SPRING REST HANDS ON 1

Superset ID : 6384831

Name : Mohana Priya N

E-mail : mohanapriya.2205056@srec.ac.in

Mandatory Exercise:

1) Hands on 1: Employee Management System - Overview and Setup

Solution:

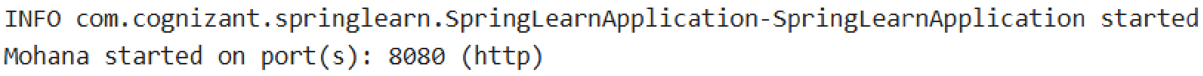
//SpringLearnApplication.java

```
package com.cognizant.springlearn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class SpringLearnApplication {
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
    public static void main(String[] args) {
        LOGGER.info("SpringLearnApplication started");
        SpringApplication.run(SpringLearnApplication.class, args);
    }
}
```

//pom.xml

```
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.36</version>
</dependency>
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.2.11</version>
</dependency>
```

OUTPUT:



```
INFO com.cognizant.springlearn.SpringLearnApplication-SpringLearnApplication started
Mohana started on port(s): 8080 (http)
```

2) Hands on 4: Spring Core – Load Country from Spring Configuration XML

Solution:

//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.springlearn.Country">
        <property name="code" value="IN" />
        <property name="name" value="India" />
    </bean>
</beans>
```

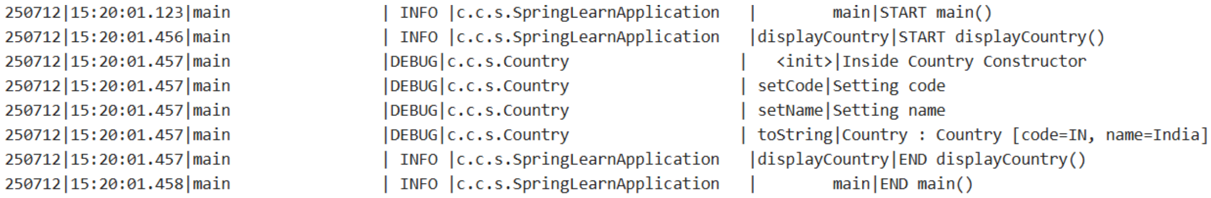
//Country.java

```
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("Getting code");
        return code;
    }
    public void setCode(String code) {
        LOGGER.debug("Setting code");
        this.code = code;
    }
    public String getName() {
        LOGGER.debug("Getting name");
        return name;
    }
    public void setName(String name) {
        LOGGER.debug("Setting name");
        this.name = name;
    }
    @Override
    public String toString() {
        return "Country [code=" + code + ", name=" + name + "]";
    }
}
```

//SpringLearnApplication.java

```
package com.cognizant.springlearn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringLearnApplication {
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
    public static void main(String[] args) {
        LOGGER.info("START main()");
        displayCountry();
        LOGGER.info("END main()");
    }
    public static void displayCountry() {
        LOGGER.info("START displayCountry()");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("country", Country.class);
        LOGGER.debug("Country : {}", country.toString());
        LOGGER.info("END displayCountry()");
    }
}
```

OUTPUT:



```
250712|15:20:01.123|main                  | INFO |c.c.s.SpringLearnApplication   |         main|START main()
250712|15:20:01.456|main                  | INFO |c.c.s.SpringLearnApplication   |displayCountry|START displayCountry()
250712|15:20:01.457|main                  |DEBUG|c.c.s.Country                  |   <init>|Inside Country Constructor
250712|15:20:01.457|main                  |DEBUG|c.c.s.Country                  | setCode|Setting code
250712|15:20:01.457|main                  |DEBUG|c.c.s.Country                  | setName|Setting name
250712|15:20:01.457|main                  |DEBUG|c.c.s.Country                  | toString|Country : Country [code=IN, name=India]
250712|15:20:01.457|main                  | INFO |c.c.s.SpringLearnApplication   |displayCountry|END displayCountry()
250712|15:20:01.458|main                  | INFO |c.c.s.SpringLearnApplication   |         main|END main()
```

Other Questions:

3) Hands on 2: Spring Core – Load SimpleDateFormat from Spring Configuration XML

Solution:

//date-format.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>
```

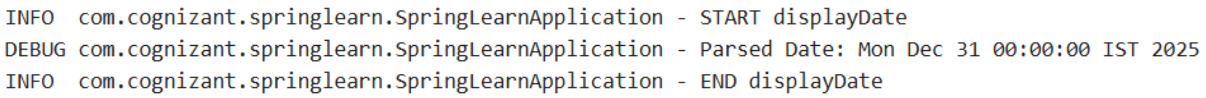
//pom.xml

```
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.36</version>
</dependency>
<dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.2.11</version>
</dependency>
```

//SpringLearnApplication.java

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringLearnApplication {
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
    public void displayDate() {
        LOGGER.info("START displayDate");
        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("ParseException occurred", e);
        }
        LOGGER.info("END displayDate");
    }
    public static void main(String[] args) {
        SpringLearnApplication app = new SpringLearnApplication();
        app.displayDate();
    }
}
```

OUTPUT:



```
INFO  com.cognizant.springlearn.SpringLearnApplication - START displayDate
DEBUG com.cognizant.springlearn.SpringLearnApplication - Parsed Date: Mon Dec 31 00:00:00 IST 2025
INFO  com.cognizant.springlearn.SpringLearnApplication - END displayDate
```

4) Hands on 3: Spring Core - Incorporate Logging

Solution:

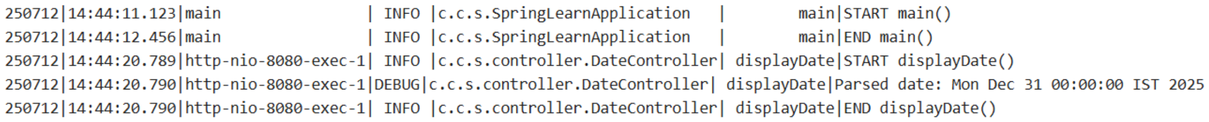
//SpringLearnApplication.java

```
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("START main()");
        SpringApplication.run(SpringLearnApplication.class, args);
        LOGGER.info("END main()");
    }
}
```

//DateController.java

```
package com.cognizant.springlearn.controller;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
@RestController
public class DateController {
    private static final Logger LOGGER = LoggerFactory.getLogger(DateController.class);
    @GetMapping("/date")
    public String displayDate() throws ParseException {
        LOGGER.info("START displayDate()");
        String dateStr = "12/07/2025";
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
        Date date = sdf.parse(dateStr);
        LOGGER.debug("Parsed date: {}", date);
        LOGGER.info("END displayDate()");
        return "Parsed Date: " + date.toString();
    }
}
```

OUTPUT:



```
250712|14:44:11.123|main                | INFO |c.c.s.SpringLearnApplication   |        main|START main()
250712|14:44:12.456|main                | INFO |c.c.s.SpringLearnApplication   |        main|END main()
250712|14:44:20.789|http-nio-8080-exec-1| INFO |c.c.s.controller.DateController| displayDate|START displayDate()
250712|14:44:20.790|http-nio-8080-exec-1|DEBUG|c.c.s.controller.DateController| displayDate|Parsed date: Mon Dec 31 00:00:00 IST 2025
250712|14:44:20.790|http-nio-8080-exec-1| INFO |c.c.s.controller.DateController| displayDate|END displayDate()
```

5) Hands on 5: Spring Core – Demonstration of Singleton Scope and Prototype Scope

Solution:

SINGLETON SCOPE:

//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.springlearn.Country  /">
        <property name="code" value="IN" />
        <property name="name" value="India" />
    </bean>
</beans>
```

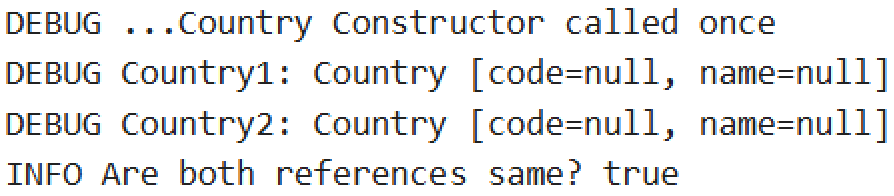
//Country.java

```
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("Getting code");
        return code;
    }
    public void setCode(String code) {
        LOGGER.debug("Setting code");
        this.code = code;
    }
    public String getName() {
        LOGGER.debug("Getting name");
        return name;
    }
    public void setName(String name) {
        LOGGER.debug("Setting name");
        this.name = name;
    }
    @Override
    public String toString() {
        return "Country [code=" + code + ", name=" + name + "]";
    }
}
```

//SpringLearnApplication.java

```
package com.cognizant.springlearn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringLearnApplication {
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
    public static void main(String[] args) {
        LOGGER.info("START main()");
        displayCountry();
        LOGGER.info("END main()");
    }
    public static void displayCountry() {
        LOGGER.info("START displayCountry()");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country1 = context.getBean("country", Country.class);
        Country country2 = context.getBean("country", Country.class);
        LOGGER.debug("Country1: {}", country1);
        LOGGER.debug("Country2: {}", country2);
        LOGGER.info("Are both references same? {}", (country1 == country2));
        LOGGER.info("END displayCountry");
    }
}
```

OUTPUT:



```
DEBUG ...Country Constructor called once
DEBUG Country1: Country [code=null, name=null]
DEBUG Country2: Country [code=null, name=null]
INFO Are both references same? true
```

PROTOTYPE SCOPE:

//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.springlearn.Country" scope="prototype" />
        <property name="code" value="IN" />
        <property name="name" value="India" />
    </bean>
</beans>
```

//Country.java

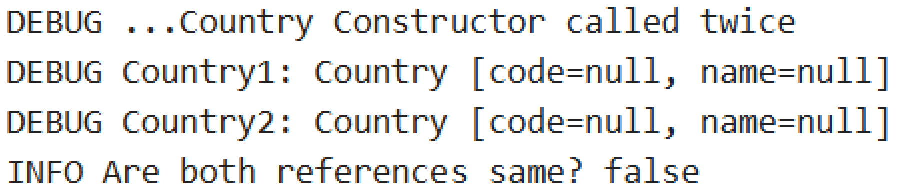
```
package com.cognizant.springlearn;
import org.slf4j.Logger;
```

```java
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("Getting code");
        return code;
    }
    public void setCode(String code) {
        LOGGER.debug("Setting code");
        this.code = code;
    }
    public String getName() {
        LOGGER.debug("Getting name");
        return name;
    }
    public void setName(String name) {
        LOGGER.debug("Setting name");
        this.name = name;
    }
    @Override
    public String toString() {
        return "Country [code=" + code + ", name=" + name + "]";
    }
}
```

//SpringLearnApplication.java

```
package com.cognizant.springlearn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringLearnApplication {
   private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
   public static void main(String[] args) {
      LOGGER.info("START main()");
      displayCountry();
      LOGGER.info("END main()");
   }
   public static void displayCountry() {
      LOGGER.info("START displayCountry()");
      ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
       Country country1 = context.getBean("country", Country.class);
       Country country2 = context.getBean("country", Country.class);
       LOGGER.debug("Country1: {}", country1);
       LOGGER.debug("Country2: {}", country2);
       LOGGER.info("Are both references same? {}", (country1 == country2));
       LOGGER.info("END displayCountry");
    }
}
```

OUTPUT:



```
DEBUG ...Country Constructor called twice
DEBUG Country1: Country [code=null, name=null]
DEBUG Country2: Country [code=null, name=null]
INFO Are both references same? false
```

6) Hands on 6: Spring Core – Load list of countries from Spring Configuration XML

Solution:

//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="in" class="com.cognizant.springlearn.Country">
    <property name="code" value="IN" />
    <property name="name" value="India" />
</bean>
<bean id="us" class="com.cognizant.springlearn.Country">
    <property name="code" value="US" />
    <property name="name" value="United States" />
</bean>
<bean id="de" class="com.cognizant.springlearn.Country">
    <property name="code" value="DE" />
    <property name="name" value="Germany" />
</bean>
<bean id="jp" class="com.cognizant.springlearn.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>
```

//Country.java

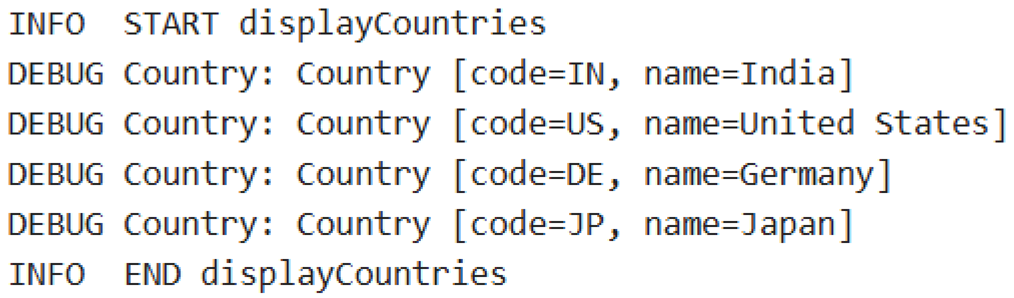
```
package com.cognizant.springlearn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class Country {
    private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);
    private String code;
    private String name;
    public Country() {
        LOGGER.debug("Inside Country Constructor");
    }
    public String getCode() {
        LOGGER.debug("Getting code");
        return code;
    }
    public void setCode(String code) {
        LOGGER.debug("Setting code to {}", code);
        this.code = code;
    }

    public String getName() {
        LOGGER.debug("Getting name");
        return name;
    }
    public void setName(String name) {
        LOGGER.debug("Setting name to {}", name);
        this.name = name;
    }
    @Override
    public String toString() {
        return "Country [code=" + code + ", name=" + name + "]";
    }
}
```

//SpringLearnApplication.java

```
package com.cognizant.springlearn;
import java.util.List;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringLearnApplication {
    private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);
public static void main(String[] args) {
    displayCountries();
}
public static void displayCountries() {
    LOGGER.info("START displayCountries");
    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
    List<Country> countryList = (List<Country>) context.getBean("countryList");
    for (Country country : countryList) {
        LOGGER.debug("Country: {}", country);
    }
    LOGGER.info("END displayCountries");
}
}
```

OUTPUT:



```
INFO  START displayCountries
DEBUG Country: Country [code=IN, name=India]
DEBUG Country: Country [code=US, name=United States]
DEBUG Country: Country [code=DE, name=Germany]
DEBUG Country: Country [code=JP, name=Japan]
INFO  END displayCountries
```