

Due Monday June 4th, 2012

If you did not do a previous tutorial, then create a web application named Tutorials in NetBeans. Review the web page that contains instructions on creating a servlet in a NetBeans project: Creating a Project in NetBeans: <http://users.cis.fiu.edu/~downeyt/cgs4825/netbeans.html>

All previous Tutorials should still be able to be accessed and run from the application.

If you had errors in any of the previous tutorials, fix them, so that you do not lose points again for old mistakes.

There are more detailed instructions at <http://bytesizebook.com/?/book/tutorials/chapter4.html> for the tutorial on the book's web site.

It is essential that you complete Tutorial 4, if you plan to complete any of the remaining assignments. Even if you do not get Tutorial 4 working, please submit it so that I can give you feedback on getting it to work.

Implement the Enhanced Controller in the Tutorials web application. It must work exactly like the example from Chapter 4 of the book.

- Place the Controller.java and ControllerHelper.java in the ch4.enhanced package in the Source Packages folder in NetBeans.
- As in Tutorial 3, place the RequestDataDefault.java file in the ch3.defaultValidate package in the Source Packages folder in NetBeans.
- Place the .jsp files in the ch4.enhanced package in the Source Packages folder in NetBeans.
- Use the JSPs from the Reorganised controller, but copy them into the ch4.enhanced package. If you did not do Tutorial 3, then review the instructions in it for creating the JSP files. Place all three JSPs in this directory.
- Remove all the hidden fields from the JSPs. You don't need them anymore, since all the data is saved in the session. Please note that you are NOT placing these in the Web Pages folder.
- You may copy the JSPs from Tutorial 3 or you may create new ones. If you create new ones, name them so that the names match the names in the ControllerHelper class. Names are case-sensitive. Be sure that the data is accessed through the bean that is stored in the helper. you will not be able to use a NetBeans template, as this will create the JSPs in the wrong folder. Instead, create an empty file and copy the JSP contents into it.
- Create a new package named shared in the Source Packages folder. Place the following in the package.
InitLog4j.java
HelperBaseCh4.java
ButtonMethod.java
- Download the following JAR files to a directory on your system. In NetBeans, right click the Libraries folder and add each of the downloaded JAR files to the web application.
log4j-1.2.11.jar
commons-collections-2.1.1.jar
commons-logging-1.0.4.jar
commons-beanutils.jar
- In the Web Pages folder, open the WEB-INF folder. Create a subdirectory of WEB-INF named logs. Create a file in the logs folder named error.log.
- Modify the web.xml file so that it has a servlet definition and a servlet mapping for the servlet controller. Map the controller to the URL /ch4/enhanced/Controller.
- Create a servlet definition for the InitLog4j servlet.
- Add the load-on-startup tag, so that the servlet will be loaded every time Tomcat is restarted. Do not create a servlet-mapping for the servlet.
- Add a hypertext link in index.jsp to the controller.
- Be sure that the index.jsp uses the strict DOCTYPE.
- Use a relative link from the current location to the controller. Use the url-pattern from the web.xml, but omit the leading /.
- All the previous Tutorials should also have links from this page. All of them should still work.
- When you are done with everything, your directory and file structure should look like this.

- Take a look at your log file, you will see that there are messages in it.

- Open the Files tab in NetBeans and navigate to build -> web -> WEB-INF -> logs
- Open the error.log file. It will look something like this.

Submitting Homework

- Be sure that your NetBeans project is configured so that .java files are added to the WAR file: Adding .java files to a WAR file
- Clean and Build your web application. After doing this, navigate to the WAR file in NetBeans and verify that the .java files are included: View WAR File
- Since you are using a log file that is in the web application, you will need to shut down Tomcat before you can do a Clean and Build.
- Open the Services folder and open the Servers folder. Right-click the Tomcat server and select Stop. Now you can do a Clean and Build.
- Email me the WAR files and bring the print outs to class. Due at the beginning of class.

=====

NOTE: (DO NOT DO THE FOLLOWING)

At this time we can not do the following instructions because we have not access to ocelot. They are JUST for your infomation as to what else can we do with this program.

- In the operating system (not in NetBeans) navigate to the dist folder in the NetBeans project.
- Deploy the WAR file on ocelot:
Deploy a WAR File. Be sure to change the permission of the error.log on ocelot to 606.
- In the index.jsp file in the ROOT web app on ocelot, add a hypertext link to the Tutorials web app. Use a relative reference.
- Upload the WAR file to me: Submitting homework on-line via the Web.