

# 1.Spring \_REST Handson

## Hands on 1

### Create a Spring Web Project using Maven

The screenshot shows the Spring Initializr web application interface. At the top, it says "start.spring.io". The main heading is "spring initializr". Below the heading, there are two sections: "Project" and "Language". Under "Project", there are two radio buttons: "Gradle - Groovy" and "Gradle - Kotlin". Under "Language", there are three radio buttons: "Java" (selected), "Kotlin", and "Groovy". In the center, there is a section for "Spring Boot" with three radio buttons: "4.0.0 (SNAPSHOT)", "3.5.4 (SNAPSHOT)" (selected), and "3.4.8 (SNAPSHOT)". Below this, there is a section for "Project Metadata" with three fields: "Group" (com.cognizant), "Artifact" (Spring\_learn), and "Name" (Spring\_learn). At the bottom of the page, there is a "Build" button.

This screenshot shows the continuation of the Spring Initializr configuration. It includes fields for "Description" (Demo project for Spring Boot), "Package name" (com.cognizant.Spring\_learn), and "Packaging" (Jar, selected over War). It also shows Java version options: 24, 21, and 17, with 17 selected. The background is dark, and the text is white.

## Dependencies

**ADD DEPENDENCIES... CTRL + B**

---

**Spring Web** WEB

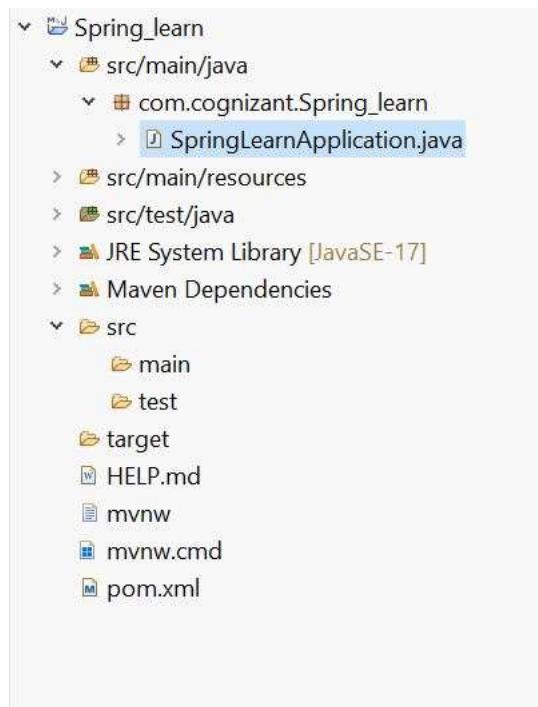
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

---

**Spring Boot DevTools** DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

## Import in Eclipse



## Downloading packages

```
C:\Users\ADMIN\ode C:\Users\ADMIN\eclipse-workspace\Spring_learn
C:\Users\ADMIN\eclipse-workspace\Spring_learn\mvn clean package -DskipTests
[INFO] Scanning for projects...
[INFO]
[INFO] -----> [ com.cognizant:Spring_learn ] -----
[INFO] Building Spring_learn 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----{ jar }-----
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.4.1/maven-clean-plugin-3.4.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.4.1/maven-clean-plugin-3.4.1.jar (3.0 kB at 7.2 kB/s)
[INFO]
[INFO] --- clean:clean (default-clean) @ Spring_learn ---
[INFO] Deleting C:\Users\ADMIN\eclipse-workspace\Spring_learn\target
[INFO]
[INFO] --- resources:resources (default-resources) @ Spring_learn ---
[INFO] Copying 1 Resource From src/main/resources to target\classes
[INFO] Copying 0 resource from src/main/resources to target\classes
[INFO]
[INFO] --- compiler:3.14.0:compile (default-compile) @ Spring_learn ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target\classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ Spring_learn ---
[INFO] skip non existing resourceDirectory C:\Users\ADMIN\eclipse-workspace\Spring_learn\src\test\resources
[INFO]
[INFO] --- compiler:3.14.0:testCompile (default-testCompile) @ Spring_learn ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target\test\classes
[INFO]
[INFO] --- surefire:3.3.3:test (default-test) @ Spring_learn ---
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-api/3.3.3/surefire-api-3.3.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-api/3.3.3/surefire-api-3.3.3.jar (3.7 kB at 899 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-reReporter/3.3.3/surefire-reReporter-3.3.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-reporter/3.3.3/surefire-reporter-3.3.3.pom (3.0 kB at 2.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-shared-utils/3.3.3/surefire-shared-utils-3.3.3.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-shared-utils/3.3.3/surefire-shared-utils-3.3.3.jar (4.0 kB at 3.2 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-extensions/3.3.3/surefire-extensions-3.3.3.pom
```

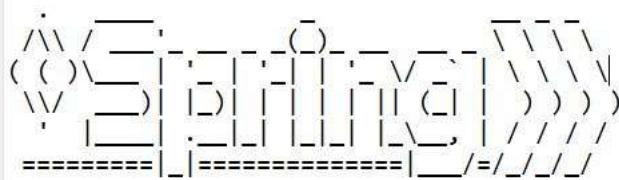
```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.27.1/commons-compress-1.27.1.jar (1.1 kB)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/httpcomponents/core5/httpcore5-h2/5.3.4/httpcore5-h2-5.3.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/httpcomponents/core5/httpcore5-h2/5.3.4/httpcore5-h2-5.3.4.jar (242 kB)
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-loader-tools/3.5.3/spring-boot-loader-tools-3.5.3.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/httpcomponents/client/httpclient5/5.5/httpclient5-5.5.jar (355 kB)
Downloaded from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-observation/1.14.8/micrometer-observation-1.14.8.jar
Downloaded from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-observation/1.14.8/micrometer-observation-1.14.8.jar (1.1 kB)
Downloaded from central: https://repo.maven.apache.org/maven2/org/micrometer/micrometer-commons/1.14.8/micrometer-commons-1.14.8.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-loader-tools/3.5.3/spring-boot-loader-tools-3.5.3.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-shade-plugin/3.6.0/maven-shade-plugin-3.6.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/io/micrometer/micrometer-commons/1.14.8/micrometer-commons-1.14.8.jar (48 kB)
Downloaded from central: https://repo.maven.apache.org/maven2/org/owl/asn/asn/9.7/asn-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-shade-plugin/3.6.0/maven-shade-plugin-3.6.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/owl/asn/asn-commons/9.7/asn-commons-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/owl/asn/asn/9.7/asn-9.7.jar (125 kB at 15 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/owl/asn/asn-tree/9.7/asn-tree-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/net/java/dev/jna/jna-platform/5.17.0/jna-platform-5.17.0.jar (1.4 kB at 165 kB)
Downloaded from central: https://repo.maven.apache.org/maven2/org/vafer/jdependency/2.18/jdependency-2.18.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/owl/asn/asn-tree/9.7/asn-tree-9.7.jar (62 kB at 6.2 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/owl/asn/asn-commons/9.7/asn-commons-9.7.jar (73 kB at 8.6 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/vafer/jdependency/2.18/jdependency-2.18.jar (416 kB at 48 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/net/java/dev/jna/5.17.0/jna-5.17.0.jar (2.0 kB at 215 kB/s)
[INFO] Replacing main artifact C:\Users\ADMIN\eclipse-workspace\Spring_learn\target\Spring_learn-0.0.1-SNAPSHOT.jar with repackaged archive, OOT-INF/.
[INFO] The original artifact has been renamed to C:\Users\ADMIN\eclipse-workspace\Spring_learn\target\Spring_learn-0.0.1-SNAPSHOT.jar.origins
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 81:09 min
[INFO] Finished at: 2025-07-11T19:34:10+05:30
[INFO]
```

```

SpringLearnApplication.java ×
1 package com.cognizant.Spring_learn;
2
3+import org.springframework.boot.SpringApplication;□
5
6 @SpringBootApplication
7 public class SpringLearnApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(SpringLearnApplication.class, args);
11     }
12
13 }
14

```

## OUTPUT:



:: Spring Boot :: (v3.5.3)

```

2025-07-11T19:34:56.418+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : Starting SpringLearnApplication using Java 21.0.
2025-07-11T19:34:56.423+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : No active profile set, falling back to 1 default
2025-07-11T19:34:56.520+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : DevTools property defaults active! Set 'spring.devtools.restart.enabled' to false for additional web related logging consider setting 'spring.devtools.restart.enabled=false'
2025-07-11T19:34:56.521+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : Tomcat initialized with port 8080 (http)
2025-07-11T19:34:58.586+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : Tomcat initialized with port 8080 (http)
2025-07-11T19:34:58.587+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : Starting service [Tomcat]
2025-07-11T19:34:58.587+05:30 INFO 12236 --- [Spring_learn] o.apache.catalina.core.StandardService : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-11T19:34:58.642+05:30 INFO 12236 --- [Spring_learn] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-07-11T19:34:58.645+05:30 INFO 12236 --- [Spring_learn] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 4 ms
2025-07-11T19:34:59.587+05:30 INFO 12236 --- [Spring_learn] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2025-07-11T19:34:59.689+05:30 INFO 12236 --- [Spring_learn] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''
2025-07-11T19:34:59.712+05:30 INFO 12236 --- [Spring_learn] c.c.Spring_learn.SpringLearnApplication : Started SpringLearnApplication in 4.138 seconds

```

## Hands on 2

### Spring Core – Load SimpleDateFormat from Spring Configuration XML

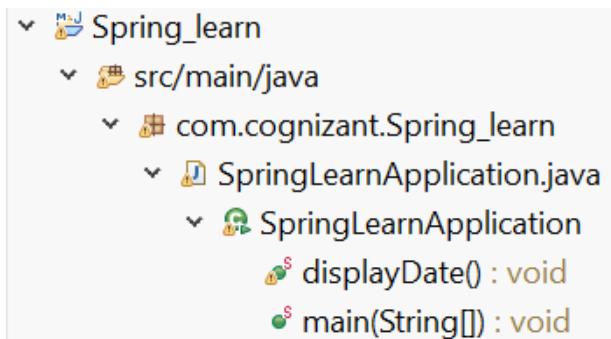


date-format.xml

```
<?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
                           http://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.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

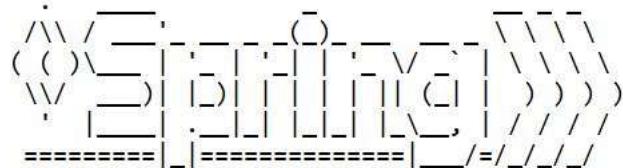
import java.text.SimpleDateFormat;
import java.text.ParseException;
import java.util.Date;

@SpringBootApplication
public class SpringLearnApplication {

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

    public static void displayDate() {
        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);
        try {
            Date date = format.parse("31/12/2018");
            System.out.println(date);
        } catch (ParseException e) {
            e.printStackTrace();
        }
    }
}
```

## OUTPUT:



:: Spring Boot :: (v3.5.3)

```
2025-07-11T20:28:25.650+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] c.c.Spring_learn.SpringLearnApplication : Starting SpringLearnAp
2025-07-11T20:28:25.651+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] c.c.Spring_learn.SpringLearnApplication : No active profile set,
2025-07-11T20:28:25.890+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with
2025-07-11T20:28:25.891+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomc
2025-07-11T20:28:25.892+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engin
2025-07-11T20:28:25.915+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring em
2025-07-11T20:28:25.915+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationCon
2025-07-11T20:28:25.987+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is r
2025-07-11T20:28:25.995+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port
2025-07-11T20:28:25.999+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] c.c.Spring_learn.SpringLearnApplication : Started SpringLearnApp
2025-07-11T20:28:26.001+05:30 INFO 12236 --- [Spring_learn] [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation u
```

Mon Dec 31 00:00:00 IST 2018

## Hands on 3

### Spring Core - Incorporate Logging

application.properties

```
1spring.application.name=Spring_learn
2server.port=8081
3# Set log levels
4logging.level.org.springframework=INFO
5logging.level.com.cognizant=DEBUG
6
7# Customize log format
8logging.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.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

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

@SpringBootApplication
public class SpringLearnApplication {

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

    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
        displayDate(); // Call the method with logging
    }

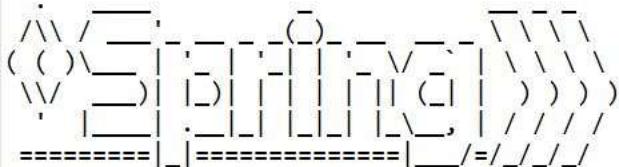
    public static void displayDate() {
        LOGGER.info("START");

        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);

            try {
                Date date = format.parse("31/12/2018");
                LOGGER.debug("Parsed date: {}", date);
            } catch (ParseException e) {
                LOGGER.error("Date parsing failed", e);
            }
        }
        LOGGER.info("END");
    }
}
```

## OUTPUT:



:: Spring Boot :: (v3.5.3)

```
250711|20:45:18.993|restartedMain | INFO|.S.SpringLearnApplication| logStarting|Starting SpringLearnApplication usin
250711|20:45:18.993|restartedMain | DEBUG|.S.SpringLearnApplication| logStarting|Running with Spring Boot v3.5.3, Spr
250711|20:45:18.994|restartedMain | INFO|.S.SpringLearnApplication| logStartupProfileInfo|No active profile set, falling back
250711|20:45:19.285|restartedMain | INFO|.s.b.w.e.t.TomcatWebServer| initialize|Tomcat initialized with port 8081 (h
250711|20:45:19.287|restartedMain | INFO|o.a.c.c.StandardService| log|Starting service [Tomcat]
250711|20:45:19.288|restartedMain | INFO|o.a.c.core.StandardEngine| log|Starting Servlet engine: [Apache Tom
250711|20:45:19.289|restartedMain | INFO|o.a.c.c.C.[.[]]| log|Initializing Spring embedded WebApp
250711|20:45:19.298|restartedMain | INFO|bServerApplicationContext| prepareWebApplicationContext|Root WebApplicationContext: initi
250711|20:45:19.299|restartedMain | INFO|OptionalLiveReloadServer| startServer|LiveReload server is running on port
250711|20:45:19.311|restartedMain | INFO|.s.b.w.e.t.TomcatWebServer| start|Tomcat started on port 8081 (http) w
250711|20:45:19.315|restartedMain | INFO|.S.SpringLearnApplication| logStarted|Started SpringLearnApplication in 0,
250711|20:45:19.319|restartedMain | INFO|ationDeltaLoggingListener| onApplicationEvent|Condition evaluation unchanged
250711|20:45:19.320|restartedMain | INFO|.S.SpringLearnApplication| displayDate|START
250711|20:45:19.343|restartedMain | DEBUG|.S.SpringLearnApplication| displayDate|Parsed date: Mon Dec 31 00:00:00 IST
250711|20:45:19.345|restartedMain | INFO|.S.SpringLearnApplication| displayDate|END
```

```
logStarting|Starting SpringLearnApplication using Java 21.0.7 with PID 12236 (C:\Users\ADMIN\eclipse-workspace\Spring
logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
initialize|Tomcat initialized with port 8081 (http)
    log|Starting service [Tomcat]
    log|Starting Servlet engine: [Apache Tomcat/10.1.42]
    log|Initializing Spring embedded WebApplicationContext
prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 231 ms
    startServer|LiveReload server is running on port 35729
        start|Tomcat started on port 8081 (http) with context path '/'
    logStarted|Started SpringLearnApplication in 0.352 seconds (process running for 4224.575)
onApplicationEvent|Condition evaluation unchanged
    displayDate|START
    displayDate|Parsed date: Mon Dec 31 00:00:00 IST 2018
    displayDate|END
```

## Hands on 4

### Spring Core – Load Country from Spring Configuration XML

Country.java

```
package com.cognizant.Spring_learn;

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 + "]";
    }
}
```

## 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
                           http://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:

```
logStarting|Starting SpringLearnApplication using Java 21.0.7 with PID 12236 (C:\Users\ADMIN\)
logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8
logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
    initialize|Tomcat initialized with port 8085 (http)
        log|Starting service [Tomcat]
        log|Starting Servlet engine: [Apache Tomcat/10.1.42]
        log|Initializing Spring embedded WebApplicationContext
prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 282 ms
    startServer|LiveReload server is running on port 35729
        start|Tomcat started on port 8085 (http) with context path '/'
    logStarted|Started SpringLearnApplication in 0.389 seconds (process running for 8804.782)
onApplicationEvent|Condition evaluation unchanged
    displayDate|START
    displayDate|Parsed date: Mon Dec 31 00:00:00 IST 2018
    displayDate|END
    displayCountry|START
        <init>|Inside Country Constructor
        setCode|Setting code
        setName|Setting name
    displayCountry|Country : Country [code=IN, name=India]
    displayCountry|END
```

## Hands on 5

### Spring Core – Demonstration of Singleton Scope and Prototype Scope

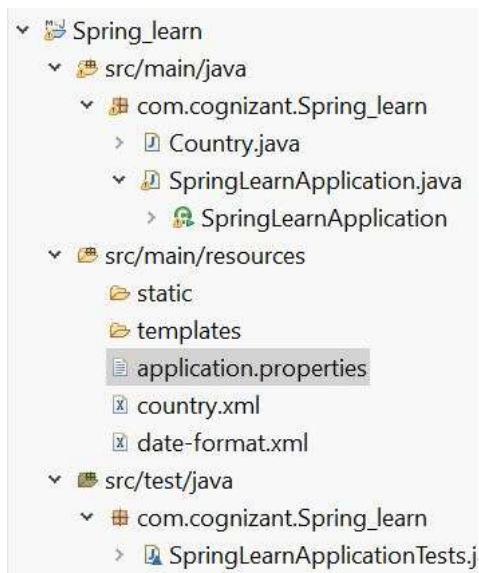
#### Country.xml

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

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

    <bean id="anotherCountry" class="com.cognizant.Spring_learn.Country" scope="prototype">
        <property name="code" value="US"/>
        <property name="name" value="United States"/>
    </bean>

</beans>
```



#### Application properties

```
spring.application.name=Spring_learn
server.port=8085
# Set log levels
logging.level.org.springframework=INFO
logging.level.com.cognizant=DEBUG
```

## SpringLearnApplication.java

```
package com.cognizant.Spring_learn;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

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

@SpringBootApplication
public class SpringLearnApplication {

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

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

    public static 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("Date parsing failed", e);
        }

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

```

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

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

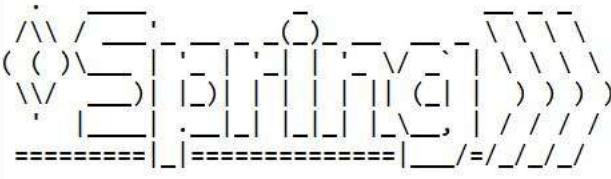
            Country country1 = context.getBean("country", Country.class);
            Country country2 = context.getBean("country", Country.class);
            LOGGER.debug("country1 == country2 : {}", country1 == country2);

            Country anotherCountry1 = context.getBean("anotherCountry", Country.class);
            Country anotherCountry2 = context.getBean("anotherCountry", Country.class);
            LOGGER.debug("anotherCountry1 == anotherCountry2 : {}", anotherCountry1 == anotherCountry2);

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

```

## OUTPUT:



:: Spring Boot :: (v3.5.3)

250711 22:12:46.969 restartedMain	INFO S.SpringLearnApplication
250711 22:12:46.981 restartedMain	DEBUG S.SpringLearnApplication
250711 22:12:46.983 restartedMain	INFO S.SpringLearnApplication
250711 22:12:47.139 restartedMain	INFO ertyDefaultsPostProcessor
250711 22:12:47.140 restartedMain	INFO ertyDefaultsPostProcessor
250711 22:12:49.265 restartedMain	INFO s.b.w.e.t.TomcatWebServer
250711 22:12:49.295 restartedMain	INFO o.a.c.c.StandardService
250711 22:12:49.296 restartedMain	INFO o.a.c.core.StandardEngine
250711 22:12:49.379 restartedMain	INFO o.a.c.c.C.[.[]]
250711 22:12:49.381 restartedMain	INFO bServerApplicationContext
250711 22:12:49.990 restartedMain	INFO .OptionalLiveReloadServer
250711 22:12:50.040 restartedMain	INFO s.b.w.e.t.TomcatWebServer
250711 22:12:50.058 restartedMain	INFO S.SpringLearnApplication
250711 22:12:50.068 restartedMain	INFO S.SpringLearnApplication
250711 22:12:50.244 restartedMain	DEBUG S.SpringLearnApplication
250711 22:12:50.246 restartedMain	INFO S.SpringLearnApplication
250711 22:12:50.247 restartedMain	INFO S.SpringLearnApplication
250711 22:12:50.287 restartedMain	DEBUG c.c.Spring_learn.Country
250711 22:12:50.288 restartedMain	DEBUG c.c.Spring_learn.Country
250711 22:12:50.289 restartedMain	DEBUG c.c.Spring_learn.Country

250711	22:12:49.265	restartedMain	INFO s.b.w.e.t.TomcatWebServer
250711	22:12:49.295	restartedMain	INFO o.a.c.c.StandardService
250711	22:12:49.296	restartedMain	INFO o.a.c.core.StandardEngine
250711	22:12:49.379	restartedMain	INFO o.a.c.c.C.[.[]]
250711	22:12:49.381	restartedMain	INFO bServerApplicationContext
250711	22:12:49.990	restartedMain	INFO .OptionalLiveReloadServer
250711	22:12:50.040	restartedMain	INFO s.b.w.e.t.TomcatWebServer
250711	22:12:50.058	restartedMain	INFO .S.SpringLearnApplication
250711	22:12:50.068	restartedMain	INFO .S.SpringLearnApplication
250711	22:12:50.244	restartedMain	DEBUG .S.SpringLearnApplication
250711	22:12:50.246	restartedMain	INFO .S.SpringLearnApplication
250711	22:12:50.247	restartedMain	INFO .S.SpringLearnApplication
250711	22:12:50.287	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.288	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.289	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.292	restartedMain	DEBUG .S.SpringLearnApplication
250711	22:12:50.294	restartedMain	INFO .S.SpringLearnApplication
250711	22:12:50.295	restartedMain	INFO .S.SpringLearnApplication
250711	22:12:50.337	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.341	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.345	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.349	restartedMain	DEBUG .S.SpringLearnApplication
250711	22:12:50.351	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.353	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.361	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.362	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.363	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.364	restartedMain	DEBUG c.c.Spring_learn.Country
250711	22:12:50.365	restartedMain	DEBUG .S.SpringLearnApplication
250711	22:12:50.365	restartedMain	INFO .S.SpringLearnApplication

```

    <init>|Inside Country Constructor
    setCode|Setting code
    setName|Setting name
displayScope|country1 == country2 : true
    <init>|Inside Country Constructor
    setCode|Setting code
    setName|Setting name
    <init>|Inside Country Constructor
    setCode|Setting code
    setName|Setting name
displayScope|anotherCountry1 == anotherCountry2 : false
displayScope|END

```

## Hands on 6

### Spring Core – Load list of countries from Spring Configuration XML

SpringLearnApplication.java

```
package com.cognizant.Spring_learn;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.text.SimpleDateFormat;
import java.text.ParseException;
import java.util.Date;
import java.util.List;
import java.util.*;
@SpringBootApplication
public class SpringLearnApplication {

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

    public static void main(String[] args) {
        SpringApplication.run(SpringLearnApplication.class, args);
        displayDate();
        displayCountry();
        //displayScope();
        displayCountries();
    }

    public static 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("Date parsing failed", e);
        }

        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);

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

```

public static void displayScope() {
    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("country1 == country2 : {}", country1 == country2);

    Country anotherCountry1 = context.getBean("anotherCountry", Country.class);
    Country anotherCountry2 = context.getBean("anotherCountry", Country.class);
    LOGGER.debug("anotherCountry1 == anotherCountry2 : {}", anotherCountry1 == anotherCountry2);

    LOGGER.info("END");
}

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

    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
    List<Country> countries = context.getBean("countryList", List.class);

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

    LOGGER.info("END");
}

}

```

## Country.xml

---

```

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

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

    <bean id="anotherCountry" class="com.cognizant.Spring_learn.Country" scope="prototype">
        <property name="code" value="AU"/>
        <property name="name" value="Australia"/>
    </bean>

```

```

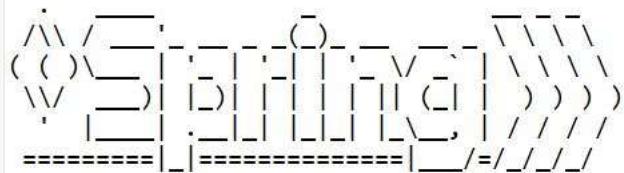
<bean id="country2" class="com.cognizant.Spring_learn.Country">
    <property name="code" value="US"/>
    <property name="name" value="United States"/>
</bean>

<bean id="countryList" class="java.util.ArrayList">
    <constructor-arg>
        <list>
            <ref bean="country"/>
            <ref bean="country2"/>
        </list>
    </constructor-arg>
</bean>

</beans>

```

## OUTPUT:



:: Spring Boot :: (v3.5.3)

250712 12:57:23.811 restartedMain	INFO .S.SpringLearnApplication
250712 12:57:23.815 restartedMain	DEBUG .S.SpringLearnApplication
250712 12:57:23.815 restartedMain	INFO .S.SpringLearnApplication
250712 12:57:24.257 restartedMain	INFO s.b.w.e.t.TomcatWebServer
250712 12:57:24.259 restartedMain	INFO o.a.c.c.StandardService
250712 12:57:24.260 restartedMain	INFO o.a.c.core.StandardEngine
250712 12:57:24.288 restartedMain	INFO o.a.c.c.C.[.[]]
250712 12:57:24.288 restartedMain	INFO bServerApplicationContext
250712 12:57:24.721 restartedMain	INFO .OptionalLiveReloadServer
250712 12:57:24.739 restartedMain	INFO s.b.w.e.t.TomcatWebServer
250712 12:57:24.747 restartedMain	INFO .S.SpringLearnApplication
250712 12:57:24.752 restartedMain	INFO ationDeltaLoggingListener
250712 12:57:24.753 restartedMain	INFO .S.SpringLearnApplication
250712 12:57:24.780 restartedMain	DEBUG .S.SpringLearnApplication
250712 12:57:24.784 restartedMain	INFO .S.SpringLearnApplication
250712 12:57:24.786 restartedMain	INFO .S.SpringLearnApplication

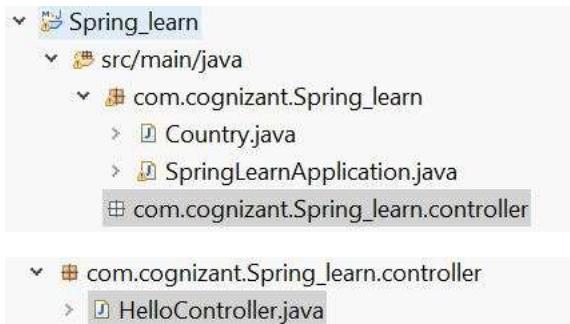
```
250712|12:57:24.752|restartedMain | INFO|ationDeltaLoggingListener
250712|12:57:24.753|restartedMain | INFO|S.SpringLearnApplication
250712|12:57:24.780|restartedMain | DEBUG|S.SpringLearnApplication
250712|12:57:24.784|restartedMain | INFO|S.SpringLearnApplication
250712|12:57:24.786|restartedMain | INFO|S.SpringLearnApplication
250712|12:57:25.014|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.025|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.026|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.028|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.029|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.029|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.037|restartedMain | DEBUG|S.SpringLearnApplication
250712|12:57:25.038|restartedMain | INFO|S.SpringLearnApplication
250712|12:57:25.041|restartedMain | INFO|S.SpringLearnApplication
250712|12:57:25.095|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.099|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.100|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.110|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.113|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.114|restartedMain | DEBUG|c.c.Spring_learn.Country
250712|12:57:25.119|restartedMain | DEBUG|S.SpringLearnApplication
250712|12:57:25.119|restartedMain | DEBUG|S.SpringLearnApplication
250712|12:57:25.119|restartedMain | INFO|S.SpringLearnApplication

onApplicationEvent|Condition evaluation unchanged
    displayDate|START
    displayDate|Parsed date: Mon Dec 31 00:00:00 IST 2018
    displayDate|END
    displayCountry|START
        <init>|Inside Country Constructor
        setCode|Setting code
        setName|Setting name
        <init>|Inside Country Constructor
        setCode|Setting code
        setName|Setting name
    displayCountry|Country : Country [code=IN, name=India]
    displayCountry|END
    displayCountries|START
        <init>|Inside Country Constructor
        setCode|Setting code
        setName|Setting name
        <init>|Inside Country Constructor
        setCode|Setting code
        setName|Setting name
    displayCountries|Country: Country [code=IN, name=India]
    displayCountries|Country: Country [code=US, name=United States]
    displayCountries|END
```

## 2.Spring \_REST Handson

### Hello World RESTful Web Service Hello World RESTful Web Service

Create new package



#### 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 {

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

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

```

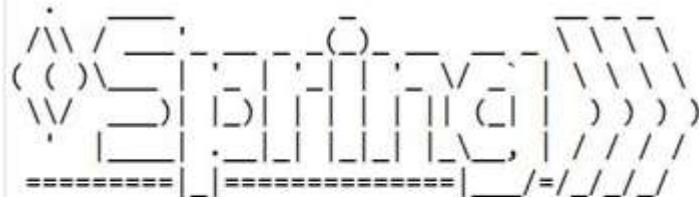
package com.cognizant.Spring_learn;

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

@SpringBootApplication
@ComponentScan(basePackages = "com.cognizant.Spring_learn")
public class SpringLearnApplication {

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

```



:: Spring Boot :: (v3.5.3)

250712 14:04:38.311 restartedMain	INFO .S.SpringLearnApplication
250712 14:04:38.313 restartedMain	INFO .S.SpringLearnApplication
250712 14:04:38.586 restartedMain	INFO s.b.w.e.t.TomcatWebServer
250712 14:04:38.587 restartedMain	INFO o.a.c.c.StandardService
250712 14:04:38.588 restartedMain	INFO o.a.c.core.StandardEngine
250712 14:04:38.607 restartedMain	INFO o.a.c.c.C.[.[]/]
250712 14:04:38.608 restartedMain	INFO bServerApplicationContext
250712 14:04:38.700 restartedMain	INFO .OptionalLiveReloadServer
250712 14:04:38.718 restartedMain	INFO s.b.w.e.t.TomcatWebServer
250712 14:04:38.723 restartedMain	INFO .S.SpringLearnApplication
250712 14:04:38.729 restartedMain	INFO ationDeltaLoggingListener
250712 14:05:56.182 http-nio-8080-exec-2	INFO o.a.c.c.C.[.[]/]
250712 14:05:56.186 http-nio-8080-exec-2	INFO o.s.w.s.DispatcherServlet
250712 14:05:56.193 http-nio-8080-exec-2	INFO o.s.w.s.DispatcherServlet

← → ⌂ localhost:8080/hello

Hello World!!

```
logStarting|Starting SpringLearnApplication using Java 21.0.7 with PID 4860 (C:\Users\ADMIN
logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
    initialize|Tomcat initialized with port 8080 (http)
        log|Starting service [Tomcat]
        log|Starting Servlet engine: [Apache Tomcat/10.1.42]
        log|Initializing Spring embedded WebApplicationContext
prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 285 ms
    startServer|LiveReload server is running on port 35729
        start|Tomcat started on port 8080 (http) with context path '/'
logStarted|Started SpringLearnApplication in 0.475 seconds (process running for 4225.917)
onApplicationEvent|Condition evaluation unchanged
    log|Initializing Spring DispatcherServlet 'dispatcherServlet'
initServletBean|Initializing Servlet 'dispatcherServlet'
initServletBean|Completed initialization in 3 ms
```

## OUTPUT:

```
sayHello|START - sayHello()
sayHello|END - sayHello()
```

## REST - Country Web Service

Country.java

```
package com.cognizant.Spring_learn;

public class Country {
    private String code;
    private String name;

    public Country() {
    }

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getName() {
        return name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

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

## 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.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class CountryController {

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

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

        Country country = new Country();
        country.setCode("IN");
        country.setName("India");

        LOGGER.info("END - getCountryIndia()");
        return country;
    }
}
```



:: Spring Boot :: (v3.5.3)

250712 14:12:32.948 restartedMain	INFO  .S.SpringLearnApplication
250712 14:12:32.948 restartedMain	INFO  .S.SpringLearnApplication
250712 14:12:33.165 restartedMain	INFO  s.b.w.e.t.TomcatWebServer
250712 14:12:33.166 restartedMain	INFO  o.a.c.c.StandardService
250712 14:12:33.167 restartedMain	INFO  o.a.c.core.StandardEngine
250712 14:12:33.191 restartedMain	INFO  o.a.c.c.C.[..[.]]
250712 14:12:33.193 restartedMain	INFO  bServerApplicationContext
250712 14:12:33.285 restartedMain	INFO  .OptionalLiveReloadServer
250712 14:12:33.296 restartedMain	INFO  s.b.w.e.t.TomcatWebServer
250712 14:12:33.300 restartedMain	INFO  .S.SpringLearnApplication
250712 14:12:33.303 restartedMain	INFO  ationDeltaLoggingListener
250712 14:13:33.761 http-nio-8080-exec-1	INFO  o.a.c.c.C.[..[.]]
250712 14:13:33.762 http-nio-8080-exec-1	INFO  o.s.w.s.DispatcherServlet
250712 14:13:33.762 http-nio-8080-exec-1	INFO  o.s.w.s.DispatcherServlet
250712 14:13:33.767 http-nio-8080-exec-1	INFO  c.c.S.c.CountryController
250712 14:13:33.768 http-nio-8080-exec-1	INFO  c.c.S.c.CountryController

localhost:8080/country

Pretty-print □

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

```
logStarting|Starting SpringLearnApplication using Java 21.0.7 with PID 4860 (C:\Users\ADMINI
logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"
    initialize|Tomcat initialized with port 8080 (http)
        log|Starting service [Tomcat]
        log|Starting Servlet engine: [Apache Tomcat/10.1.42]
        log|Initializng Spring embedded WebApplicationContext
prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 242 ms
    startServer|LiveReload server is running on port 35729
        start|Tomcat started on port 8080 (http) with context path '/'
    logStarted|Started SpringLearnApplication in 0.378 seconds (process running for 4700.493)
onApplicationEvent|Condition evaluation unchanged
    log|Initializng Spring DispatcherServlet 'dispatcherServlet'
initServletBean|Initializng Servlet 'dispatcherServlet'
```

## OUTPUT:

```
initServletBean|Completed initialization in 0 ms
getCountryIndia|START - getCountryIndia()
getCountryIndia|END - getCountryIndia()
```

## REST - Get country based on country code

### Country.xml

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

    <bean id="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="cn" class="com.cognizant.Spring_learn.Country">
        <property name="code" value="CN"/>
        <property name="name" value="China"/>
    </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.PathVariable;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class CountryController {
```

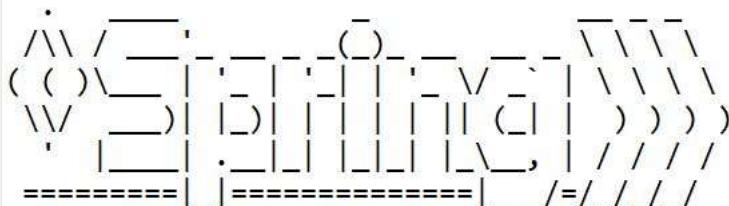
```

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

@GetMapping("/country")
public Country getCountryIndia() {
    LOGGER.info("START - getCountryIndia()");
    Country country = new Country();
    country.setCode("IN");
    country.setName("India");
    LOGGER.info("END - getCountryIndia()");
    return country;
}

@GetMapping("/countries/{code}")
public Country getCountry(@PathVariable String code) {
    LOGGER.info("START - getCountry() for code: {}", code);
    ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
    Country country = (Country) context.getBean(code.toLowerCase());
    LOGGER.info("END - getCountry()");
    return country;
}
}

```



:: Spring Boot :: (v3.5.3)

```

2025-07-12T14:31:22.267+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.SpringApplication
2025-07-12T14:31:22.275+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.SpringApplication
2025-07-12T14:31:22.489+05:30 INFO 12268 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor
2025-07-12T14:31:22.489+05:30 INFO 12268 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor
2025-07-12T14:31:24.959+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-12T14:31:25.000+05:30 INFO 12268 --- [ restartedMain] o.apache.catalina.core.StandardService
2025-07-12T14:31:25.000+05:30 INFO 12268 --- [ restartedMain] o.apache.catalina.core.StandardEngine
2025-07-12T14:31:25.076+05:30 INFO 12268 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-12T14:31:25.081+05:30 INFO 12268 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext
2025-07-12T14:31:25.866+05:30 INFO 12268 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer
2025-07-12T14:31:25.955+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-12T14:31:25.978+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.SpringApplication

```

```
localhost:8080/countries/in
Pretty-print □
{"code":"IN","name":"India"}
```

```
localhost:8080/countries/us
Pretty-print □
{"code":"US","name":"United States"}
```

```
Starting SpringLearnApplication using Java 21.0.7 with PID 12268 (C:\Users\ADMIN\ec)
No active profile set, falling back to 1 default profile: "default"
Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to
For additional web related logging consider setting the 'logging.level.web' property
Tomcat initialized with port 8080 (http)
Starting service [Tomcat]
Starting Servlet engine: [Apache Tomcat/10.1.42]
Initializing Spring embedded WebApplicationContext
Root WebApplicationContext: initialization completed in 2586 ms
LiveReload server is running on port 35729
Tomcat started on port 8080 (http) with context path '/'
Started SpringLearnApplication in 4.493 seconds (process running for 5.189)
Initializing Spring DispatcherServlet 'dispatcherServlet'
Initializing Servlet 'dispatcherServlet'
```

## OUTPUT:

```
Completed initialization in 1 ms
START - getCountry() for code: in
END - getCountry()
START - getCountry() for code: us
END - getCountry()
```

### 3.Spring \_REST Handson

## Display Employee List and Edit Employee form using RESTful Web Service

```
✓ src/main/resources
  static
  templates
  application.properties
  country.xml
  date-format.xml
  employee.xml
```

### Employee.xml

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

    <bean id="department1" class="com.cognizant.Spring_learn.model.Department">
        <property name="id" value="1" />
        <property name="name" value="Human Resources" />
    </bean>

    <bean id="department2" class="com.cognizant.Spring_learn.model.Department">
        <property name="id" value="2" />
        <property name="name" value="Finance" />
    </bean>

    <bean id="skill1" class="com.cognizant.Spring_learn.model.Skill">
        <property name="id" value="1" />
        <property name="name" value="Java" />
    </bean>

    <bean id="skill2" class="com.cognizant.Spring_learn.model.Skill">
        <property name="id" value="2" />
        <property name="name" value="SQL" />
    </bean>

    <bean id="skill3" class="com.cognizant.Spring_learn.model.Skill">
        <property name="id" value="3" />
        <property name="name" value="Python" />
    </bean>
```

```

<bean id="employee1" class="com.cognizant.Spring_learn.model.Employee">
    <property name="id" value="1001" />
    <property name="name" value="Alice Johnson" />
    <property name="department" ref="department1" />
    <property name="skills">
        <list>
            <ref bean="skill1" />
            <ref bean="skill2" />
        </list>
    </property>
</bean>

<bean id="employee2" class="com.cognizant.Spring_learn.model.Employee">
    <property name="id" value="1002" />
    <property name="name" value="Bob Williams" />
    <property name="department" ref="department2" />
    <property name="skills">
        <list>
            <ref bean="skill2" />
            <ref bean="skill3" />
        </list>
    </property>
</bean>

<bean id="employee3" class="com.cognizant.Spring_learn.model.Employee">
    <property name="id" value="1003" />
    <property name="name" value="Charlie Smith" />
    <property name="department" ref="department1" />
    <property name="skills">
        <list>
            <ref bean="skill3" />
        </list>
    </property>
</bean>

<bean id="employeeList" class="java.util.ArrayList">
    <constructor-arg>
        <list>
            <ref bean="employee1" />
            <ref bean="employee2" />
            <ref bean="employee3" />
        </list>
    </constructor-arg>
</bean>

</beans>

```

## Create new package

 com.cognizant.Spring\_learn.model  
 >  Department.java

## Department.java

```
package com.cognizant.Spring_learn.model;

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

    public Department() {}

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Department [id=" + id + ", name=" + name + "]";
    }
}
```

## Skill.java

```
package com.cognizant.Spring_learn.model;

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

    public Skill() {}

    public int getId() {
        return id;
    }

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

```
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Skill [id=" + id + ", name=" + name + "]";
    }
}
```

## Employee.java

```
package com.cognizant.Spring_Learn.model;

import java.util.List;

public class Employee {
    private int id;
    private String name;
    private Department department;
    private List<Skill> skills;

    public Employee() {}

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Skill [id=" + id + ", name=" + name + "]";
    }
}
```

```

package com.cognizant.Spring_learn.model;

import java.util.List;

public class Employee {
    private int id;
    private String name;
    private Department department;
    private List<Skill> skills;

    public Employee() {}

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Department getDepartment() {
        return department;
    }

    public void setDepartment(Department department) {
        this.department = department;
    }

    public List<Skill> getSkills() {
        return skills;
    }

    public void setSkills(List<Skill> skills) {
        this.skills = skills;
    }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ","
            + " department=" + department + ", skills=" + skills + "]";
    }
}

```

## Create new package

com.cognizant.Spring\_learn.dao  
 EmployeeDao.java

```
package com.cognizant.Spring_learn.dao;

import com.cognizant.Spring_learn.model.Employee;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class EmployeeDao {

    public List<Employee> getAllEmployees() {
        ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");
        return context.getBean("employeeList", List.class);
    }
}
```

## New package for EmployeeService

```
▼ com.cognizant.Spring_learn.service
  > EmployeeService.java
```

```
package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;
import com.cognizant.Spring_learn.model.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }
}
```

```
package com.cognizant.Spring_learn.controller;

import com.cognizant.Spring_learn.model.Employee;
import com.cognizant.Spring_learn.service.EmployeeService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

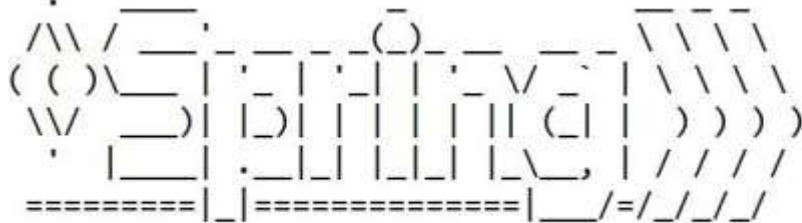
import java.util.List;

@RestController
public class EmployeeController {

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

    @Autowired
    private EmployeeService employeeService;

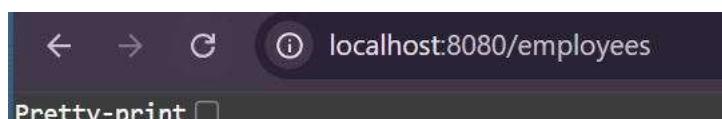
    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        LOGGER.info("START - getAllEmployees()");
        List<Employee> employees = employeeService.getAllEmployees();
        LOGGER.info("END - getAllEmployees()");
        return employees;
    }
}
```



:: Spring Boot :: (v3.5.3)

```
2025-07-12T15:38:54.124+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:38:54.124+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:38:54.319+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:38:54.320+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:38:54.320+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:38:54.353+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:38:54.353+05:30 INFO 12268 --- [ restartedMain]

c.c.Spring_learn.SpringApplication : No active profile set, falling back to 1 default profile
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
o.apache.catalina.core.StandardService : Starting service [Tomcat]
o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
```



```
[{"id":1001,"name":"Alice Johnson","department":{"id":1,"name":"Human Resources"}, "skills":[{"id":1,"name":"Java"}, {"id":2,"name":"SQL"}]}, {"id":1002,"name":"Bob Williams","department":{"id":2,"name":"Finance"}, "skills":[{"id":2,"name":"SQL"}, {"id":3,"name":"Python"}]}, {"id":1003,"name":"Charlie Smith","department":{"id":1,"name":"Human Resources"}, "skills":[{"id":3,"name":"Python"}]}]
```

## Create static employee list data using spring xml configuration

```
package com.cognizant.Spring_learn.dao;

import com.cognizant.Spring_learn.model.Employee;

@Repository
public class EmployeeDao {

    public List<Employee> getAllEmployees() {
        ApplicationContext context = new
            ClassPathXmlApplicationContext("employee.xml");
        return context.getBean("employeeList", List.class);
    }

    public Employee getEmployeeById(int id) {
        ApplicationContext context = new
            ClassPathXmlApplicationContext("employee.xml");
        List<Employee> employeeList = context.
            getBean("employeeList", List.class);

        for (Employee e : employeeList) {
            if (e.getId() == id) {
                return e;
            }
        }
        return null;
    }
}
```

### Employee.xml

```
<bean id="employeeList" class="java.util.ArrayList">
    <constructor-arg>
        <list>
            <ref bean="employee1" />
            <ref bean="employee2" />
            <ref bean="employee3" />
        </list>
    </constructor-arg>
</bean>

</beans>
```

```
package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;
import com.cognizant.Spring_learn.model.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }

    public Employee getEmployeeById(int id) {
        return employeeDao.getEmployeeById(id);
    }

}
```

```
package com.cognizant.Spring_learn.controller;

import com.cognizant.Spring_learn.model.Employee;
import com.cognizant.Spring_learn.service.EmployeeService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;
import org.springframework.web.bind.annotation.PathVariable;

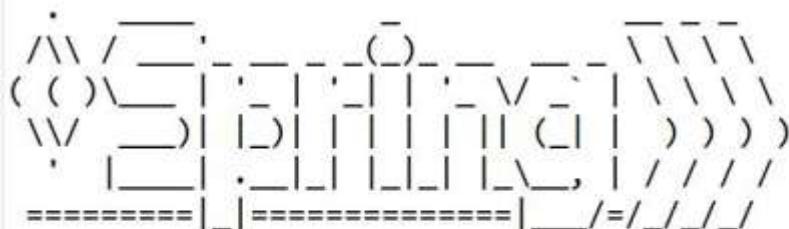
@RestController
public class EmployeeController {

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

    @Autowired
    private EmployeeService employeeService;

    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        LOGGER.info("START - getAllEmployees()");
        List<Employee> employees = employeeService.getAllEmployees();
        LOGGER.info("END - getAllEmployees()");
        return employees;
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployeeById(@PathVariable int id) {
        LOGGER.info("START - getEmployeeById({})", id);
        Employee employee = employeeService.getEmployeeById(id);
        LOGGER.info("END - getEmployeeById");
        return employee;
    }
}
```



:: Spring Boot :: (v3.5.3)

```
2025-07-12T15:51:05.507+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.507+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.684+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.685+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.685+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.699+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.699+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.749+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.764+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.767+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T15:51:05.768+05:30 INFO 12268 --- [ restartedMain]
```

```
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
o.apache.catalina.core.StandardService   : Starting service [Tomcat]
o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.17]
o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1 ms
o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
```

## OUTPUT:

```
← → ⌂ ⓘ localhost:8080/employees/1002
Pretty-print □
```

```
{
  "id": 1002,
  "name": "Bob Williams",
  "department": {
    "id": 2,
    "name": "Finance"
  },
  "skills": [
    {
      "id": 2,
      "name": "SQL"
    }
  ],
  "created_at": "2025-07-12T15:51:05.768Z"
}
```

```
c.c.Spring_learn.SpringLearnApplication  : Started SpringLearnApplication in 0.28 sec  
.ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged  
o.a.c.c.C.[Tomcat].[localhost]. [/]      : Initializing Spring DispatcherServlet 'disp  
o.s.web.servlet.DispatcherServlet       : Initializing Servlet 'dispatcherServlet'  
o.s.web.servlet.DispatcherServlet       : Completed initialization in 1 ms  
c.c.S.controller.EmployeeController     : START - getEmployeeById(1002)  
c.c.S.controller.EmployeeController     : END - getEmployeeById  
  
: Completed initialization in 1 ms  
: START - getEmployeeById(1002)  
: END - getEmployeeById
```

## Create REST service to gets all employees

### EmployeeDao.java

```
package com.cognizant.Spring_learn.dao;

import com.cognizant.Spring_learn.model.Employee;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.
ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class EmployeeDao {

    private static List<Employee> employeeList;

    public List<Employee> getAllEmployees() {
        if (employeeList == null) {
            ApplicationContext context = new
                ClassPathXmlApplicationContext("employee.xml");
            employeeList = context.getBean
                ("employeeList", List.class);
        }
        return employeeList;
    }
}
```

### Employee.xml

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

    <bean id="department1" class="com.cognizant.Spring_learn.model.Department">
        <property name="id" value="1" />
        <property name="name" value="HR" />
    </bean>

    <bean id="department2" class="com.cognizant.Spring_learn.model.Department">
        <property name="id" value="2" />
        <property name="name" value="Finance" />
    </bean>

```

```
<bean id="skill1" class="com.cognizant.Spring_learn.model.Skill">
    <property name="id" value="1" />
    <property name="name" value="Java" />
</bean>

<bean id="skill2" class="com.cognizant.Spring_learn.model.Skill">
    <property name="id" value="2" />
    <property name="name" value="SQL" />
</bean>

<bean id="employee1" class="com.cognizant.Spring_learn.model.Employee">
    <property name="id" value="1001" />
    <property name="name" value="Alice Johnson" />
    <property name="department" ref="department1" />
    <property name="skills">
        <list>
            <ref bean="skill1"/>
            <ref bean="skill2"/>
        </list>
    </property>
</bean>

<bean id="employee2" class="com.cognizant.Spring_learn.model.Employee">
    <property name="id" value="1002" />
    <property name="name" value="Bob Williams" />
    <property name="department" ref="department2" />
    <property name="skills">
        <list>
            <ref bean="skill2"/>
        </list>
    </property>
</bean>

<bean id="employeeList" class="java.util.ArrayList">
    <constructor-arg>
        <list>
            <ref bean="employee1" />
            <ref bean="employee2" />
        </list>
    </constructor-arg>
</bean>
</beans>
```

### **EmployeeService.java**

```
package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;
import com.cognizant.Spring_learn.model.Employee;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }
}
```

### **EmployeeController.java**

```
package com.cognizant.Spring_learn.controller;

import com.cognizant.Spring_learn.model.Employee;
import com.cognizant.Spring_learn.service.EmployeeService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
public class EmployeeController {

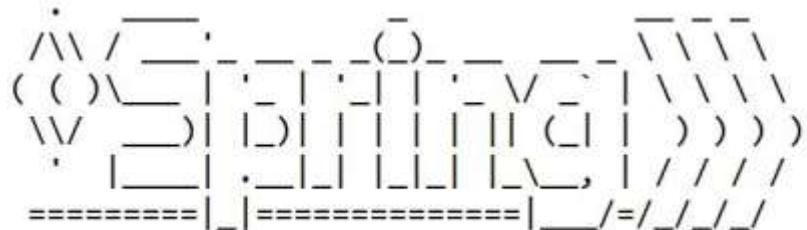
    private static final Logger LOGGER = LoggerFactory.
            getLogger(EmployeeController.class);

    @Autowired
    private EmployeeService employeeService;
```

```

    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        LOGGER.info("START - getAllEmployees()");
        List<Employee> employeeList = employeeService.getAllEmployees();
        LOGGER.info("END - getAllEmployees()");
        return employeeList;
    }
}

```



:: Spring Boot :: (v3.5.3)

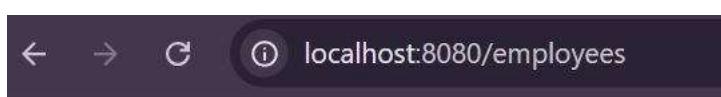
```

2025-07-12T16:04:18.132+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.132+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.285+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.285+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.285+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.309+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.309+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.358+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.365+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.368+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T16:04:18.369+05:30 INFO 12268 --- [ restartedMain]

c.c.Spring_learn.SpringApplication : No active profile set, falling back to 1 d
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
o.apache.catalina.core.StandardService : Starting service [Tomcat]
o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10
o.a.c.c.C.[Tomcat].localhost.1/1 : Initializing Spring embedded WebApplicatio

```

## OUTPUT:



```
← → ⌂ localhost:8080/employees  
Pretty-print   
[  
  {  
    "id": 1001,  
    "name": "Alice Johnson",  
    "department": {  
      "id": 1,  
      "name": "HR"  
    },  
    "skills": [  
      {  
        "id": 1,  
        "name": "Java"  
      },  
      {  
        "id": 2,  
        "name": "SQL"  
      }  
    ]  
  },  
  {  
    "id": 1002,  
    "name": "Bob Williams",  
    "department": {  
      "id": 2,  
      "name": "Finance"  
    },  
    "skills": [  
      {  
        "id": 2,  
        "name": "SQL"  
      }  
    ]  
  },  
  {
```

```
Completed initialization in 1 ms  
START - getAllEmployees()  
END - getAllEmployees()
```

### 3.1) REST - Get Employee by ID

#### EmployeeDao.java

```
package com.cognizant.Spring_learn.dao;

import com.cognizant.Spring_learn.model.Employee;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.
ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class EmployeeDao {

    private static List<Employee> employeeList;

    public List<Employee> getAllEmployees() {
        if (employeeList == null) {
            ApplicationContext context = new
                ClassPathXmlApplicationContext("employee.xml");
            employeeList = context.getBean
                ("employeeList", List.class);
        }
        return employeeList;
    }

    public Employee getEmployeeById(int id) {
        List<Employee> list = getAllEmployees();
        for (Employee e : list) {
            if (e.getId() == id) {
                return e;
            }
        }
        return null;
    }
}
```

### **EmployeeService.java**

```
package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }

    public Employee getEmployeeById(int id) {
        return employeeDao.getEmployeeById(id);
    }

}
```

### **EmployeeController.java**

```
package com.cognizant.Spring_learn.controller;

import com.cognizant.Spring_learn.model.Employee;
import com.cognizant.Spring_learn.service.EmployeeService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.PathVariable;

import java.util.List;

@RestController
public class EmployeeController {

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

    @Autowired
    private EmployeeService employeeService;
```

```

    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        LOGGER.info("START - getAllEmployees()");
        List<Employee> employeeList = employeeService.getAllEmployees();
        LOGGER.info("END - getAllEmployees()");
        return employeeList;
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployeeById(@PathVariable int id) {
        LOGGER.info("START - getEmployeeById({})", id);
        Employee employee = employeeService.getEmployeeById(id);
        LOGGER.info("END - getEmployeeById");
        return employee;
    }
}

```



:: Spring Boot :: (v3.5.3)

```

2025-07-12T16:13:14.468+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.SpringApplication
2025-07-12T16:13:14.468+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.SpringApplication
2025-07-12T16:13:14.628+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-12T16:13:14.628+05:30 INFO 12268 --- [ restartedMain] o.apache.catalina.core.StandardService
2025-07-12T16:13:14.629+05:30 INFO 12268 --- [ restartedMain] o.apache.catalina.core.StandardEngine
2025-07-12T16:13:14.643+05:30 INFO 12268 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]
2025-07-12T16:13:14.643+05:30 INFO 12268 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext
2025-07-12T16:13:14.680+05:30 INFO 12268 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer
2025-07-12T16:13:14.689+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer
2025-07-12T16:13:14.692+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.SpringApplication
2025-07-12T16:13:14.693+05:30 INFO 12268 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener

```

```

Tomcat initialized with port 8080 (http)
Starting service [Tomcat]
Starting Servlet engine: [Apache Tomcat/10.1.42]

```

## OUTPUT:

← → C ⓘ localhost:8080/employees/1001

```
{"id":1001,"name":"Alice Johnson","department":
{"id":1,"name":"HR"},"skills":
[{"id":1,"name":"Java"}, {"id":2,"name":"SQL"}]}
```



```
[{"id": 1001, "name": "Alice Johnson", "department": {"id": 1, "name": "HR"}, "skills": [{"id": 1, "name": "Java"}, {"id": 2, "name": "SQL"}]}]
```

### 3.2) REST - Add New Employee (POST)

```
package com.cognizant.Spring_learn.dao;

import com.cognizant.Spring_learn.model.Employee;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.
ClassPathXmlApplicationContext;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class EmployeeDao {

    private static List<Employee> employeeList;

    public List<Employee> getAllEmployees() {
        if (employeeList == null) {
            ApplicationContext context = new
                ClassPathXmlApplicationContext("employee.xml");
            employeeList = context.getBean
                ("employeeList", List.class);
        }
        return employeeList;
    }
}
```

```
    public Employee getEmployeeById(int id) {
        List<Employee> list = getAllEmployees();
        for (Employee e : list) {
            if (e.getId() == id) {
                return e;
            }
        }
        return null;
    }

    public void addEmployee(Employee employee) {
        getAllEmployees().add(employee);
    }

}
```

### EmployeeService.java

```
package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }

    public Employee getEmployeeById(int id) {
        return employeeDao.getEmployeeById(id);
    }

    public void addEmployee(Employee employee) {
        employeeDao.addEmployee(employee);
    }

}
```

### EmployeeController.java

```
package com.cognizant.Spring_learn.controller;

import com.cognizant.Spring_learn.model.Employee;
import com.cognizant.Spring_learn.service.EmployeeService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.PostMapping;
import java.util.List;

@RestController
public class EmployeeController {

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

    @Autowired
    private EmployeeService employeeService;

    ...
    ...
    ...

    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        LOGGER.info("START - getAllEmployees()");
        List<Employee> employeeList = employeeService.getAllEmployees();
        LOGGER.info("END - getAllEmployees()");
        return employeeList;
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployeeById(@PathVariable int id) {
        LOGGER.info("START - getEmployeeById({})", id);
        Employee employee = employeeService.getEmployeeById(id);
        LOGGER.info("END - getEmployeeById");
        return employee;
    }

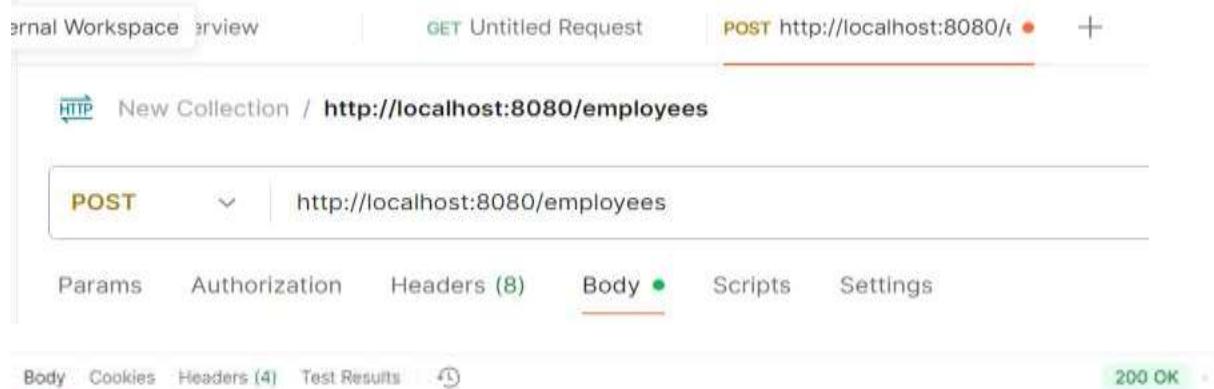
    @PostMapping("/employees")
    public void addEmployee(@RequestBody Employee employee) {
        LOGGER.info("START - addEmployee()");
        employeeService.addEmployee(employee);
        LOGGER.info("END - addEmployee()");
    }
}
```



:: Spring Boot :: (v3.5.3)

```
2025-07-12T16:19:23.083+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.S
2025-07-12T16:19:23.083+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.S
2025-07-12T16:19:23.344+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded.1
2025-07-12T16:19:23.346+05:30 INFO 12268 --- [ restartedMain] o.apache.catalina.
2025-07-12T16:19:23.346+05:30 INFO 12268 --- [ restartedMain] o.apache.catalina.
2025-07-12T16:19:23.368+05:30 INFO 12268 --- [ restartedMain] o.a.c.c.C.[Tomcat]
2025-07-12T16:19:23.368+05:30 INFO 12268 --- [ restartedMain] w.s.c.ServletWebSe
2025-07-12T16:19:23.438+05:30 INFO 12268 --- [ restartedMain] o.s.b.d.a.Optional
2025-07-12T16:19:23.445+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded.1
2025-07-12T16:19:23.448+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learn.S
2025-07-12T16:19:23.449+05:30 INFO 12268 --- [ restartedMain] .ConditionEvaluati
```

## Add new employee using postman



A screenshot of the Postman application interface. At the top, there are tabs for 'External Workspace' and 'Overview'. Below that, a 'GET Untitled Request' tab is shown, followed by a 'POST http://localhost:8080/employees' tab, which is currently active. The main area shows a collection named 'New Collection / http://localhost:8080/employees'. A 'POST' request is selected with the URL 'http://localhost:8080/employees'. Below the URL, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Scripts', and 'Settings'. The 'Body' tab is selected, indicated by a green dot. Under 'Body', there are tabs for 'Body', 'Cookies', 'Headers (4)', and 'Test Results'. To the right, a status indicator shows '200 OK'. The body content is currently empty.

## New employee



A screenshot of the Postman application interface, showing the JSON payload for a new employee. The 'Body' tab is selected, and the 'JSON' option is chosen. The JSON code is as follows:

```
1 {  
2   "id": 1003,  
3   "name": "John David",  
4   "department": {  
5     "id": 5,  
6     "name": "Arts"  
7   },  
8   "skills": [  
9     { "id": 2, "name": "HTML" },  
10    { "id": 4, "name": "Css" }  
11  ]  
12}  
13
```

← → ⌂ localhost:8080/employees

Pretty-print

```
[  
  {  
    "id": 1001,  
    "name": "Alice Johnson",  
    "department": {  
      "id": 1,  
      "name": "HR"  
    },  
    "skills": [  
      {  
        "id": 1,  
        "name": "Java"  
      },  
      {  
        "id": 2,  
        "name": "SQL"  
      }  
    ]  
  },  
  {  
    "id": 1002,  
    "name": "Bob Williams",  
    "department": {  
      "id": 2,  
      "name": "Finance"  
    },  
    "skills": [  
      {  
        "id": 2,  
        "name": "SQL"  
      }  
    ]  
  },  
  {  
    "id": 1003,  
    "name": "Charlie Brown",  
    "department": {  
      "id": 3,  
      "name": "Marketing"  
    },  
    "skills": [  
      {  
        "id": 3,  
        "name": "SEO"  
      },  
      {  
        "id": 4,  
        "name": "SEM"  
      }  
    ]  
  },  
  {  
    "id": 1004,  
    "name": "David Lee Updated",  
    "department": {  
      "id": 3,  
      "name": "Engineering"  
    },  
    "skills": [  
      {  
        "id": 3,  
        "name": "Spring Boot"  
      },  
      {  
        "id": 4,  
        "name": "MySQL"  
      }  
    ]  
  },  
]
```

```
{  
    "id": 1003,  
    "name": "John David",  
    "department": {  
        "id": 5,  
        "name": "Arts"  
    },  
    "skills": [  
        {  
            "id": 2,  
            "name": "HTML"  
        },  
        {  
            "id": 4,  
            "name": "Css"  
        }  
    ]  
}
```

```
START - addEmployee()  
END - addEmployee()  
START - getAllEmployees()  
END - getAllEmployees()
```

### 3.3) PUT – Update existing employee

#### EmployeeController.java

```
package com.cognizant.Spring_learn.controller;  
  
import com.cognizant.Spring_learn.model.Employee;  
  
@RestController  
public class EmployeeController {  
  
    private static final Logger LOGGER = LoggerFactory.  
        getLogger(EmployeeController.class);  
  
    @Autowired  
    private EmployeeService employeeService;  
  
    @GetMapping("/employees")  
    public List<Employee> getAllEmployees() {  
        LOGGER.info("START - getAllEmployees()");  
        List<Employee> employeeList = employeeService.getAllEmployees();  
        LOGGER.info("END - getAllEmployees()");  
        return employeeList;  
    }  
}
```

```

    @GetMapping("/employees/{id}")
    public Employee getEmployeeById(@PathVariable int id) {
        LOGGER.info("START - getEmployeeById({})", id);
        Employee employee = employeeService.getEmployeeById(id);
        LOGGER.info("END - getEmployeeById");
        return employee;
    }

    @PostMapping("/employees")
    public void addEmployee(@RequestBody Employee employee) {
        LOGGER.info("START - addEmployee()");
        employeeService.addEmployee(employee);
        LOGGER.info("END - addEmployee()");
    }

    @PutMapping("/employees")
    public void updateEmployee(@RequestBody Employee employee) {
        LOGGER.info("START - updateEmployee()");
        LOGGER.debug("Employee to update: {}", employee);
        employeeService.updateEmployee(employee);
        LOGGER.info("END - updateEmployee()");
    }

}

```

### **EmployeeService.java**

```

package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;□

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }

    public Employee getEmployeeById(int id) {
        return employeeDao.getEmployeeById(id);
    }

    public void addEmployee(Employee employee) {
        employeeDao.addEmployee(employee);
    }
}

```

```
public void updateEmployee(Employee employee) {  
    List<Employee> employees = employeeDao.getAllEmployees();  
    for (int i = 0; i < employees.size(); i++) {  
        if (employees.get(i).getId() == employee.getId()) {  
            employees.set(i, employee);  
            return;  
        }  
    }  
}
```

## Console

## Update existing employee details using postman

The screenshot shows the Postman application interface. A POST request is being made to the URL `http://localhost:8080/employees`. The request body is defined in JSON format, containing fields for id, name, department, and skills. The response status is 200 OK.

```
1 {  
2   "id": 1002,  
3   "name": "Updated Employee",  
4   "department": {  
5     "id": 2,  
6     "name": "HR"  
7   },  
8   "skills": [  
9     { "id": 1, "name": "Java" },  
10    { "id": 2, "name": "Spring" }  
11  ]  
12}  
13
```

Body 200 OK

### OUTPUT:

```
{  
  "id": 1002,  
  "name": "Updated Employee",  
  "department": {  
    "id": 2,  
    "name": "HR"  
  },  
  "skills": [  
    {  
      "id": 1,  
      "name": "Java"  
    },  
    {  
      "id": 2,  
      "name": "Spring"  
    }  
  ]  
}
```

```
: START - addEmployee()  
: END - addEmployee()  
: START - getAllEmployees()  
: END - getAllEmployees()
```

### 3.4) DELETE – Remove an Employee by ID

#### EmployeeController.java

```
package com.cognizant.Spring_learn.controller;

import com.cognizant.Spring_learn.model.Employee;

@RestController
public class EmployeeController {

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

    @Autowired
    private EmployeeService employeeService;

    @GetMapping("/employees")
    public List<Employee> getAllEmployees() {
        LOGGER.info("START - getAllEmployees()");
        List<Employee> employeeList = employeeService.getAllEmployees();
        LOGGER.info("END - getAllEmployees()");
        return employeeList;
    }

    @GetMapping("/employees/{id}")
    public Employee getEmployeeById(@PathVariable int id) {
        LOGGER.info("START - getEmployeeById({})", id);
        Employee employee = employeeService.getEmployeeById(id);
        LOGGER.info("END - getEmployeeById");
        return employee;
    }

    @PostMapping("/employees")
    public void addEmployee(@RequestBody Employee employee) {
        LOGGER.info("START - addEmployee()");
        employeeService.addEmployee(employee);
        LOGGER.info("END - addEmployee()");
    }

    @PutMapping("/employees")
    public void updateEmployee(@RequestBody Employee employee) {
        LOGGER.info("START - updateEmployee()");
        LOGGER.debug("Employee to update: {}", employee);
        employeeService.updateEmployee(employee);
        LOGGER.info("END - updateEmployee()");
    }
}
```

```
    }

    @DeleteMapping("/employees/{id}")
    public void deleteEmployee(@PathVariable int id) {
        LOGGER.info("START - deleteEmployee()");
        employeeService.deleteEmployee(id);
        LOGGER.info("END - deleteEmployee()");
    }

}
```

### EmployeeService.java

```
package com.cognizant.Spring_learn.service;

import com.cognizant.Spring_learn.dao.EmployeeDao;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeDao employeeDao;

    public List<Employee> getAllEmployees() {
        return employeeDao.getAllEmployees();
    }

    public Employee getEmployeeById(int id) {
        return employeeDao.getEmployeeById(id);
    }

    public void addEmployee(Employee employee) {
        employeeDao.addEmployee(employee);
    }

    public void updateEmployee(Employee employee) {
        List<Employee> employees = employeeDao.getAllEmployees();
        for (int i = 0; i < employees.size(); i++) {
            if (employees.get(i).getId() == employee.getId()) {
                employees.set(i, employee);
                return;
            }
        }
    }

    public void deleteEmployee(int id) {
        employeeDao.deleteEmployee(id);
    }
}
```

## EmployeeDao.java

```
package com.cognizant.Spring_learn.dao;

import com.cognizant.Spring_learn.model.Employee;

@Repository
public class EmployeeDao {

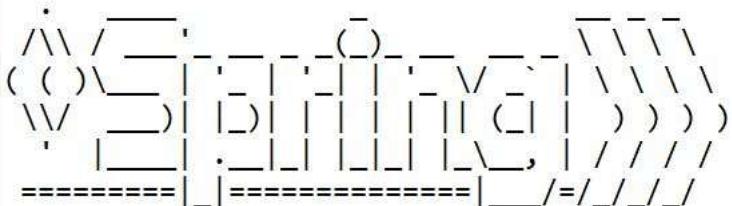
    private static List<Employee> employeeList;

    public List<Employee> getAllEmployees() {
        if (employeeList == null) {
            ApplicationContext context = new
                ClassPathXmlApplicationContext("employee.xml");
            employeeList = context.getBean
                ("employeeList", List.class);
        }
        return employeeList;
    }

    public Employee getEmployeeById(int id) {
        List<Employee> list = getAllEmployees();
        for (Employee e : list) {
            if (e.getId() == id) {
                return e;
            }
        }
        return null;
    }

    public void addEmployee(Employee employee) {
        getAllEmployees().add(employee);
    }

    public void deleteEmployee(int id) {
        List<Employee> employees = getAllEmployees();
        employees.removeIf(e -> e.getId() == id);
    }
}
```



:: Spring Boot :: (v3.5.3)

```
2025-07-12T17:20:56.000+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learni
2025-07-12T17:20:56.000+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learni
2025-07-12T17:20:56.300+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded
2025-07-12T17:20:56.303+05:30 INFO 12268 --- [ restartedMain] o.apache.catalin
2025-07-12T17:20:56.303+05:30 INFO 12268 --- [ restartedMain] o.apache.catalin
2025-07-12T17:20:56.324+05:30 INFO 12268 --- [ restartedMain] o.a.c.c.C.[Tomc
2025-07-12T17:20:56.324+05:30 INFO 12268 --- [ restartedMain] w.s.c.ServletWel
2025-07-12T17:20:56.393+05:30 INFO 12268 --- [ restartedMain] o.s.b.d.a.Option
2025-07-12T17:20:56.407+05:30 INFO 12268 --- [ restartedMain] o.s.b.w.embedded
2025-07-12T17:20:56.411+05:30 INFO 12268 --- [ restartedMain] c.c.Spring_learni
2025-07-12T17:20:56.413+05:30 INFO 12268 --- [ restartedMain] .ConditionEvalu
```

## Delete the employee details using postman

The screenshot shows a Postman request configuration. The method is set to **DELETE**, the URL is **http://localhost:8080/employees/1002**, and the **Body** tab is selected with the **JSON** option chosen. The body content is a simple JSON object with two fields: **1** and **2**. Below the request, the response status is shown as **200 OK** with a **8 ms** duration.

## OUTPUT:

```
START - deleteEmployee()
END - deleteEmployee()
START - getAllEmployees()
END - getAllEmployees()
```

## Create REST service for department

```
<?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="departmentList" class="java.util.ArrayList">
        <constructor-arg>
            <list>
                <bean class="com.cognizant.Spring_learn.model.Department">
                    <property name="id" value="1"/>
                    <property name="name" value="HR"/>
                </bean>
                <bean class="com.cognizant.Spring_learn.model.Department">
                    <property name="id" value="2"/>
                    <property name="name" value="Finance"/>
                </bean>
            </list>
        </constructor-arg>
    </bean>

</beans>

package com.cognizant.Spring_learn.model;

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

    public Department() {}

    public Department(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

```
package com.cognizant.Spring_learn.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.beans.factory.annotation.Qualifier;
import com.cognizant.Spring_learn.model.Department;

@Service
public class DepartmentService {

    @Autowired
    @Qualifier("departmentList")
    private List<Department> departmentList;

    public List<Department> getAllDepartments() {
        return departmentList;
    }
}

package com.cognizant.Spring_learn.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import com.cognizant.Spring_learn.model.Department;
import com.cognizant.Spring_learn.service.DepartmentService;

@RestController
public class DepartmentController {

    @Autowired
    private DepartmentService departmentService;

    @GetMapping("/departments")
    public List<Department> getAllDepartments() {
        return departmentService.getAllDepartments();
    }
}
```

```

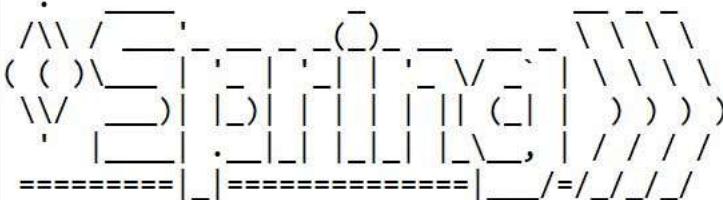
package com.cognizant.Spring_learn;

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

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

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

```



:: Spring Boot :: (v3.5.3)

```

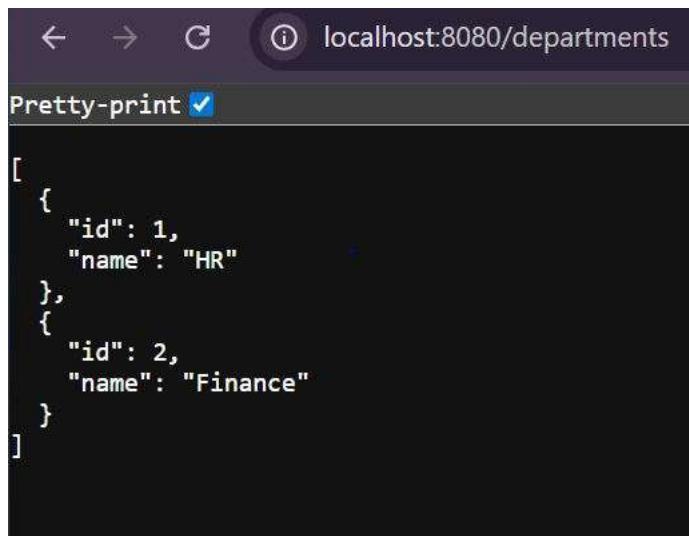
2025-07-12T18:48:24.794+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:24.795+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.208+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.216+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.221+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.309+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.309+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.458+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.477+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.492+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:25.500+05:30 INFO 12268 --- [ restartedMain]
2025-07-12T18:48:49.723+05:30 INFO 12268 --- [nio-8080-exec-1]
2025-07-12T18:48:49.724+05:30 INFO 12268 --- [nio-8080-exec-1]
2025-07-12T18:48:49.725+05:30 INFO 12268 --- [nio-8080-exec-1]

```

```

c.c.Spring_learn.SpringLearnApplication : Starting SpringLearnApplication using Java 21.0.7 with PID
c.c.Spring_learn.SpringLearnApplication : No active profile set, falling back to 1 default profile: "dev"
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
o.apache.catalina.core.StandardService : Starting service [Tomcat]
o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
o.a.c.c.C.[Tomcat-8].[localhost].[/] : Initializing Spring embedded WebApplicationContext
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 516 ms
o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
c.c.Spring_learn.SpringLearnApplication : Started SpringLearnApplication in 0.741 seconds (process time: 0.741ms)
.c.ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
o.a.c.c.C.[Tomcat-8].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms

```



localhost:8080/departments

Pretty-print

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