

# Tapestry5HowToUseTapestryHibernate

This is a very simple tutorial explaining the use of Tapestry-Hibernate module. You can find the official documentation here:<http://tapestry.apache.org/tapestry5/tapestry-hibernate/conf.html>

you can download example project (tested in Eclipse with mvnecclipse) [hb.zip](#)

First I will assume that you have properly set up Tapestry, Hibernate-core, Hibernate-Annotation, MySQL in your system.

Following are steps:

## Step 1:

Create a working Tapestry project, you can follow the examples in the <http://tapestry.apache.org/tapestry5/tutorial1/first.html> or simply does the following:

```
mvn archetype:create -DarchetypeGroupId=org.apache.tapestry -DarchetypeArtifactId=quickstart -
DarchetypeVersion=5.0.5 -DgroupId=org.example -DartifactId=hb -DpackageName=org.example.hb

cd hb

mvn jetty:run
```

open <http://localhost:8080/hb/> in your browser, if you see 'hb Start Page' then proceed to step 2 otherwise fix the problems first. (if you want another port use: `mvn -Djetty.port=8080 jetty:run`)

## Step 2:

Switch to an IDE and import the maven project, if you are using IntelliJ you can use following command:

```
mvn idea:idea
```

open the project under your IDE and create a package under org.example.hb, name it 'entities', then create a Hello.java under it, copy the following source into Hello.java

### Hello.java

```
package org.example.hb.entities;

import javax.persistence.*;

@Entity
@Table(name="hello")
public class Hello {

    @Id
    @GeneratedValue
    private long id;
    private String message;

    public long getId() { return id;}

    private void setId(long id) { this.id = id; }

    public String getMessage() { return message;}

    public void setMessage(String message) { this.message = message;}
}
```

We can't compile this program yet as we need those hibernate's jar files.

## Step 3:

a)

This procedure depends on IDE used, the main goal is to make hibernate jar file available to our project, in my case I create a hibernateLib directory which contains following jar files:

```

antlr-2.7.6.jar
asm.jar
c3p0-0.9.1.jar
cglib-2.1.3.jar
commons-codec-1.3.jar
5.0.5.jar
commons-collections-2.1.1.jar
commons-logging-1.0.4.jar
jar
dom4j-1.6.1.jar
ejb3-persistence.jar
hibernate-annotations.jar
hibernate-commons-annotations.jar
hibernate3.jar
hsqldb.jar
javassist-3.4.ga.jar
jta.jar
junit-3.8.1.jar
log4j-1.2.9.jar
tapestry-core-5.0.5.jar
tapestry-hibernate-
tapestry-ioc-5.0.5.jar
testng-5.1-jdk15.

```

some of jar files are duplicates of Tapestry 5, you can exclude them, so find a way in your IDE to add those jar files to the project, then compile, if no error found proceed to Step 4.

b)

if maven is integrated into your IDE (mvnecclipse...) just add this to your pom

```

<dependency>
  <groupId>org.apache.tapestry</groupId>
  <artifactId>tapestry-hibernate</artifactId>
  <version>${tapestry-release-version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate</artifactId>
  <version>3.2.2.ga</version>
</dependency>
  <dependency>
    <groupId>c3p0</groupId>
    <artifactId>c3p0</artifactId>
    <version>0.9.0</version>
  </dependency>
<dependency>
  <groupId>geronimo-spec</groupId>
  <artifactId>geronimo-spec-jta</artifactId>
  <version>1.0-M1</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-annotations</artifactId>
  <version>3.2.1.ga</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>3.1.12</version>
</dependency>

```

## Step 4:

We need to setup a database in MySQL and create a link between our app and MySQL in this step. first use MySQL console to create a database called 'hb', then create a file 'hibernate.cfg.xml' under resources directory and copy following contents into it:

### hibernate.cfg.xml

```

<!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="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.url">jdbc:mysql://localhost/hb</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password"></property>
        <!-- comment following line during 2nd run -->
        <property name="hbm2ddl.auto">create</property>

        <!-- pool via c3p0 which knows how to reconnect to server and does it nicely-->
        <property name="connection.provider_class">org.hibernate.connection.C3P0ConnectionProvider<
/property>
        <property name="hibernate.c3p0.acquire_increment">1</property>
        <property name="hibernate.c3p0.idle_test_period">100</property> <!-- seconds -->
        <property name="hibernate.c3p0.max_size">10</property>
        <property name="hibernate.c3p0.max_statements">0</property>
        <property name="hibernate.c3p0.min_size">1</property>
        <property name="hibernate.c3p0.timeout">100</property> <!-- seconds -->

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

```

**Note:** for IntelliJ IDEA user, if you encounter a notification that 'New Hibernate Facet detected', just choose 'Remove Hibernate and don't detect Hibernate Facets in module', this will make our tutorial simple.

Try to compile and run the program, for running it's IDE specific, you need to configure something to run it under Tomcat. if everything is fine at this point, let's proceed to step 5 to use Tapestry-Hibernate feature

## Step 5:

Under 'pages' package, create a class called HbAction.java and copy following into the file, you don't have to create a html file for this class:

**HbAction.java**

```

package org.example.hb.pages;

import java.util.List;

import org.apache.tapestry.annotations.Inject;
import org.example.hb.entities.Hello;
import org.hibernate.Session;

public class HbAction {

    @Inject
    private Session _session;

    private Hello    current;

    @CommitAfter
    public void onAction() {
        Hello h = new Hello();
        h.setMessage("Hello World");
        _session.save(h);
    }

    public List getList() {
        return _session.createCriteria(Hello.class).list();
    }

    public Hello getCurrent() {
        return current;
    }

    public void setCurrent(Hello current) {
        this.current = current;
    }

}

```

Also add HbAction.htm in webapp/WEB-INF/

```

<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_0_0.xsd">
  <head>
    <title>Hibernate example app</title>
  </head>
  <body>
    <t:loop source="list" value="current">
      ${current.message}<br/>
    </t:loop>
    <t:ActionLink>add</t:ActionLink>
  </body>
</html>

```

Compile and deploy the application:

<http://localhost:8080/HbAction>

**Note:** depends on server setup, it might be <http://localhost:8080/hb/HbAction>, adjust it if needed.

It will show a link "add" in browser, that's good, if you encounter some exceptions, that usually means those jar files are not properly set up, fix it first.

Now, go back to IDE and look for the hibernate.cfg.xml, comment the line out:

```
<!-- <property name="hbm2ddl.auto">create</property> -->
```

This will avoid Hibernate to drop and re-create the tables in MySQL so that we can retrieve what we have just added, re-compile, re-deploy the app, then try this command:

<http://localhost:8080/HbAction>

try to click "add" link multiple times, you will see multiple "Hello World" in the browser. this ends this simple tutorial.

## Credits:

From a very simple explanation(<http://tapestry.apache.org/tapestry5/tapestry-hibernate/conf.html>) I started learning how to use Tapestry-Hibernate module, I was thinking it might just take a short while to learn, however it took me whole day to finally understand it with the helps of people in the Tapestry Forum, particularly Davor Hrg and Marcus Schmidke-2. After that, I came back to that simple explanation and wondering why it took me so long, the documentation has sufficiently explained the module, I believe a sample program will help in this learning process, with the encouragement of Davor I finally sit down and wrote a wiki first time in my life 😊 hope other newbies will find this tutorial informative and help them reduce the time spent, actually all informations here are from postings of Davor and Marcus, my job is to put them together, every step in this tutorial has been tested, credits go to two of them and others who have helped me, any mistake are mine, Thanks.

You can find the original discussions here:<http://www.nabble.com/T5%3A-Hibernate-tf4409684.html#a12605784>

## Few more advanced things to do

After this basic setup, we can create a very simple form (BeanEditform is explained at <http://tapestry.apache.org/tapestry5/tutorial1/forms.html>)

## Updates

10/4/2008, added @CommitAfter on the onAction method. Without it, a commit never occurs and the new Hello object is never persisted.