# Question 1 - Create static employee list data using Spring XML configuration

Code 1: 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">  
  
 <!-- Department beans -->  
 <bean id="department1" class="com.cognizant.springlearn.Department">  
 <property name="id" value="1"/>  
 <property name="name" value="Human Resources"/>  
 </bean>  
  
 <bean id="department2" class="com.cognizant.springlearn.Department">  
 <property name="id" value="2"/>  
 <property name="name" value="Finance"/>  
 </bean>  
  
 <!-- Employee beans -->  
 <bean id="employee1" class="com.cognizant.springlearn.Employee">  
 <property name="id" value="1"/>  
 <property name="name" value="John Doe"/>  
 <property name="salary" value="50000"/>  
 <property name="department" ref="department1"/>  
 </bean>  
  
 <bean id="employee2" class="com.cognizant.springlearn.Employee">  
 <property name="id" value="2"/>  
 <property name="name" value="Jane Smith"/>  
 <property name="salary" value="60000"/>  
 <property name="department" ref="department2"/>  
 </bean>  
  
 <bean id="employeeList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="employee1"/>  
 <ref bean="employee2"/>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

Code 2: EmployeeDao.java

package com.cognizant.springlearn.dao;  
  
import java.util.ArrayList;  
import java.util.List;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Repository;  
import com.cognizant.springlearn.Employee;  
  
@Repository  
public class EmployeeDao {  
 private static List<Employee> EMPLOYEE\_LIST;  
  
 public EmployeeDao() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");  
 EMPLOYEE\_LIST = (ArrayList<Employee>) context.getBean("employeeList");  
 }  
  
 public List<Employee> getAllEmployees() {  
 return EMPLOYEE\_LIST;  
 }  
}

Output:  
Loaded employees:   
[Employee [id=1, name=John Doe, salary=50000.0, department=Department [id=1, name=Human Resources]],  
 Employee [id=2, name=Jane Smith, salary=60000.0, department=Department [id=2, name=Finance]]]

# Question 2 - Create REST service to get all employees

Code 1: EmployeeService.java

package com.cognizant.springlearn.service;  
  
import java.util.List;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
import com.cognizant.springlearn.Employee;  
import com.cognizant.springlearn.dao.EmployeeDao;  
  
@Service  
public class EmployeeService {  
  
 @Autowired  
 private EmployeeDao employeeDao;  
  
 @Transactional  
 public List<Employee> getAllEmployees() {  
 return employeeDao.getAllEmployees();  
 }  
}

Code 2: EmployeeController.java

package com.cognizant.springlearn.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.springlearn.Employee;  
import com.cognizant.springlearn.service.EmployeeService;  
  
@RestController  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 @GetMapping("/employees")  
 public List<Employee> getAllEmployees() {  
 return employeeService.getAllEmployees();  
 }  
}

Output (Sample JSON Response):  
[  
 { "id": 1, "name": "John Doe", "salary": 50000.0, "department": { "id": 1, "name": "Human Resources" } },  
 { "id": 2, "name": "Jane Smith", "salary": 60000.0, "department": { "id": 2, "name": "Finance" } }  
]