Date : 09/01/2021
Spring Boot 9AM
Mr. RAGHU
---
javabyraghu@gmail.com
Spring Data JPA

ORM: Object Relational Mapping (Theory)

Before to ORM --> JDBC (Pure SQL based Programming).
Database Operations in OOPs -- Sun specification - JPA
JPA - Java Persistency API.

JPA Implementation: IBatis, TopLink, nHibernate, JBoss Hibernate*..etc

*) Programmer, we follow specification and write application.
That internally follow one implementation (Ex: JBoss Hibernate).

API Provider Sun/Oracle API name : JPA
API vendor Implementation: Hibernate
API Application : Our code
---

*) If we compare JDBC--Project and JPA--Project,
inside JPA Project No.of Lines of Code is less.
Even error rate also. JDBC , Manual SQL coding.

*) We can not write code using only JPA, internally one Impl must be there.

*) In JPA also some coding lines exist, still they are reduced into few lines, ie new Implementation is : Spring Data JPA.

*) Difference between JPA (with Hibernate) and Spring Data JPA
is no.of lines of code only. Spring Data JPA reduces coding lines.

Spring Data JPA:
a) Programmer need not to define code for basic Database operations
Just provide model class/entity class and PrimaryKey DataType.
[No need of wiriting code in high level]

b) Supports Embedded Databases : no download + no install required
Used only for development purpose only. Never use in Production.

=> H2**, HyperSQL(HSQL), Apache Derby.
[Embedded/InMemoryDatabase/RAM Database]

c) Custom Query Programming using : findBy, @Query(JPQL/HQL,SQL).

@Entity : It maps our class with Database table.

@Id : It indicates PrimaryKey.

@Table : To provide table details like name.

(it is optional. If we do not provide table name, default is classname)

@Column : To provide column details we can use this. Like column name. (it is optional. If we do not provide column name, default is variablename)

---code----------

Model class/Entity class

```
@Entity
@Table(name="emptab")
class Employee {
  @Id
  @Column(name="eid")
  Integer empId;
  @Column(name="ename")
  String empName;
  @Column(name="esal")
  Double empSal;
}
```

*) (new)Entity/Model / JavaBean/POJO (not using in Spring)
*) BO VO DTO (Legacy Design)- Strtus based design.

----code----------

#1 Create one Starter Project
Dep : Spring Web, lombok, Spring Data Jpa, H2.

#2. Model class

```
package in.nareshit.raghu.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

import lombok.Data;

@Data
@Entity
@Table(name="emptab")
public class Employee {
        @Id
        @Column(name="eid")
        private Integer empId;
        @Column(name="ename")
        private String empName;
        @Column(name="esal")
        private Double empSal;
}
```

#3 application.properties

```
server.port=9898
spring.jpa.show-sql=true
spring.h2.console.enabled=true
spring.datasource.url=jdbc:h2:mem:testdb
```

#4. Run Application and Enter URL as:
http://localhost:9898/h2-console
Click on Connect (if failed modify JDBC URL=jdbc:h2:mem:testdb)
