1. Why to learn advanced java?
2. What you understand by frontend?

What we see on website is frontend .

To build frontend we need Html , css , Js , Angular , react.

1. What is Backend ?

To process any request we need Backend.

To develop backend we need Spring boot ,Servlet , Hibernate , Jsp ,Java.

1. When we elaborate Request Response?

From frontend we send request to backend and from backend we receive response to frontend

1. Maven :-Used to manage dependencies in our project

Pom.xml-dependencie

Windows+ShowView+Problem-pom.Xml

Windows+preferences+XML+Apply and Close

**Hibernate Basic Setup:-**

Inserrt

Update

Delete

Select-Columns

Where

Spring-Angular

Crud operation by using built in methods

Criteria

Query

Projection

Hql

Mappings-one to many

**Spring Replacement-Servlet/Jsp**

**Angular -React**

**JDBC Steps-**

Connection-UserName,Password,Url

Statement

ResultSet

For/While

Column by Column Data fetching

---------------------------------------------------------------------------------------------------------------------------

**Hibernate Project Creation steps-**

>Table

>Pojo=plain old java object

>hibernate.cfg.xml

>clint-main

>hibernae jar

---------------------------------------------------------------------------------------------------------------------------

**Hibernate Load(Fetch the data from database);**

Configuration cfg=new Configuration();

Cfg.configuration();

Cfg.addAnnotedClass(car.class);

SessionFactory factory=cfg.buildSessionFactory();

Session session=factory.openSession();

Car car=session.load(car.class,1);

Syso(car);

---------------------------------------------------------------------------------------------------------------------------

**Errors till we faced -**

Cfg file put in src/java

In pom.xml hibernate dependency,mysql dependency

For unknown entity- @Entity,cfg.addAnnotatedClass(.class)

For Identifier-@Id

For Column-@column(name=”---”)

**Steps to Hibernate Setup-**

1. Create Maven Project
2. Hibernae dependency add in pom.xml
3. Mysql connector in pom.xml
4. Create table Car
5. Create package in project
6. Create pojo car- constructor,Getters/Setters,toString-Same name as column name
7. @Entity at Pojo
8. Client class ClientSelect.java

Configuration cfg=new Configuration();

Cfg.configure();

Cfg.addAnnotedClass(car.class);

SessionFactory factory=cfg.buildSessionFactory();

Session session=factory.openSession();

Car car=session.load(car.class,1);

Syso(car);

Hibernate.cfg.xml

<!-- Version 8 MySQL hiberante-cfg.xml example for Hibernate 5 -->

<hibernate-configuration>

<session-factory>

<property name=*"connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/test</property>

<property name=*"dialect"*>org.hibernate.dialect.MySQL8Dialect</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>root</property>

<property name=*"show\_sql"*>true</property>

<property name=*"format\_sql"*>true</property>

<property name=*"hbm2ddl.auto"*>update</property>

</session-factory>

</hibernate-configuration>

**Interview Questions:-**

1. **What is Use of load.-at atime you can fetch only one record using load method.**

----------------------------------------------------------------------------------------------

**Hibernate Crud Operation-Built-In-Methods**

**Interview Question:-**

1. What do you understand by ORM.
2. Do you know any ORM tool in java
3. Any other competitor or alternative for hibernate which is based on ORM.-Mybatis
4. Do you know any inbuilt method of hibernate.
5. In hibernate which methods perform this operation-Load/Insert/Update/Delete/Save/SaveorUpdate.
6. Which package you import while developing hibernate Example.-javax.persistence,org.hibernate.
7. Steps for executing any operation in hibernate.
8. Hibernate internally uses stmt or pstmt?-pstmt
9. Update-It works on primary key

-If exist then update

-if not exist then exception occurs

-Internally hibernate first select then check is data really need to be Update then it fires update query.

1. **Exception occurs in hibernate.**
2. **File not found exception/Configuration Exception**
3. **classNotFoundException**
4. **UnknownEntityTypeException/AnnotationException:No identifier Specified**
5. **HibernateException-Default Const**
6. **SqlSyntaxErrorException:Uknown=@column**
7. **ObjectNotFoundException:**

**Hibernate Criteria and Query**

**Limitation->only single record used to fetch**

**->No control on where clause**

**->No control on Select Clause**

**->Not able to select complete table**

**-In hibernate you never communicate directly with tables and column but you deals with class object and variables that’s why hibernate is OQM tool**

**OQM(Object relational model)**

**-In hibernate each row means one object.**

**Many rows list of object.**

**-In hibernate each column means one String**

**-Whenever you fire load method with same primary key repeatedly it will fire only one query as primary key is same this is because of hibernate use caching which is byDefault called as first cache which is at session level and we can not disable it.**

**-caching concept in hibernate-**

**Session-first level cache-**

**Hibernate Insert-**

Configuration cfg=**new** Configuration();

cfg.configure();

cfg.addAnnotatedClass(Car.**class**);

SessionFactory factory=cfg.buildSessionFactory();

//Car car=new Car(6,"Hyundai","Viru",600000);

Session session=factory.openSession();

Transaction transaction=session.beginTransaction();

Car c1=**new** Car(5,"Thar","Basanti",600000);

session.update(c1);

transaction.commit();

factory.close();

**Hibernate Update-**

Configuration cfg=**new** Configuration();

cfg.configure();

cfg.addAnnotatedClass(Car.**class**);

SessionFactory factory=cfg.buildSessionFactory();

//Car car=new Car(6,"Hyundai","Viru",550000);

Session session=factory.openSession();

Transaction transaction=session.beginTransaction();

Car c1=**new** Car(2,"Honda","Kaliya",800000);

session.Update(c1);//if primary key is present then update otherwise save.

transaction.commit();

factory.close();

**Hibernate Save-**

Configuration cfg=**new** Configuration();

cfg.configure();

cfg.addAnnotatedClass(Car.**class**);

SessionFactory factory=cfg.buildSessionFactory();

//Car car=new Car(6,"Hyundai","Viru",550000);

Session session=factory.openSession();

Transaction transaction=session.beginTransaction();

Car c1=**new** Car(7,"Swift","Gabbar",700000);

session.save(c1);

transaction.commit();

factory.close();

**Hibernate SaveorUpdate-**

Configuration cfg=**new** Configuration();

cfg.configure();

cfg.addAnnotatedClass(Car.**class**);

SessionFactory factory=cfg.buildSessionFactory();

//Car car=new Car(6,"Hyundai","Viru",550000);

Session session=factory.openSession();

Transaction transaction=session.beginTransaction();

Car c1=**new** Car(7,"Swift","Gabbar",700000);

session.saveorupdate(c1);

transaction.commit();

factory.close();

**Hibernate Delete-**

Configuration cfg=**new** Configuration();

cfg.configure();

cfg.addAnnotatedClass(Car.**class**);

SessionFactory factory=cfg.buildSessionFactory();

//Car car=new Car(6,"Hyundai","Viru",550000);

Session session=factory.openSession();

Transaction transaction=session.beginTransaction();

Car c1=**new** Car(2);

session.delete(c1);

transaction.commit();

factory.close();

**Hibernate Criteria And Query**

**Criteria:-(Select)**

**Configuration cfg=new Configuration();**

**cfg.addAnnotatedClass(Bike.class);**

**cfg.configure();**

**SessionFactory factory=cfg.buildSessionFactory();**

**Session session=factory.openSession();**

**Criteria criteria=session.~~createCriteria~~(Bike.class);**

**//Select \* from Bike**

**List<Bike> bickList=criteria.list();**

**for(Bike bike:bikeList) {**

**System.*out*.println(bike);**

**}**

**Criteria:-(Where)**

**Configuration cfg=new Configuration();**

**cfg.addAnnotatedClass(Bike.class);**

**cfg.configure();**

**SessionFactory factory=cfg.buildSessionFactory();**

**Session session=factory.openSession();**

**//Select \* from Bike**

**Criteria criteria=session.~~createCriteria~~(Bike.class);**

**//Where Condition**

**criteria.add(Restrictions.*eq*("bCompany","TATA"));**

**criteria.add(Restrictions.eq("bownerName","Viru")); criteria.add(Restrictions.eq("bId",3));**

**criteria.add(Restrictions.*gt*("cost",500000));**

**List<Bike> bikeList=criteria.list();**

**for(Bike bike:bikeList) {**

**System.*out*.println(bike);**

**}**

**Criteria Projection:-(Single column Select)**

**Configuration cfg=new Configuration();**

**cfg.addAnnotatedClass(Bike.class);**

**cfg.configure();**

**SessionFactory factory=cfg.buildSessionFactory();**

**Session session=factory.openSession();**

**Criteria criteria=session.~~createCriteria~~(Bike.class);**

**//Select \* from Bick**

**//Select Bick ownerName from bike**

**PropertyProjection projection1=Projections.*property*("bownerName");**

**PropertyProjection projection2=Projections.*property*("cost");**

**ProjectionList projection=Projections.*projectionList*();**

**projection.add(projection1);**

**projection.add(projection2);**

**criteria.setProjection(projection);**

**List<Object[]> bickList=criteria.list();**

**for( Object[] bick : bickList) {**

**for(Object objects:bick){**

**System.*out*.println(objects.toString());**

**}**

**}**

**Criteria Projection:-(Multiple column Select):-**

**Configuration cfg=new Configuration();**

**cfg.addAnnotatedClass(Bike.class);**

**cfg.configure();**

**SessionFactory factory=cfg.buildSessionFactory();**

**Session session=factory.openSession();**

**//Select \* from Bike**

**Criteria criteria=session.~~createCriteria~~(Bike.class);**

**//Select Bike ownerName from bike**

**Projection projection1=Projections.*property*("bownerName");**

**Projection projection2=Projections.*property*("cost");**

**ProjectionList projection=Projections.*projectionList*();**

**projection.add(projection1);**

**projection.add(projection2);**

**criteria.setProjection(projection);**

**List<Object[]> bikeList=criteria.list();**

**for (Object[] bike: bikeList) {**

**for (Object objects : bike) {**

**System.*out*.println(objects.toString());**

**}**

**}**

**Query:-(Select All)**

**Configuration cfg=new Configuration();**

**cfg.configure();**

**cfg.addAnnotatedClass(Bike.class);**

**SessionFactory factory=cfg.buildSessionFactory();**

**Session session=factory.openSession();**

**Query query=session.createQuery("from Bike");**

**List<Bike> bikeList=query.list();**

**for (Bike bike : bikeList) {**

**System.*out*.println(bike);**

**}**

**Query:-(Select Single)**

**Configuration cfg=new Configuration();**

**cfg.configure();**

**cfg.addAnnotatedClass(Bike.class);**

**SessionFactory factory=cfg.buildSessionFactory();**

**Session session=factory.openSession();**

**Query query=session.createQuery("select cost from Bike");//HQL**

**List<String> bikeList=query.list();**

**for (String bike : bikeList) {**

**System.*out*.println(bike);**

**}**

**Queries:-(Select Multiple)**

**Interview Question:-**

1. **Why to use Query Interface.**
2. **Why to use Criteria Interface.**
3. Is criteria/Query is interfaces or classes.
4. Which package you import for criteria or query.-org.hibernate
5. Why to use Projection.
6. What is package name for projection
7. What is syntax for creating a projection Object.
8. If simple object is created by projection then what will be return type.-List<String>
9. How to add projection into Criteria object.-criteria.setProjection(projection);
10. Can we use multiple projection? How to use?
11. When multiple projection is used rhe what is return type of list method?List<Object>
12. How to create query object?
13. How it is different from criteria?

**Hibernate Mapping**

**Significance of Dialect:-to generate Queries in particular grammer of database**

**If you want to use Oracle instead of mysql then where will be changes you do in your cfg.xml file:-**

**1. Connection Driver class**

**2. Dialect**

**3. Connection URl**

**Mapping:-**

**1.One to Many**

**2.One to One**

**1. One to Many:-**

**-------------------------------------------------------------**

**a. Database Perspective**

**Faculty Table FacultySubject Table**

**Fac\_ID int Primary Key; Sub\_Id int;**

**Fac\_Name varchar(50); Sub\_Name Varchar(50);**

**Fac\_ID int;-As Foreign key**

**--------------------------------------------------------------**

**b. Java Perspective**

**class Faculty{ class FacultySubject{**

**int fcId; int SubId;**

**String fcName; String SubName**

**List<FacultySubject> FSubList; }**

**}**

**--------------------------------------------------------------**

**Transaction**

Configuration cfg=**new** Configuration();

cfg.configure();

cfg.addAnnotatedClass(Faculty2.**class**);

SessionFactory factory=cfg.buildSessionFactory();

Session session=factory.openSession();

Transaction transaction=session.beginTransaction();

Faculty2 faculty=**new** Faculty2(11,"tka");

session.save(faculty);

transaction.commit();

**Interview Questions:-**

**1. Significance of format Sql**

**2. Significance of show\_Sql**

**3. Significance of Dialect-to generate a queries of particular format**

**4. How to comment In xml.**

**5. *hbm2ddl.auto-meaning***

***6. dtd-Document type definition-xml grammer***

***7. Hibernate update built in function give me some example to solve.***

***8. Boiler plate code***

***9. Why need Lombok dependency-No need of boiler plate code***

***10. Can you explain how Lombok works***

***11. What is one to many and one to one***

***12. Explain one to may and one to one at DB level and Java Level.***

***13. Can you Give me a Scenario where you used Refactoring.***

**Spring Boot**

**Spring initializer**

**Com.tka- Application class**

**Com.tka.entity-**

**Com.tka.controller-**

**Com.tka.service-**

**Com.tka.dao-**