
        LOGGER.info("Creating BCryptPasswordEncoder");
        return new BCryptPasswordEncoder();
    }

    @Bean
    public UserDetailsService userDetailsService(PasswordEncoder passwordEncoder) {
        LOGGER.info("Creating in-memory users");
        InMemoryUserDetailsManager manager = new InMemoryUserDetailsManager();
        manager.createUser(User.withUsername("user").password(passwordEncoder.encode("pwd")).roles("USER").build());
        manager.createUser(User.withUsername("admin").password(passwordEncoder.encode("pwd")).roles("ADMIN").build());
        return manager;
    }

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.csrf(csrf -> csrf.disable())
                .authorizeHttpRequests(auth -> auth
                        .requestMatchers("/authenticate").permitAll()
                        .requestMatchers("/countries/**").hasRole("USER")
                        .anyRequest().authenticated()
                )
                .httpBasic();
        return http.build();
    }

    @Bean
    public AuthenticationManager authenticationManager(AuthenticationConfiguration
config) throws Exception {
        return config.getAuthenticationManager();
    }
}
```

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>spring-learn</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>spring-learn</name>
<description>Demo project for Spring Boot</description>

<properties>
    <java.version>21</java.version>
</properties>

<dependencies>

    <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>

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

    <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>

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

OUTPUT :



http://localhost:8083/authenticate

GET http://localhost:8083/authenticate

Auth Type: Basic Auth

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

Username: user

Password: pwd

200 OK • 369 ms • 475 B

```
{
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzUyMzA5NjU4LCJleHAiOjE3NTIzMTMyNTh9.
        ar6VQj8IZqPs0jD3_Dm-GKciSXJxvYI4KUOBQc1Tahs"
}
```