**Assisted Practice: 3.1 Configuration of Hibernate**

This section will guide you to:

* Set up Eclipse to work with Hibernate
* Create an HTML page to call a servlet
* Create a servlet that will initialize Hibernate and then close it

**Development Environment**

* Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
* Apache Tomcat Server v9.0
* JRE: OpenJDK Runtime Environment 11.0.2
* Hibernate for Java 5.2.1
* MySQL Connector for Java 8.0.16
* JTA v 1.1
* Java XML Bind (no version)
* JAXB OSGI v.2.4.0
* Java Activation (no version)

This lab has twelve subsections, namely:

* + 1. Creating a dynamic web project
    2. Adding the jar files for Hibernate and its dependencies
    3. Configuring Hibernate with hibernate.cfg.xml
    4. Creating an HTML page index.html
    5. Creating a HibernateUtil class to initiate Hibernate in code
    6. Creating an InitHibernate servlet
    7. Configuring web.xml
    8. Checking for servlet-api.jar
    9. Building the project
    10. Publishing and starting the project
    11. Running the project
    12. Pushing the code to your GitHub repositories

**Step 3.1.1:** Creating a dynamic web project

* Open Eclipse
* Go the **File** menu. Choose **New->Dynamic Web Project**
* Enter the project name as **HibernateConfig**. Click on **Next**
* Enter nothing in the next screen and click on **Next**
* Check the checkbox **Generate web.xml deployment descriptor** and click on **Finish**
* This will create the project files in the Project Explorer

**Step 3.1.2:** Adding the jar files for Hibernate and its dependencies

* Hibernate.jar file is already present in your practice lab. (Refer FSD: Lab guide - Phase 2)
* Take **hibernate.jar** from folder mentioned in the lab guide for phase 2 and add it to your project’s **WebContent/WEB-INF/lib** folder
* mysql-connector-java.jar file is present in your practice lab. (Refer FSD: Lab Guide - Phase 2)
* Take **mysql-connector-java.jar file** from the folder mentioned in the lab guide for phase 2 and add it to your project’s **WebContent/WEB-INF/lib** folder
* Go to <http://www.java2s.com/Code/Jar/j/Downloadjta11jar.htm>
* Click on **jta-1\_1.jar.zip** link to download it
* Extract **jta-1\_1.jar** from it and add it to your project’s **WebContent/WEB-INF/lib** folder
* Go to <http://www.java2s.com/Code/Jar/j/Downloadjavaxxmlbindjar.htm>
* Click on **javax.xml/javax.xml.bind.jar.zip** link to download it
* Extract **javax.xml.bind.jar** from it and add it to your project’s **WebContent/WEB-INF/lib** folder
* Go to <https://jar-download.com/artifacts/com.sun.xml.bind>
* Click on the button **Download jaxb-osgi.jar** to download it
* Extract **jaxb-osgi-2.4.0-b180830.0438.jar** from it and add it to your project’s **WebContent/WEB-INF/lib** folder

**Step 3.1.3:** Configuring Hibernate with hibernate.cfg.xml

* In the Project Explorer, expand **HibernateConfig->Java Resources**
* Right click on **src** and choose **New->Other**
* Select **General->File** and click **Next**
* In filename, enter **hibernate.cfg.xml** and click on **Finish**
* Enter the following code:

<?xml version='1.0' encoding='utf-8'?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

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

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

<property name="connection.username">mysql user id</property>

<property name="connection.password">mysql password</property>

</session-factory>

</hibernate-configuration>

**Step 3.1.4:** Creating an HTML page index.html

* In the Project Explorer, expand the project **HibernateConfig**
* Expand **WebContent**. Right click on **WebContent**. Choose **New->HTML File**
* Enter the filename as **index.html** and click on **Finish**
* Enter the following code:

<!DOCTYPE html>

<**html**>

<**head**>

<**meta** charset="UTF-8">

<**title**>Hibernate Configuration Example</**title**>

</**head**>

<**body**>

<**a** href="init">Initialize Hibernate</**a**><**br**>

</**body**>

</**html**>

* Click on the **Save** icon

**Step 3.1.5:** Creating a HibernateUtil class to initiate Hibernate in code

* In the Project Explorer, expand **HibernateConfig->Java Resources**
* Right click on **src** and choose **New->Class**
* In Package Name, enter **com.ecommerce** and in **Name** enter **HibernateUtil** and click on **Finish**
* Enter the following code:

**package** com.ecommerce;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** HibernateUtil {

**private** **static** **final** SessionFactory sessionFactory;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

sessionFactory = metaData.getSessionFactoryBuilder().build();

} **catch** (**Throwable** th) {

**throw** **new** **ExceptionInInitializerError**(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** sessionFactory;

}

}

**Step 3.1.6:** Creating an InitHibernate servlet

* In the Project Explorer, expand **HibernateConfig->Java Resources**
* Right click on **src** and choose **New->Servlet**
* In **Class Name,** enter **InitHibernate** and click on **Finish**
* Enter the following code:

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.**\***;

**import** javax.xml.bind.**\***;

**import** java.io.Serializable;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

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

**import** com.ecommerce.EProduct;

**import** com.ecommerce.HibernateUtil;

/\*\*

**\*** Servlet implementation class InitHibernate

\*/

**@WebServlet("/InitHibernate")**

**public** **class** InitHibernate **extends** HttpServlet {

**private** **static** **final** long serialVersionUID = 1L;

/\*\*

**\*** **@see** HttpServlet**#**HttpServlet()

\*/

**public** InitHibernate() {

**super**();

// TODO Auto-generated constructor stub

}

/\*\*

**\*** **@see** HttpServlet**#**doGet(HttpServletRequest request**,** HttpServletResponse response)

\*/

**protected** void doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, **IOException** {

// TODO Auto-generated method stub

**try** {

**PrintWriter** out = response.getWriter();

out.println("<html><body>");

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

out.println("Hibernate Session opened.<br>");

session.close();

out.println("Hibernate Session closed.<br>");

out.println("</body></html>");

} **catch** (**Exception** ex) {

**throw** ex;

}

}

/\*\*

**\*** **@see** HttpServlet**#**doPost(HttpServletRequest request**,** HttpServletResponse response)

\*/

**protected** void doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, **IOException** {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**Step 3.1.7:** Configuring web.xml

* In the Project Explorer, expand **HibernateConfig->WebContent->WEB-INF**
* Double click on **web.xml** to open it in the editor
* Enter the following script:

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">

<display-name>HibernateConfig</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>InitHibernate</servlet-name>

<servlet-class>InitHibernate</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>InitHibernate</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

**Step 3.1.8:** Checking for servlet-api.jar

* Before building the project, we need to add **servlet-api.jar** to the project
* Servlet-api.jar file is already present in your practice lab. (Refer FSD: Lab Guide - Phase 2)
* To add it to the project, follow the below mentioned steps:
  + In the Project Explorer, right click on **HibernateConfig** and choose **Properties**
  + Select **Java Build Path** from the options on the left
  + Click on **Libraries** tab on the right
  + Under **ClassPath,** expand the node that says **Apache Tomcat**
  + If there is an existing entry for **servlet-api.jar,** then click on **Cancel** and exit the window
  + If it is not there, then click on **Classpath** entry and click on **Add External JARs** button on the right
  + From the file list, select **servlet-api.jar** file and click on **Ok**
  + Click on **Apply and Close**

**Step 3.1.9:** Building the project

* From the **Project** menu at the top, click on **Build**
* If any compile errors are shown, fix them as required

**Step 3.1.10:** Publishing and starting the project

* If you do not see the **Servers** tab near the bottom of the IDE, go to **Window** menu and click **Show View->Servers**
* Right click on the **Server** entry and choose **Add and Remove**
* Click the **Add** button to move **HibernateConfig** from the **Available** list to the **Configured** list
* Click on **Finish**
* Right click on the **Server** entry and click on **Publish**
* Right click on the **Server** entry and click on **Start**
* This will start the server

**Step 3.1.11:** Running the project

* To run the project, open a web browser and type: [**http://localhost:8080/**](http://localhost:8080/ServletConcept)**HibernateConfig**

**Step 3.1.12:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**