***Spring boot with database using ORM***

Object Relation mapping

Limitation of JDBC.

1. Using JDBC we can’t store object in database as well as can’t retrieve object from database. We need to convert our java object into sql format and vice-versa.
2. JDBC use SQL language. SQL is database dependent.
3. JDBC throw checked exception. The exception hierarchy is database dependent.
4. JDBC doesn’t support relationship ie is a as well has a.
5. In JDBC not mandatory JavaBean class must be map to table.

According to ORM.

Object Relation

Mapping

@Entity

@Table(name=”Employeedetails”) if table name is different

class Employee { Employeedetails🡪 Table

@Id PK

id

@Column(name=”ename”) id,ename,salary 🡪column

name

salary

}

Mapping 🡪 using

XML version

Annotation

Employee 🡨--🡪Employee

Id --- ID PK

Name -- NAME

Salary -- SALARY

JPA (Java Persistence API)

Hibernate

JPA is a type of EJB. JPA is technologies. Part of java people.

Hibernate is a framework. Part of jboss.

JPA provide specification as well as implementation.

Hibernate is base upon JPA. It provide implementation for JPA.

Core Java with Jdbc.

Core java with ORM tool like JPA or Hibernate

Servlet, JSP with ORM tool like JPA or Hibernate

Spring MVC with ORM tool like JPA or Hibernate inside dispatcher-servlet.xml we provide details as well as orm interaction details.

Spring Boot with ORM JPA (Spring boot doesn’t support directly hibernate API).

If we use Core JPA or Hibernate

Core ORM

JPA Hibernate

persistence.xml hibernate.cfg.xml

these file hold database information

EntityManagerFactory SessionFactory

Like connection in jdbc

EntityManager Session interface

Using some method we do directly operation on entity class.

Like Statement or PreparedStatement using query we do the operation on table.

Using JDBC by default auto commit. If we want to do commit or rollback. In connection object we need to set con.setAutoCommit(false). Then after DML operation con.commit() or con.rollback();

Through ORM it doesn’t auto commit by default. We need to use Transaction concept.

EntityTransaction Transaction

commit or rollback commit or rollback

persist() save

merge update

remove delete

find get

In Sprin boot we use JPA API. In Spring database details we provide in application.properties file.

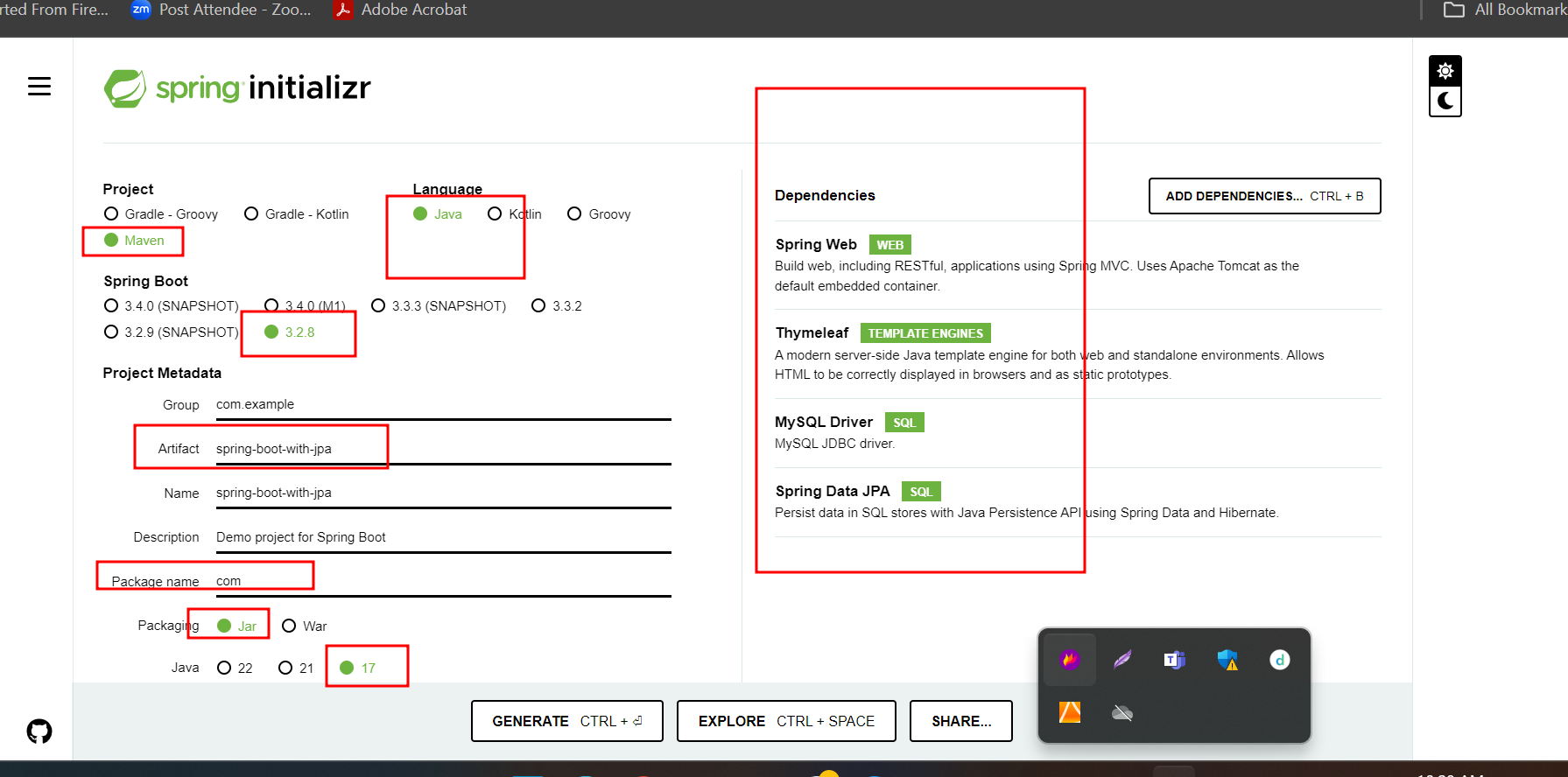
Spring boot with JPA

Web starter -🡪 Controller

Thymeleaft --🡪 for view

My SQL connector -🡪 MySQL database.

JPA starter --🡪 ORM tools



SQL

Select \* from employee; here employee is table name and not case sensitive. \* mean all column from a table.

Select eid,ename from Employee; id and name are column name.

In JPQ : JPQL

Select emp from Employee emp; where Employee is entity class name case sensitive. emp object, emp hold all variable details.

Select emp.eid,emp.ename from Employee emp, emp is object and eid is variable name

In Hibernate : HQL