

Spring Data JPA

Hands on 1

- Spring Data JPA - Quick Example

➔ solution:

Dependencies

Spring boot dev tools, Spring data JPA, MySql driver

Main.java

```
package com.cognizant.orm_learn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

@SpringBootApplication

```
public class OrmLearnApplication {
```

```
    private static final Logger LOGGER =
```

```
    LoggerFactory.getLogger(OrmLearnApplication.class);
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(OrmLearnApplication.class, args);
```

```
        LOGGER.info("Inside main");
```

```
    }
```

```
}
```

```
14:04:27.504 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4002, org.hibernate.t
14:04:27.593 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2004, org.hibernate.t
14:04:27.596 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2005, org.hibernate.t
14:04:27.596 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2011, org.hibernate.t
14:04:28.072 restartedMain INFO .p.i.JtaPlatformInitiator initiateService 59 HHH000489: No JTA platform availabl
14:04:28.077 restartedMain INFO nEntityManagerFactoryBean buildNativeEntityManagerFactory 447 Initialized JPA EntityManager
14:04:28.552 restartedMain INFO .OptionalLiveReloadServer startServer 59 LiveReload server is running on por
14:04:28.578 restartedMain INFO c.c.o.OrmLearnApplication logStarted 59 Started OrmLearnApplication in 8.41
14:04:28.584 restartedMain INFO c.c.o.OrmLearnApplication main 16 Inside main
14:04:28.722 licationShutdownHook INFO nEntityManagerFactoryBean destroy 660 Closing JPA EntityManagerFactory fc
14:04:28.728 licationShutdownHook INFO c.z.h.HikariDataSource close 349 HikariPool-1 - Shutdown initiated..
14:04:28.812 licationShutdownHook INFO c.z.h.HikariDataSource close 351 HikariPool-1 - Shutdown completed.
```

Country table

```
mysql> use myDB;
```

```
mysql> create table country(  
  -> co_code varchar(2) primary key,  
  -> co_name varchar(50)  
  -> );
```

```
mysql> insert into country values  
  -> ('IN', 'India'),  
  -> ('US', 'United States of America');
```

```
mysql> select * from country;  
+-----+-----+  
| co_code | co_name |  
+-----+-----+  
| IN      | India   |  
| US      | United States of America |  
+-----+-----+  
2 rows in set (0.04 sec)
```

Country.java

```
package com.cognizant.orm_learn.model;  
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;
```

```
@Entity
```

```
@Table(name="country")
```

```
public class Country {
```

```
    @Id
```

```
    @Column(name="co_code")
```

```
    private String code;
```

```
    @Column(name="co_name")
```

```
    private String name;
```

```
    public String getCode() {
```

```
        return code;
```

```
    }
```

```
    public void setCode(String code) {
```

```
        this.code = code;
```

```
    }
```

```

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

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

```

CountryRepository.java

```

package com.cognizant.orm_learn.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.cognizant.orm_learn.model.Country;

@Repository
public interface CountryRepository extends JpaRepository<Country, String>{

}

```

CountryService.java

```

package com.cognizant.orm_learn.service;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.cognizant.orm_learn.model.Country;
import com.cognizant.orm_learn.repository.CountryRepository;
import jakarta.transaction.Transactional;

@Service
public class CountryService {

    @Autowired
    private CountryRepository countryRepository;

    @Transactional
    public List<Country> getAllCountries(){
        return countryRepository.findAll();
    }
}

```

OrmLearnApplication.java

```
package com.cognizant.orm_learn;
import java.util.List;
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 com.cognizant.orm_learn.model.Country;
import com.cognizant.orm_learn.service.CountryService;
```

@SpringBootApplication

```
public class OrmLearnApplication {
```

```
    private static final Logger LOGGER =
```

```
    LoggerFactory.getLogger(OrmLearnApplication.class);
```

```
    private static CountryService countryService;
```

```
    private static void testGetAllCountries() {
```

```
        LOGGER.info("Start");
```

```
        List<Country> countries = countryService.getAllCountries();
```

```
        LOGGER.debug("countries={}", countries);
```

```
        LOGGER.info("End");
```

```
    }
```

```
    public static void main(String[] args) {
```

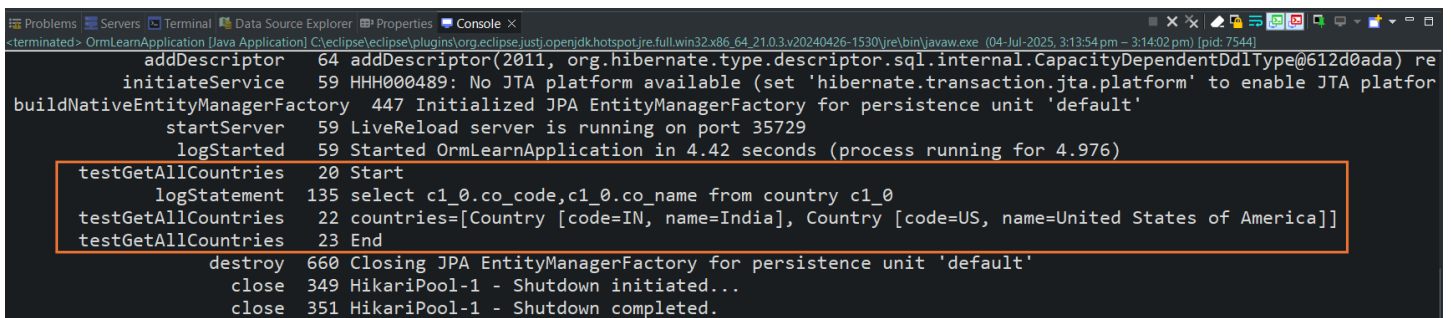
```
        ApplicationContext context = SpringApplication.run(OrmLearnApplication.class,
args);
```

```
        countryService = context.getBean(CountryService.class);
```

```
        testGetAllCountries();
```

```
    }
```

```
}
```



```
Problems Servers Terminal Data Source Explorer Properties Console X
terminated> OrmLearnApplication [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe (04-Jul-2025, 3:13:54 pm - 3:14:02 pm) [pid: 7544]
addDescriptor 64 addDescriptor(2011, org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@612d0ada) re
initiateService 59 HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platfor
buildNativeEntityManagerFactory 447 Initialized JPA EntityManagerFactory for persistence unit 'default'
startServer 59 LiveReload server is running on port 35729
logStarted 59 Started OrmLearnApplication in 4.42 seconds (process running for 4.976)
testGetAllCountries 20 Start
logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0
testGetAllCountries 22 countries=[Country [code=IN, name=India], Country [code=US, name=United States of America]]
testGetAllCountries 23 End
destroy 660 Closing JPA EntityManagerFactory for persistence unit 'default'
close 349 HikariPool-1 - Shutdown initiated...
close 351 HikariPool-1 - Shutdown completed.
```

Hands on 4

- **Difference between JPA, Hibernate and Spring Data JPA**

→ solution

i>Hibernate implementation:

hibernate.config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="hibernate.connection.password">*****</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/Employee</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="current_session_context_class">thread</property>
        <property name="show_sql">true</property>
        <property name="format_sql">true</property>
        <property name="hbm2ddl.auto">update</property>
        <mapping class="com.shom.employee_management.model.Employee" />
    </session-factory>
</hibernate-configuration>
```

Employee.java

```
package com.shom.employee_management.model;
import javax.persistence.Entity;
import javax.persistence.Id;
```

@Entity

```
public class Employee {
    @Id
    private Integer id;
    private String name;
    private String dept;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
}
```

```

    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDept() {
        return dept;
    }
    public void setDept(String dept) {
        this.dept = dept;
    }
}

```

EmployeeServiceHibernate.java

```

package com.shom.employee_management.service;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import com.shom.employee_management.model.Employee;

public class EmployeeServiceHibernate {
    private SessionFactory factory;
    public EmployeeServiceHibernate() {
        factory = new Configuration().configure().buildSessionFactory();
    }

    public void addEmployee(Employee employee) {
        Session session = factory.openSession();
        Transaction tx = null;
        try {
            tx = session.beginTransaction();
            session.save(employee);
            tx.commit();
        } catch (Exception e) {
            if (tx != null) tx.rollback();
            e.printStackTrace();
        } finally {
            session.close();
        }
    }
}

```

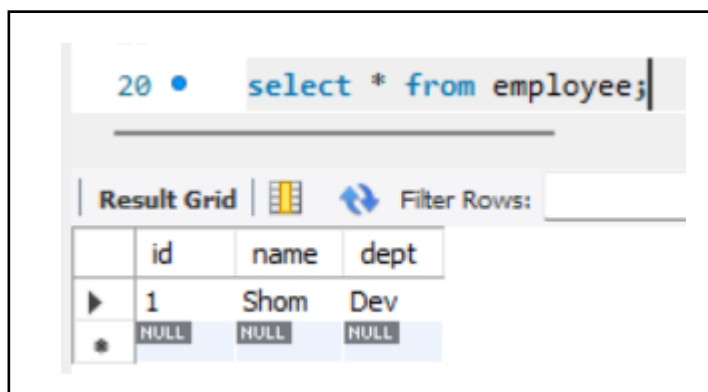
EmployeeManagementApplication.java

```
package com.shom.employee_management;  
import com.shom.employee_management.model.Employee;  
import com.shom.employee_management.service.EmployeeServiceHibernate;
```

```
public class EmployeeManagementApplication2 {  
    public static void main(String[] args) {  
        EmployeeServiceHibernate employeeServiceHibernate = new  
EmployeeServiceHibernate();  
        Employee employee = new Employee();  
        employee.setId(1);  
        employee.setName("Shom");  
        employee.setDept("Dev");  
  
        employeeServiceHibernate.addEmployee(employee);  
        System.out.println("Employee created");  
    }  
}
```

```
org.hibernate.internal.util.EntityPrinter - Listing entities:  
org.hibernate.internal.util.EntityPrinter - com.shom.employee_management.model.Employee{name=Shom, id=1, dept=Dev}
```

```
23:39:35.047 [main] DEBUG org.hibernate.SQL -  
insert  
into  
Employee  
(dept, name, id)  
values  
(?, ?, ?)  
Hibernate:  
insert  
into  
Employee  
(dept, name, id)  
values  
(?, ?, ?)  
23:39:35.256 [main] DEBUG org.hibernate.resource.jdbc.internal.LogicalConnectionManagedImpl - Initiating JDBC connection  
23:39:35.256 [main] DEBUG org.hibernate.resource.jdbc.internal.LogicalConnectionManagedImpl - Initiating JDBC connection  
Employee created
```



The screenshot shows a SQL IDE interface. At the top, a query editor contains the text "20 • select * from employee;". Below the editor, there is a toolbar with icons for "Result Grid", a table icon, and "Filter Rows:". The "Result Grid" is active, displaying a table with the following data:

	id	name	dept
▶	1	Shom	Dev
•	NULL	NULL	NULL

ii> Spring data jpa implementation:

application.properties

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.h2.console.enabled=true
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.show-sql=true
```

Employee.java

```
package com.shom.employee_management.model;
import javax.persistence.Entity;
import javax.persistence.Id;
```

@Entity

```
public class Employee {
    @Id
    private Integer id;
    private String name;
    private String dept;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDept() {
        return dept;
    }
    public void setDept(String dept) {
        this.dept = dept;
    }
}
```


EmployeeRepository.java

```
package com.shom.employee_management.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.shom.employee_management.model.Employee;

public interface EmployeeRepository extends JpaRepository<Employee, Integer>{

}
```

EmployeeService.java

```
package com.shom.employee_management.service;
import javax.transaction.Transactional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.shom.employee_management.model.Employee;
import com.shom.employee_management.repository.EmployeeRepository;

@Service
public class EmployeeService {
    @Autowired
    private EmployeeRepository employeeRepository;
    @Transactional
    public void addEmp(Employee emp) {
        employeeRepository.save(emp);
    }
}
```

EmployeeController.java

```
package com.shom.employee_management.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.shom.employee_management.model.Employee;
import com.shom.employee_management.service.EmployeeService;

@RestController
@RequestMapping("/employees")
public class EmployeeController {
    @Autowired
    private EmployeeService employeeService;
    @PostMapping
    public String addEmployee(@RequestBody Employee employee) {
        employeeService.addEmp(employee);
        return "Employee added successfully!";
    }
}
```

EmployeeManagement.java

```
package com.shom.employee_management;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com.shom.employee_management")
public class EmployeeManagementApplication {
    public static void main(String[] args) {
        SpringApplication.run(EmployeeManagementApplication.class, args);
    }
}
```

