Spring framework

What is DI and IOC

IOC : Inversion control

IOC is a programming design pattern. In place of creating any resources explicitly allow to create by container. If container create those resource it maintain properly. So base upon requirement we can pull it and use it. IOC is a concept.

DI : Dependency Injection : DI is an implementation of IOC.

Types of DI

1. Constructor base
2. Setter base

We need to configure

Using

1. Xml file configuration
2. Annotation base

Spring modules

Spring core

Spring context

Spring mvc model view controller

Spring dao data access object using core jdbc or jdbcTemplate

Spring orm object relation mapping : hibernate or jpa

Spring jpa data jdbc, hibernate , jpa, spring data etc.

Spring rest rest full api

Spring aop

Spring security

Spring cloud

Spring boot

Junit 5 testing

Spring testing

JPA : Java Persistence API .

JPA is part of Java technologies. JPA provide specification as well as implementation.

Hibernate is framework. Hibernate provide implementation for JPA. Hibernate is part of jBoss.

Core java with JPA/ hibernate

Spring framework with jpa/hibernate

Spring boot with JPA.

Spring boot with Spring JPA data.

In JPA inside persistence.xml file we need to provide database details like driverclasname,url,username and password.

hibernate.cfg.xml file

In JDBC first we need create table in database and then we can interact with database to

Store, delete, update and retrieve.

Using ORM

Object relation mapping

Class Product { Product table

Pid,pname,price variable Pid,pname,price etc

}

EntityManagerFactory : it is an interface which help to get the connection from persistence.xml file with help of Persistence class. It is like a Connection in JDBC.

EntityManager : it is an interface which provide set of method which help to do operation on entity class. It is like a statement or preparedstatement in jdbc.

EntityTransaction : it is an interface which help to do transaction.

Through JDBC if we do any DML(insert/delete/update) Operation. By Default JDBC use auto commit.

TCL : Transaction control language : commit, rollback and savepoint etc

Update account set amount = amount – 500 where accno=100

Update account set amount = amount + 500 where accno=101

Start transaction; in mysql

Then we can do any dml operation

Commit or rollback;

If we want transaction using JDBC.

con.setAutocommit(false); by default true

after DML operation

con.commit(); or con.rollback();

In JPA we need to use transaction while doing DML Operation. By default transaction off.