

Hibernate with Annotation

The hibernate application can be created with annotation. There are many annotations that can be used to create hibernate application such as @Entity, @Id, @Table etc.

Hibernate Annotations are based on the JPA 2 specification and supports all the features.

All the JPA annotations are defined in the javax.persistence.* package. Hibernate **EntityManager** implements the interfaces and life cycle defined by the JPA specification.

The core advantage of using hibernate annotation is that you don't need to create mapping (hbm) file. Here, hibernate annotations are used to provide the meta data.

Example to create the hibernate application with Annotation

There are 4 steps to create the hibernate application with annotation.

1. **Add the jar file for oracle (if your database is oracle) and annotation**
2. **Create the Persistent class**
3. **Add mapping of Persistent class in configuration file**
4. **Create the class that retrieves or stores the persistent object**

1) Add the jar file for oracle and annotation

For oracle you need to add **ojdbc14.jar** file. For using annotation, you need to add:

- **hibernate-commons-annotations.jar**
- **ejb3-persistence.jar**
- **hibernate-annotations.jar**

2) Create the Persistent class

Here, we are creating the same persistent class which we have created in the previous topic. But here, we are using annotation.

@Entity annotation marks this class as an entity.

@Table annotation specifies the table name where data of this entity is to be persisted. If you don't use @Table annotation, hibernate will use the class name as the table name by default.

@Id annotation marks the identifier for this entity.

@Column annotation specifies the details of the column for this property or field. If @Column annotation is not specified, property name will be used as the column name by default.

Employee.java

```
package com.javatpoint;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name= "emp500")
public class Employee {
    @Id
    private int id;
    private String firstName,lastName;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
```

```
    this.lastName = lastName;
}

}
```

3) Add mapping of Persistent class in configuration file

open the hibernate.cfg.xml file, and add an entry of mapping resource like this:

```
<mapping class="com.javatpoint.Employee"/>
```

Now the configuration file will look like this:

hibernate.cfg.xml

```
<?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="hbm2ddl.auto">create</property>
    <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
    <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
    <property name="connection.username">system</property>
    <property name="connection.password">oracle</property>
    <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>

    <mapping class="com.javatpoint.Employee"/>
</session-factory>

</hibernate-configuration>
```

4) Create the class that retrieves or stores the persistent object

In this class, we are simply storing the employee object to the database. Here, we are using the **AnnotationConfiguration** class to get the information of mapping from the persistent class.

```
package com.javatpoint.mypackage;

package com.javatpoint;

import org.hibernate.*;
import org.hibernate.cfg.*;

public class Test {
    public static void main(String[] args) {
        Session session=new AnnotationConfiguration()
            .configure().buildSessionFactory().openSession();

        Transaction t=session.beginTransaction();

        Employee e1=new Employee();
        e1.setId(1001);
        e1.setFirstName("sonoo");
        e1.setLastName("jaiswal");

        Employee e2=new Employee();
        e2.setId(1002);
        e2.setFirstName("vimal");
        e2.setLastName("jaiswal");

        session.persist(e1);
        session.persist(e2);

        t.commit();
        session.close();
        System.out.println("successfully saved");
    }
}
```

```
}
```

download this hibernate example (developed using Myeclipse IDE)

download this hibernate example (developed using Eclipse IDE)

[← prev](#)[next →](#)[Share 11](#)