XML configuration

Annotations

H

I

B

E

R

N

A

T

E

J

D

B

C

Java APp

DB

1. One time configuration -- Connecting to DB (Initial properties)
   * 1. hibernate.cfg.xml
     2. properties file
2. Class-Table mapping (ORM) (n)
   1. XXXX.hbm.xml or
   2. annotations

API --

1. pom.xml -- add hibernate dependencies
2. .cfg file or properites file
3. Mapping file / annotations

HQL – Hibernate Query Language

Eg : SQL – select \* from book

HQL -- from Book ; (Note : Book is Java class name)

Steps

1. Create a web app archetype maven project
2. Add dependencies to pom.xml
3. hibernate.cfg.xml
4. Create Login class username, password and role mapped to table usersnew
5. Add 1 login record for customer. (mysql interface)
6. Create table customer with customerid, name,email,phone,username and map to Customer Class
7. Validate user based on username and password.
8. If user is valid retrieves the customer details.

Class Relationships in hibernate

HQL

Casestudy

One to one

@OneToOne(cascade = CascadeType.ALL)

@JoinColumn(name="addressid")

private Address address;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "custId") // foreign key

private List<Complaint> compliants;

1. Create table customer with customerid, name,email,phone,username and map to Customer Class

usersnew

Username password role

Customernew one to one

Customerid name email phone username (fk)

1 sam [sam@egmail.com](mailto:sam@egmail.com) 888888 sam

Address onetomany

adid plotno street city state custid (fk)

1 101 church st blr KA 1  
2 304 Mount rd Chennai TN 1

PK -- @Id

FK – has a relation

Class Customer{

……

@OneToOne

Login login; // usersnew

}

Inheritance mapping – is a

Person --- (id,name) ------- 1,sam

Employee extends Person (dept) -- (2, peter,sales)

Customer extends Person (email) -- (3,john,john@gmail.com)

3 ways

1. Single table per hierarchy

Person

Id name dept email type (discriminator)

1 sam null null person

2 peter sales null emp

3 john null [john@gmail.com](mailto:john@gmail.com) cust

Pro : 1 table

Con : null values

1. Mapped SuperClass
2. Table with Join Strategy

Person

Id name

1. sam
2. peter

Employee

Dept type

sales 1

Case Study

1. Create a MVC web application using hibernate ORM for a Banking Domain. The use cases are as follows:
   1. Customer is validated with username and password.
   2. Customer can have multiple Accounts.
   3. Customer has name, email, phone, address
   4. Address has addId, plotNo,street,city , state.
   5. Account has
      1. AccountNo – autogenerated.
      2. accountType
      3. balance
      4. dateOfAccountCreation
   6. When Customer logs in all his accounts details are displayed.
   7. Customer can view his profile by clicking on MyProfile section.
   8. He/She can update password.
   9. He/She can click on forgot password to see a random password after confirming valid username with which he/she can login.
   10. Customer can open FD/RD account that will linked to the main account.