**Web Development Project**

* **How to create new dynamic web project in MyEclipse**

Using following steps create dynamic web project:

* Click on file
* Click new
* Select dynamic web project or search dynamic web project in others.
* Click next and named your project.
* Click next, next, generate index and xml if you want to
* And finally click on finish.
* **How to add Jars and TLD in MyEclipse Project**
* Create lib folder in WEB-INF, and paste all copied jars that are required.(download jars from the internet)
* Create tld folder and paste all necessary copied tld.
* **How to configure Apache Tomcat Server.**
* Right click on Server’s Console. Select configure server connector
* Select tomcat and click on configure tomcat version you want to use
* Click on enable server and browser your tomcat folder.
* Click apply and ok.
* Run the server.

Now start project, copying your js, css and others folders of html template in your web root directory. Create jsp folder under WEB-INF, copy all html files. Convert all your .hml extensions files to jsp. (You can rename your jsp files)

Web.xml

<display-name>To display Project Name</display-name>

<welcome-file-list>

<welcome-file>welcome file</welcome-file>

</welcome-file-list>

<filter>

<filter-name>your project name or you can give any</filter-name>

<filter-class>org.apache.Struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name> your project name or you can give any </filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

* **Deploy Project**

Deploy and undeploy J2EE projects by selecting project deployments.

* Select project, click on add.
* Add server.
* And finish it.

**JRE unbound error:-**

* Right click on project
* Go to build path
* Select configure build path
* Go to libraries and select unbound JRE and remove it.
* Click on add library and select JRE system library
* Select next and select alternate JRE
* Click finish and ok.

**java.lang.UnsupportedClassVersionError: com/estimate/action/Est\_EstimateAction : Unsupported major.minor version 51.0**

This problem may appear even if you have a jdk which matches JVM which you are trying it into.

* Project -> Properties -> Java Compiler
* Enable project specific settings.
* Then select Compiler Compliance Level to 1.6 or 1.5, build and test your app.
* Now, it should be fine.( https://javarevisited.blogspot.com/2015/08/how-to-solve-unsupported-majorminor-version-51-java.html)
* **Generate Random number in java**

<%@ page import=”java.util.Random” %><% Random r=new Random(); %>

Value=”<%= r.nextInt(100);

* **How to get System Date in input Field**

SimpleDateFormat sd=new SimpleDateFormat();

systemDate=sd.format(new Date()); //java class

public String systemDate; // variable with getter and setter

Always import jQuery (add jQuery Library) when you use jQuery.

Errors of jQuery: $ undefined, function undefined

<script src=”app-assets/js/core/libraries/jquery.min.js"></script>

Sql grammar Exception: check query syntax, misspelled letters or semicolon or any other user mistakes.

https://struts.apache.org/security/

http://api.jquery.com

https://datatables.net/

org.hibernate.NonUniqueObjectException: a different object with the same identifier value was already associated with the session:

accepted

Your problem is that the object you previously loaded still exists in your hibernate session. I see two ways to cope with that.

1. tell the hibernate session to merge your modified object with the one in the session

session.merge(object)

1. kick the old object out of the session before writing the updated object to the session. session.clear() might work.

org.hibernate.TypeMismatchException: Provided id of the wrong type. Expected: class java.lang.Integer, got class java.lang.String

You are providing String instead of correct type (Integer).

password encryption decryption using base64 in java

// Encode data on your side using BASE64

byte[] bytesEncoded = Base64.encodeBase64(str.getBytes());

System.out.println("encoded value is " + new String(bytesEncoded));

// Decode data on other side, by processing encoded data

byte[] valueDecoded = Base64.decodeBase64(bytesEncoded);

System.out.println("Decoded value is " + new String(valueDecoded));

JSON plugin tutorial:-

<https://struts.apache.org/plugins/json/>

[WARNING] The requested profile "pom.xml" could not be activated because it does not exist.

The warning message

[WARNING] The requested profile "pom.xml" could not be activated because it does not exist.

means that you somehow passed -P pom.xml to Maven which means "there is a profile called pom.xml; find it and activate it". Check your environment and your settings.xml for this flag and also look at all <profile> elements inside the various XML files.

Usually, mvn help:effective-pom is also useful to see what the real POM would look like.

Now the error means that you tried to configure Maven to build Java 8 code but you're not using a Java 8 runtime. Solutions:

1. Install Java 8
2. Make sure Maven uses Java 8 if you have it installed. JAVA\_HOME is your friend
3. Configure the Java compiler in your pom.xml to a Java version which you actually have.

Related:

* [Introduction to Build Profiles](http://maven.apache.org/guides/introduction/introduction-to-profiles.html)

Hibernate Exception

Caused by: org.hibernate.hql.internal.ast.QuerySyntaxException: USER is not mapped

. When doing a request to the database(in my case it Apache Derby), name of base need write the first letter upper case other in lower case.

This is wrong query:

session.createQuery("select first\_name from CUSTOMERV").

This is valid query

session.createQuery("select first\_name from Customerv").

org.hibernate.QueryException: Named parameter [y] not set

Designing Section Solutions

<sup><i class="fa fa-asterisk text-danger "></i></sup> //for mandatory fields

 data-keyboard="false" data-backdrop="static"// modal

contact number validation regex

oninput=*"this.value=this.value.replace(/[^0-9]/g,'');"*

One does not simply redirect using jQuery

jQuery is not necessary, and **[window.location.replace(...)](https://developer.mozilla.org/en-US/docs/Web/API/Location/replace)** will best simulate an HTTP redirect.

window.location.replace(...) is better than using window.location.href, because replace()does not keep the originating page in the session history, meaning the user won't get stuck in a never-ending back-button fiasco.

If you want to simulate someone clicking on a link, use **location.href**

If you want to simulate an HTTP redirect, use **location.replace**

**For example:**

// similar behavior as an HTTP redirect

window.location.replace("http://stackoverflow.com");

// similar behavior as clicking on a link

window.location.href = "http://stackoverflow.com";

To get the value of the **selected** radioName item of a form with id myForm:

$('input[name=radioName]:checked', '#myForm').val()

Here's an example:

$('#myForm input').on('change', function() {

alert($('input[name=radioName]:checked', '#myForm').val());

});

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>

<form id="myForm">

<input type="radio" name="radioName" value="1" /> 1 <br />

<input type="radio" name="radioName" value="2" /> 2 <br />

<input type="radio" name="radioName" value="3" /> 3 <br />