

Spring Core – Implement data model for Employee using Spring XML Configuration

1. Created package `com.cognizant.spring-learn.bean`
2. Created class `com.cognizant.spring-learn.bean.Employee` with following attributes

Type	Attribute Name
int	id
String	name
double	salary
boolean	permanent
Date	dateOfBirth

3. Generated getters, setters and `toString()` methods.
4. Included empty parameter constructor with debug log.
5. Defined values for the attributes in a new spring xml configuration file named `employee.xml`
6. Displayed the employee details in a new method `displayEmployee()` in main method of `SpringLearnApplication.java`

Springlearn3Application.java:

```
package com.cognizant.springlearn.bean.springlearn3;

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.ParseException;
import java.text.SimpleDateFormat;

@SpringBootApplication
public class Springlearn3Application {
    public static void displayEmployee()
    {
        ApplicationContext context = new
ClassPathXmlApplicationContext("employee.xml");
        Employee emp = context.getBean("employee", Employee.class);
        System.out.println(emp.toString());
    }

    public static void main(String[] args) {
        SpringApplication.run(Springlearn3Application.class, args);
    }
}
```

```
        displayEmployee();  
    }  
}
```

Employee.java:

```
package com.cognizant.springlearn.bean.springlearn3;  
  
import java.util.Date;  
  
public class Employee {  
  
    private int id;  
  
    private String name;  
  
    private double salary;  
  
    private boolean permanent;  
  
    private Date dateOfBirth;  
  
    public Employee() {  
        super();  
    }  
  
    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 double getSalary() {

        return salary;

    }

    public void setSalary(double salary) {

        this.salary = salary;

    }

    public boolean isPermanent() {

        return permanent;

    }

    public void setPermanent(boolean permanent) {

        this.permanent = permanent;

    }

    public Date getDateOfBirth() {

        return dateOfBirth;

    }

    public void setDateOfBirth(Date dateOfBirth) {

        this.dateOfBirth = dateOfBirth;

    }

    @Override

    public String toString() {

        return "Employee [id=" + id + ", name=" + name + ",
salary=" + salary + ", permanent=" + permanent
        + ", dateOfBirth=" + dateOfBirth + "];"

    }

}

```

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="dateFormat" class="java.text.SimpleDateFormat">
    <constructor-arg value="yyyy-MM-dd" />
  </bean>
  <bean id="employee"
        class="com.cognizant.springlearn.bean.springlearn3.Employee">
    <property name="id" value="101"/>
    <property name="name" value="John Doe"/>
    <property name="salary" value="75000.0"/>
    <property name="permanent" value="true"/>
    <property name="dateOfBirth">
      <bean factory-bean="dateFormat" factory-method="parse">
        <constructor-arg value="2010-01-31" />
      </bean>
    </property>
  </bean></beans>
```

Output:

```
Springlearn3Application.java
10 @SpringBootApplication
11 public class Springlearn3Application {
12     public static void displayEmployee()
13     {
14         ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");
15         Employee emp = context.getBean("employee", Employee.class);
16         System.out.println(emp.toString());
17     }
18 }
19
20
21 public static void main(String[] args) {
22     SpringApplication.run(Springlearn3Application.class, args);
23 }
24
25 displayEmployee();
26
```

```
<terminated> Springlearn3Application [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 20, 2024, 2:29:18 PM - 2:29:19 PM) [pid: 7340]

:: Spring Boot ::
2024-03-20T14:29:18.995+05:30 INFO 7340 --- [main] c.c.s.b.s.Springlearn3Application : Starting Springlearn
2024-03-20T14:29:18.997+05:30 INFO 7340 --- [main] c.c.s.b.s.Springlearn3Application : No active profile se
2024-03-20T14:29:19.450+05:30 INFO 7340 --- [main] c.c.s.b.s.Springlearn3Application : Started Springlearn3
Employee [id=101, name=John Doe, salary=75000.0, permanent=true, dateOfBirth=Sun Jan 31 00:00:00 IST 2010]
```

Spring Core - Implement data model for Department in Employee

1. Create `com.cognizant.spring-learn.bean.Department`
2. Added attributes `id` and `name`
3. Included empty parameter constructor with log
4. Generated getters, setters and `toString()`
5. Include department as an attribute in Employee class
6. Include getters/setters for department
7. Regenerated `toString()` method in Employee for inclusion of department
8. In `employee.xml`, using `ref` attribute include department to employee.

```
package com.cognizant.springlearn.bean.springlearn3;
```

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

```
public class Employee {
    private int id;
    private String name;
    private double salary;
    private boolean permanent;
    private Date dateOfBirth;
    private Department department;
    public Employee() {

        super();
    }
    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 double getSalary() {

        return salary;
    }
}
```

```

    }

    public void setSalary(double salary) {

        this.salary = salary;

    }

    public boolean isPermanent() {

        return permanent;

    }

    public void setPermanent(boolean permanent) {

        this.permanent = permanent;

    }

    public Date getDateOfBirth() {

        return dateOfBirth;

    }

    public void setDateOfBirth(Date dateOfBirth) {

        this.dateOfBirth = dateOfBirth;

    }

    public Department getDepartment() {
        return department;
    }

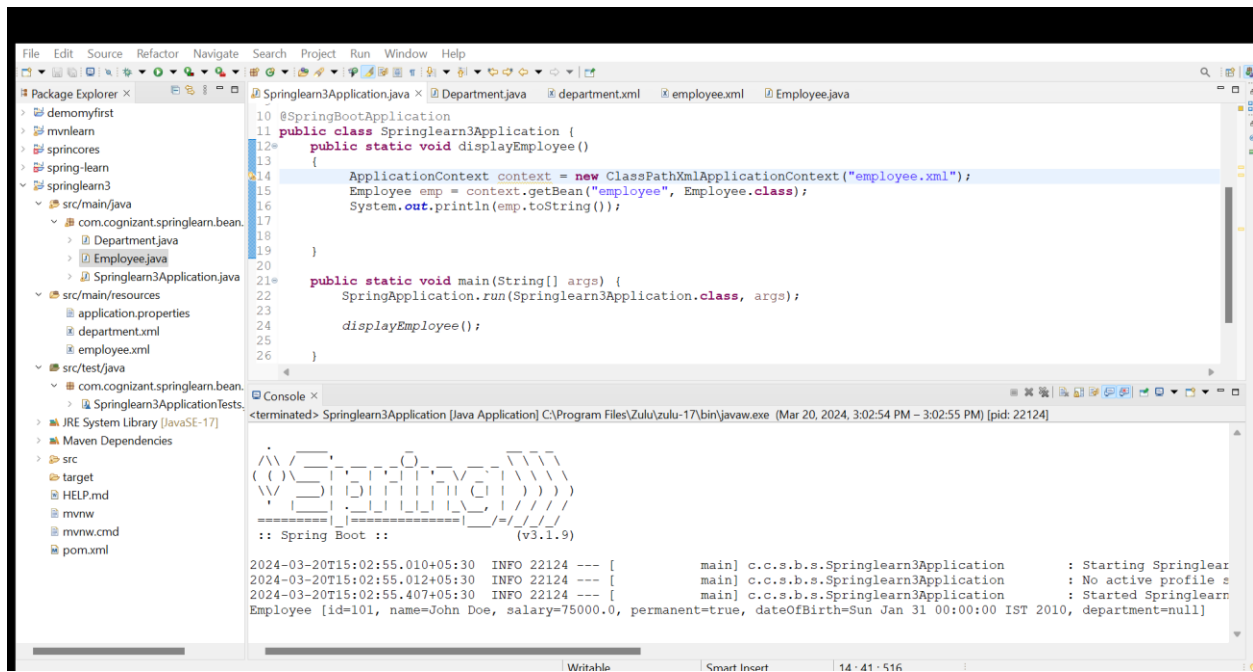
    public void setDepartment(Department department) {
        this.department = department;
    }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ",
salary=" + salary + ", permanent=" + permanent
        + ", dateOfBirth=" + dateOfBirth + ", department=" +
department + "]\n";
    }

}

```

Output:



The screenshot shows an IDE with the following components:

- Package Explorer:** Shows a project structure with packages like `com.cognizant.springlearn.bean` and `src/main/java`. The `Employee.java` file is selected.
- Editor:** Displays the `Springlearn3Application.java` file. The code includes a `displayEmployee()` method that creates a `ClassPathXmlApplicationContext` and retrieves an `Employee` bean from `employee.xml`. The `main` method calls `SpringApplication.run()` and `displayEmployee()`.
- Console:** Shows the output of the application. It starts with the Spring Boot logo and version `(v3.1.9)`. The output indicates that the application is starting, no active profile is set, and the `Employee` bean is successfully retrieved from the XML file.

```
10 @SpringBootApplication
11 public class Springlearn3Application {
12     public static void displayEmployee()
13     {
14         ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");
15         Employee emp = context.getBean("employee", Employee.class);
16         System.out.println(emp.toString());
17     }
18 }
19
20
21 public static void main(String[] args) {
22     SpringApplication.run(Springlearn3Application.class, args);
23 }
24
25 displayEmployee();
26 }
```

```
<terminated> Springlearn3Application [Java Application] C:\Program Files\Zulu\zulu-17\bin\javaw.exe (Mar 20, 2024, 3:02:54 PM - 3:02:55 PM) [pid: 22124]

:: Spring Boot :: (v3.1.9)

2024-03-20T15:02:55.010+05:30 INFO 22124 --- [main] c.c.s.b.s.Springlearn3Application : Starting Springlearn3Application
2024-03-20T15:02:55.012+05:30 INFO 22124 --- [main] c.c.s.b.s.Springlearn3Application : No active profile set, using default
2024-03-20T15:02:55.407+05:30 INFO 22124 --- [main] c.c.s.b.s.Springlearn3Application : Started Springlearn3Application
Employee [id=101, name=John Doe, salary=75000.0, permanent=true, dateOfBirth=Sun Jan 31 00:00:00 IST 2010, department=null]
```

Hands on 3

Spring Core - Include Skill details for Employee

- Create new class `com.cognizant.spring-learn.bean.Skill`
- Create instance variables `id` and `name` with data types `int` and `String` respectively.
- Include, empty parameter constructor with log, getters, setters and `toString()`
- Add `skills` as instance variable of type `Skill` array in `Employee`.
- In `Employee` class, include getters, setters and update the `toString()` method for inclusion of `Skill[]`
- Create multiple skill instances in `employee.xml` and add them to array of `Skills`.
- Set the array of `Skills` to `skills` of `Employee` bean in configuration `xml` file
- Load the `employee` class from `employee.xml` and display the data.
- Refer sample code below inclusion of skills property for `employee` in `employee.xml`

Skill.java:

```
package com.cognizant.springlearn.bean.springlearn3;
```

```
public class Skill {
    private int id;
    private String name;
    @Override
    public String toString() {
        return "Skill [id=" + id + ", name=" + name + "]";
    }
    public Skill() {
        System.out.println("Inside Skill Constructor");
    }
    public int getId() {
        System.out.println("Getting id: " + id);
        return id;
    }
    public void setId(int id) {
        System.out.println("Setting id: " + id);
        this.id = id;
    }
    public String getName() {
        System.out.println("Getting name: " + name);
        return name;
    }
    public void setName(String name) {
        System.out.println("Setting name: " + name);
        this.name = name;
    }
}
```


Springlearn3Application.java:

```
package com.cognizant.springlearn.bean.springlearn3;

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.ParseException;
import java.text.SimpleDateFormat;

@SpringBootApplication
public class Springlearn3Application {
    public static void displayEmployee()
    {
        ApplicationContext context = new
ClassPathXmlApplicationContext("employee.xml");
        Employee emp = context.getBean("employeeSkill", Employee.class);
        System.out.println(emp.toString());
    }

    public static void main(String[] args) {
        SpringApplication.run(Springlearn3Application.class, args);

        displayEmployee();
    }
}
```

Employee.java:

```
package com.cognizant.springlearn.bean.springlearn3;

import java.util.Arrays;
import java.util.Date;

public class Employee {
    private int id;
    private String name;
    private double salary;
    private boolean permanent;
    private Date dateOfBirth;
    private Department department;
    private Skill[] skills;
    public Employee() {

        super();
    }
}
```

```
    }  
    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 double getSalary() {  
        return salary;  
    }  
    public void setSalary(double salary) {  
        this.salary = salary;  
    }  
    public boolean isPermanent() {  
        return permanent;  
    }  
    public void setPermanent(boolean permanent) {  
        this.permanent = permanent;  
    }  
    public Date getDateOfBirth() {  
        return dateOfBirth;  
    }  
    public void setDateOfBirth(Date dateOfBirth) {  
        this.dateOfBirth = dateOfBirth;  
    }  
    public Department getDepartment() {
```

```

        return department;
    }
    public void setDepartment(Department department) {
        this.department = department;
    }
    public Skill[] getSkills() {
        return skills;
    }
    public void setSkills(Skill[] skills) {
        this.skills = skills;
    }
    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ", salary=" +
salary + ", permanent=" + permanent
        + ", dateOfBirth=" + dateOfBirth + ", department=" +
department + ", skills=" + Arrays.toString(skills)
        + "]";
    }
}

```

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
https://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="skillHistory"
class="com.cognizant.springlearn.bean.springlearn3.Skill">
        <property name="id" value="10"/>
        <property name="name" value="History"/>
    </bean>

    <bean id="skillGeography"
class="com.cognizant.springlearn.bean.springlearn3.Skill">
        <property name="id" value="20"/>
        <property name="name" value="Geography"/>
    </bean>

    <bean id="skillEcomics"
class="com.cognizant.springlearn.bean.springlearn3.Skill">
        <property name="id" value="3"/>
        <property name="name" value="Economics"/>
    </bean>

```

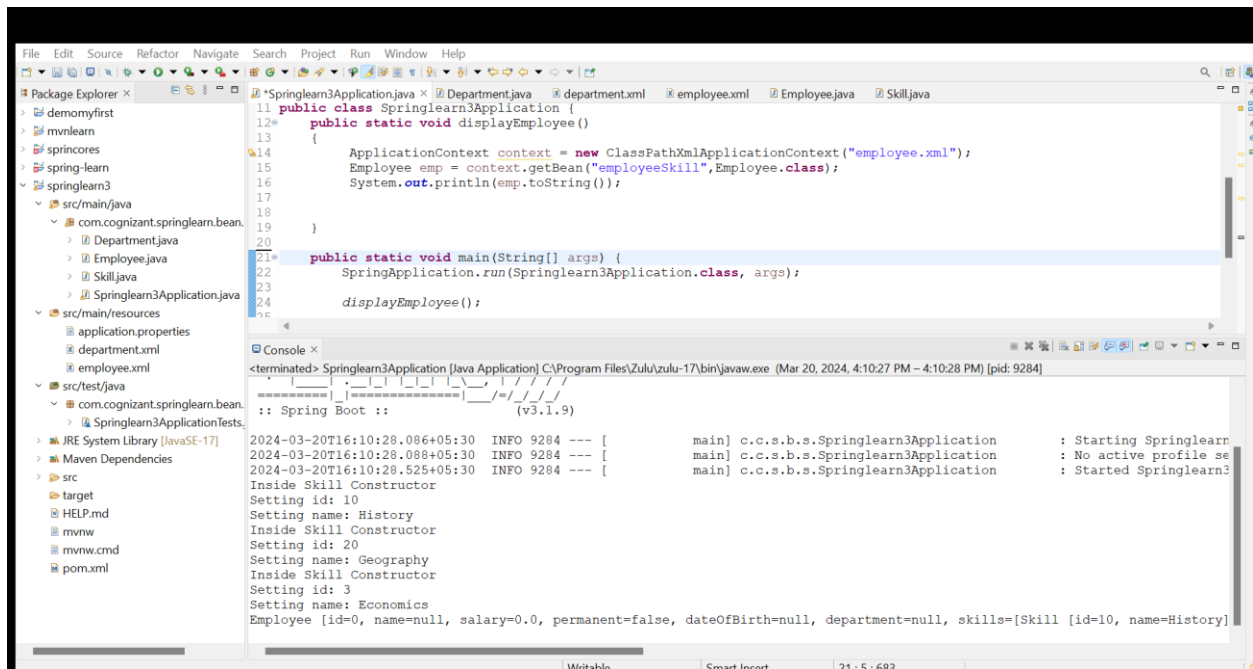
```

<bean id="employeeSkill"
class="com.cognizant.springlearn.bean.springlearn3.Employee">

    <property name="skills">
        <array>
            <ref bean="skillHistory" />
            <ref bean="skillGeography" />
            <ref bean="skillEcomics" />
        </array>
    </property>
</bean>
</beans>

```

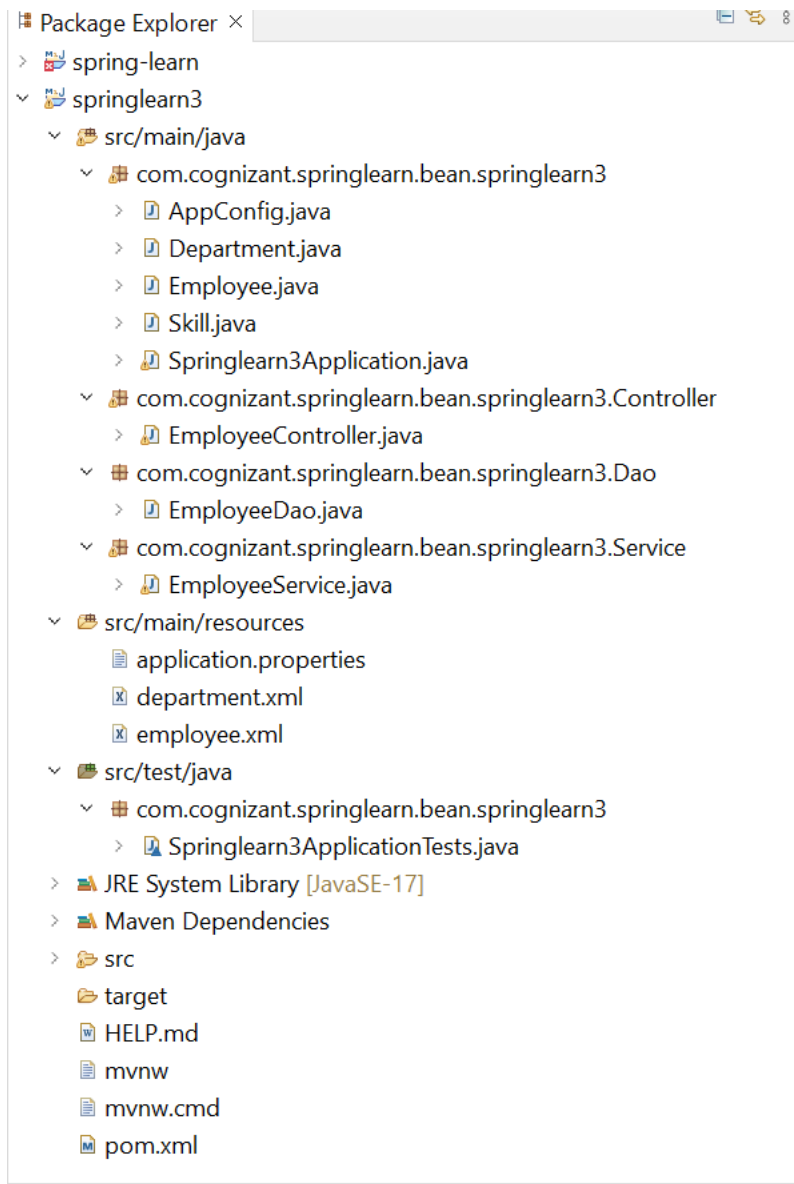
OUTPUT:



Hands on 4

Spring Core - Inversion of Control / Dependency Injection

1. Created necessary java classes with necessary attributes.
2. Created necessary getter and setter method for implementation.



EmployeeController.java:

```
package com.cognizant.springlearn.bean.springlearn3.Controller;  
  
import org.springframework.beans.factory.annotation.Autowired;
```

```

import org.springframework.stereotype.Controller;

import com.cognizant.springlearn.bean.springlearn3.Service.EmployeeService;

@Controller
public class EmployeeController {
    private EmployeeService employeeService;

    @Autowired
    public void setEmployeeService(EmployeeService employeeService) {
        this.employeeService = employeeService;
        System.out.println("Setter method called for EmployeeService in
EmployeeController");
    }

}

```

EmployeeDao.java:

```

package com.cognizant.springlearn.bean.springlearn3.Dao;
import org.springframework.stereotype.Component;
@Component
public class EmployeeDao {

    EmployeeDao() {
        System.out.println("EmployeeDao constructor called");
    }

}

```

EmployeeService.java:

```

package com.cognizant.springlearn.bean.springlearn3.Service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

import com.cognizant.springlearn.bean.springlearn3.Dao.EmployeeDao;
@Component
public class EmployeeService {
    private EmployeeDao employeeDao;

    @Autowired
    public void setEmployeeDao(EmployeeDao employeeDao) {
        this.employeeDao = employeeDao;
        System.out.println("Setter method called for EmployeeDao in
EmployeeService");
    }

}

```

AppConfig.java:

```

package com.cognizant.springlearn.bean.springlearn3;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

@Configuration
@ComponentScan("com.cognizant.springlearn")

public class AppConfig {

}

```

Springlearn3Application.java:

```

package com.cognizant.springlearn.bean.springlearn3;

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

import com.cognizant.springlearn.bean.springlearn3.Controller.EmployeeController;

import java.text.ParseException;
import java.text.SimpleDateFormat;

@SpringBootApplication
public class Springlearn3Application {
    public static void displayEmployee()
    {
        ApplicationContext context = new
ClassPathXmlApplicationContext("employee.xml");
        Employee emp = context.getBean("employeeSkill", Employee.class);
        System.out.println(emp.toString());

    }

    public static void
displayEmployeeControllerAnnotation(ApplicationContext applicationContext) {
        EmployeeController employeeController = (EmployeeController)
applicationContext.getBean("employeeController");
        System.out.println("Loaded EmployeeController bean: " +
employeeController);
    }

    public static void main(String[] args) {
        SpringApplication.run(Springlearn3Application.class, args);

        displayEmployee();
    }
}

```

Output:

