



Hibernate - IV

Hibernate 3 with annotations



Agenda

1

Hibernate 3 with annotations

Hibernate 3 with annotations



Objectives

At the end of this module, you will be able to

- Write a simple persistence class using annotations

Hibernate 3 With Annotations

Annotation	Description
@Entity	marks the class as an entity
@Table	Specifies the table name where data of the entity is to be persisted.
@Id	Marks the identifier of the entity
@Column	Specifies the details of column for the fields.

Note:

@Table annotation is not used , hibernate will use the class name as table name by default

@Column annotation is not used, hibernate will use the property name as column name by default

Persistence class

```
package com.wipro.example;
```

```
import javax.persistence.Entity;  
import javax.persistence.Id;
```

```
@Entity
```

```
@Table(name="student2")
```

```
public class Student {
```

```
    @Id
```

```
    @Column(name="studid")
```

```
    private int id;
```

```
    @Column(name="studname", length=25)
```

```
    String name;
```

```
    int fees;
```

Persistence class (Contd.).

```
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public int getId() {  
    return id;  
}  
public void setId(int id) {  
    this.id = id;  
}  
public int getFees() {  
    return fees;  
}  
public void setFees(int fees) {  
    this.fees = fees;  
}
```

Note:

Persistence class should be mapped with <mapping> tag in hibernate.cfg.xml file.
Ex: <mapping
class="com.wipro.exmaple.St
udent"/>

Hibernate Configuration File

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>

<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
<property name="connection.url">jdbc:oracle:thin:@localhost:1521:orcl</property>
<property name="connection.username">scott</property>
<property name="connection.password">tiger</property>
<property name="dialect">org.hibernate.dialect.OracleDialect</property>
<property name="show_sql">true</property>
<property name="hibernate.hbm2ddl.auto">update</property>
<mapping class="com.wipro.Student" />

</session-factory>
</hibernate-configuration>
```


Demo Class

```
package com.wipro.example;
```

```
import java.util.List;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
import org.hibernate.Transaction;
```

```
import org.hibernate.cfg.Configuration;
```

Demo Class (Contd.)

```
public class Demo {  
    public static void main(String[] args) {  
        Student s=new Student();  
        s.setId(1001);  
        s.setName("Harish");  
        s.setFees(5000);  
        SessionFactory sessionFactory = new  
            Configuration().configure().buildSessionFactory();  
        Session session = sessionFactory.openSession();  
        Transaction transaction = session.beginTransaction();  
        session.save(p);  
        transaction.commit();  
        System.out.println("Successfully saved");  
        session.flush();  
        session.close();  
    }  
}
```

Summary

At the end of this module, you were able to

- Write a simple persistence class using annotations



Thank You

