James Quach

Elizabeth O’neil

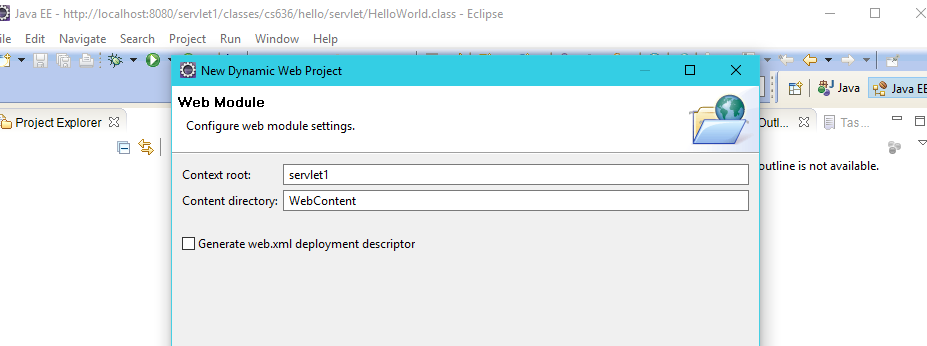
CS436

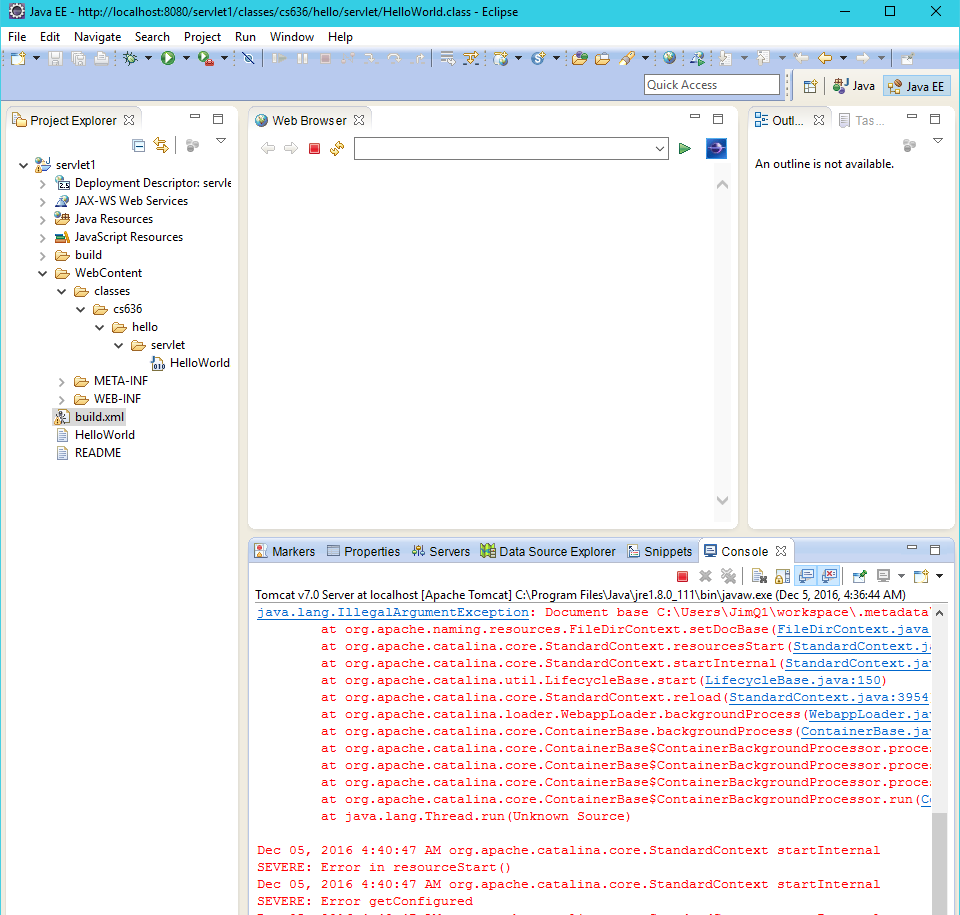
12/4/2016

**CS636 HW5:  MVC webapps, using eclipse tomcat, pizza3**

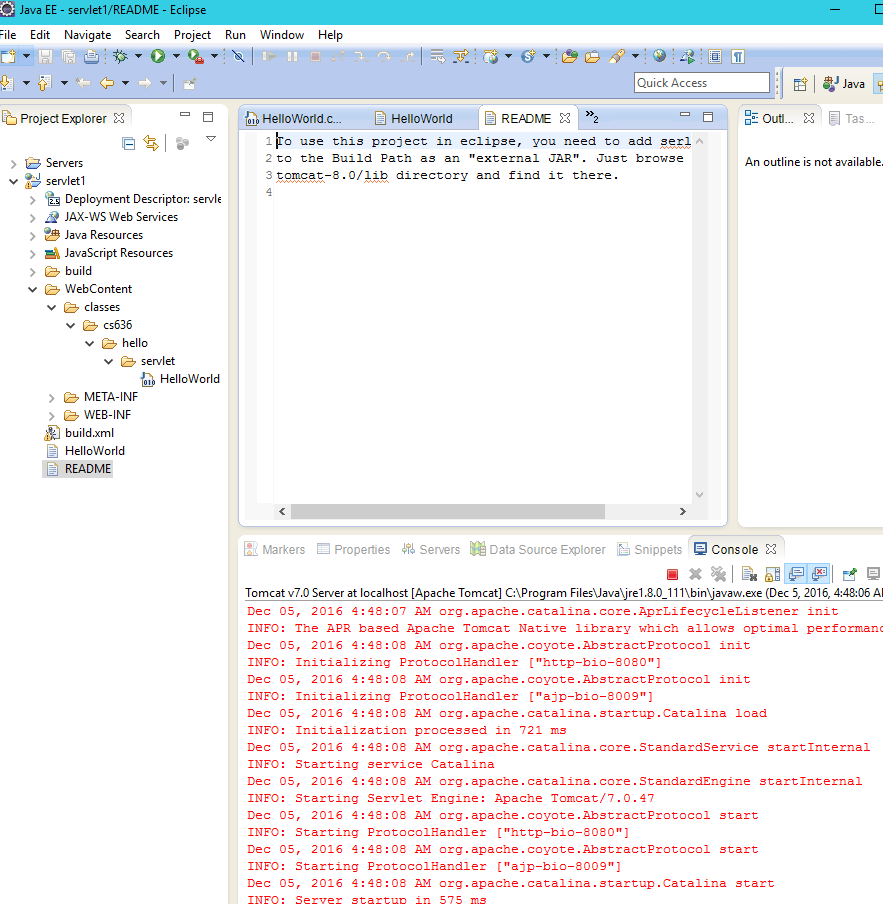
1. Hopefully you already have tomcat working from inside eclipse; if not, use the directions in [Home tomcat setup](http://www.cs.umb.edu/cs636/TomcatSetupForHome.html).

a.   Set up the $cs636/servlet1 project as a Dynamic Web Project in eclipse, if you haven't already. Report on success or any problems. Deploy the servlet1 project to tomcat running under eclipse. You can use "ant deploy" from eclipse for this, or use the Servers window to get eclipse to install it. Be sure to uncheck "Generate web.xml deployment descriptor" on the Web Module pane of the Dynamic Web Project wizard, so that the supplied web.xml will be used. Do this for any supplied web app.

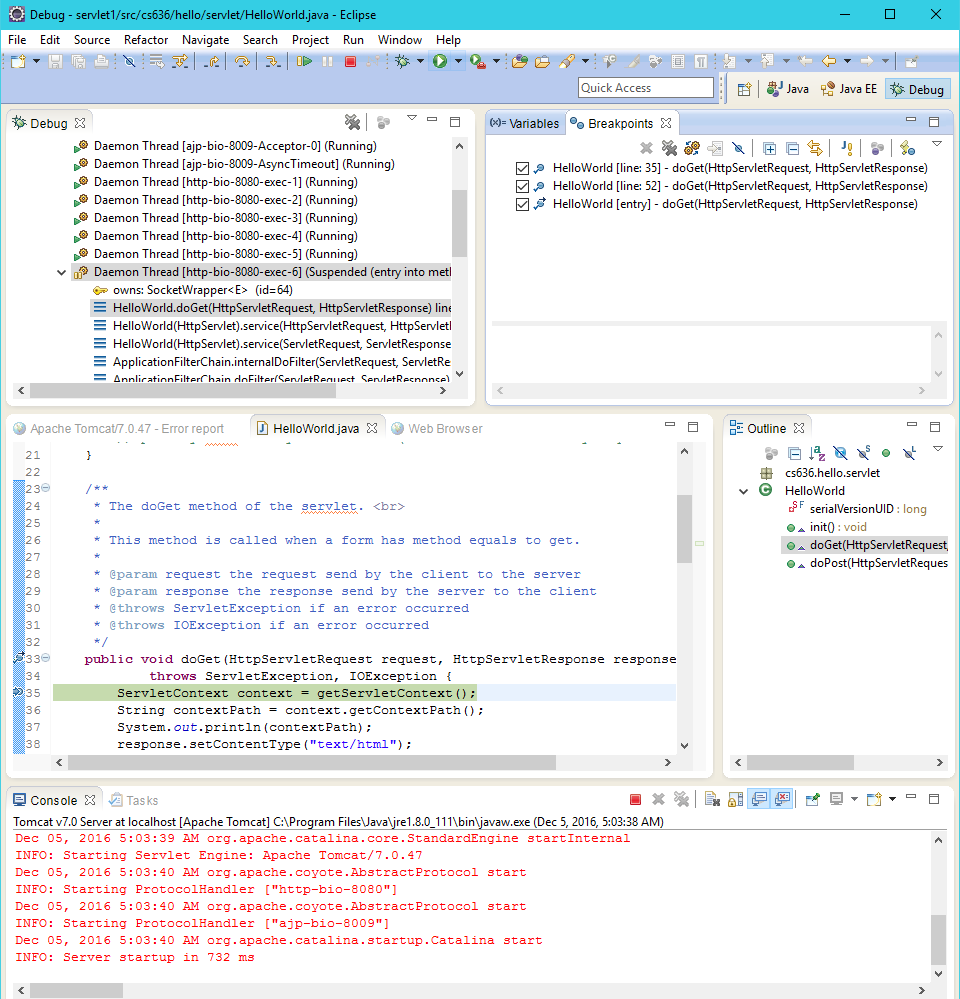
Servlet1 set as a dynamic web project below



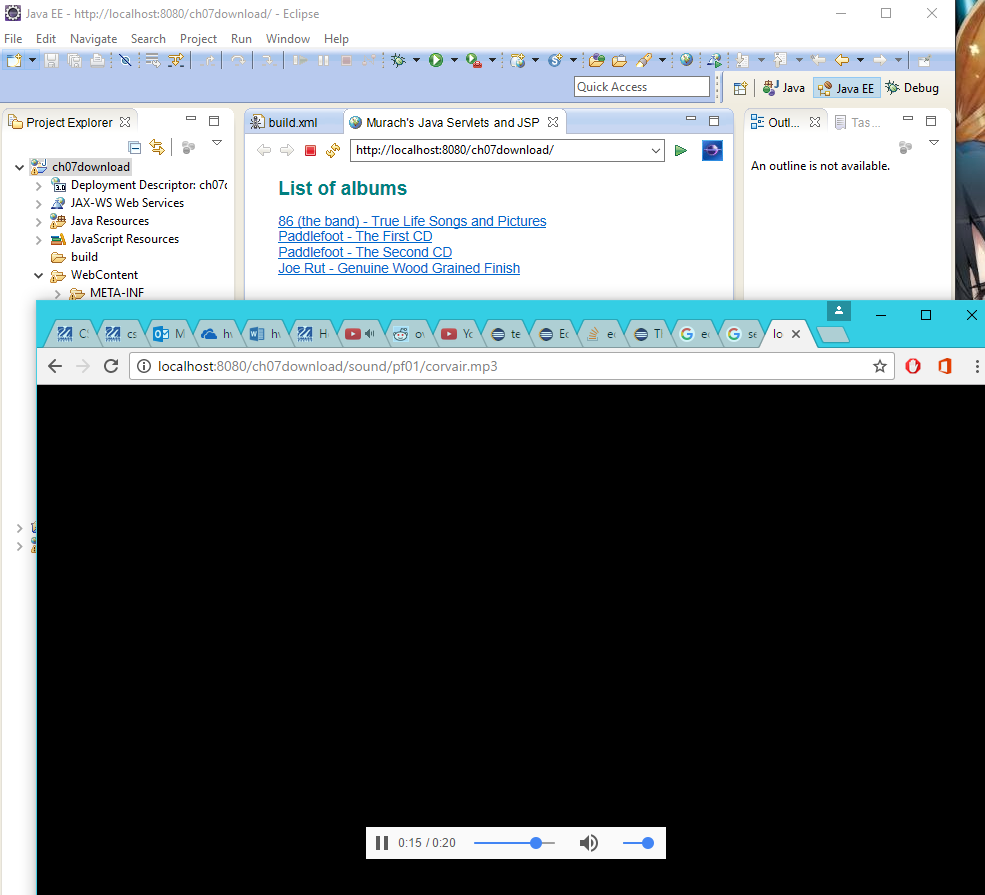
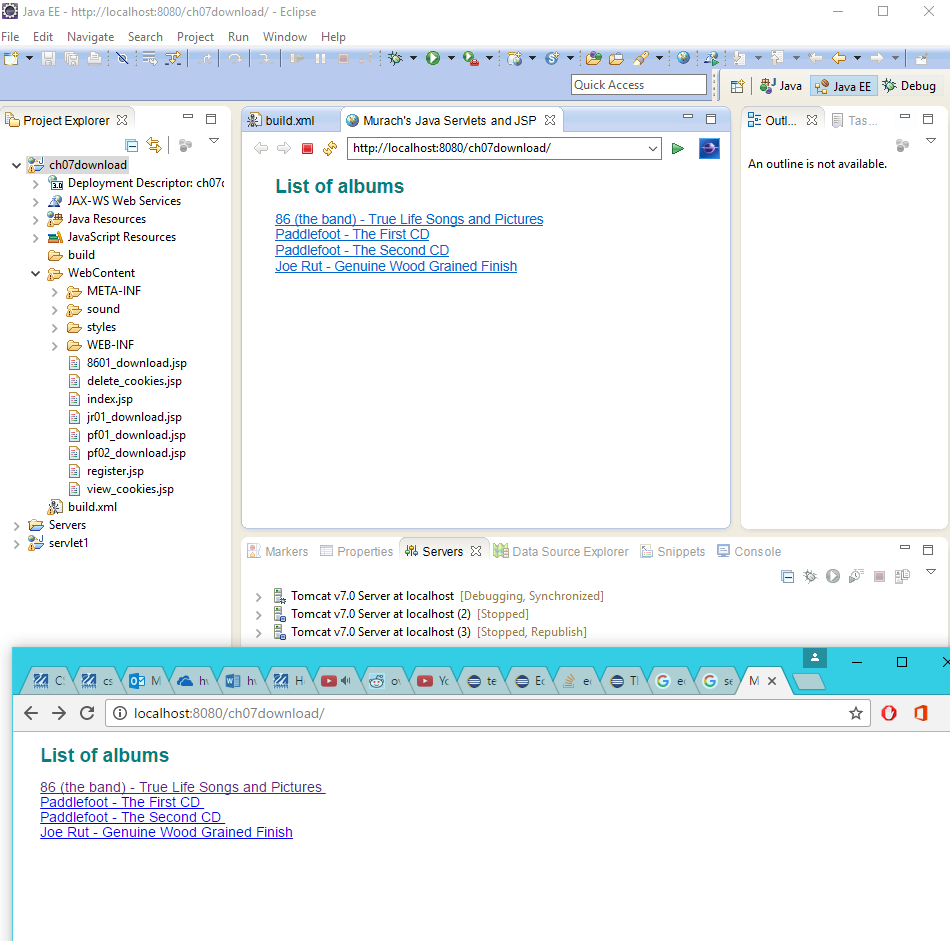
b.   Set a breakpoint in doGet. Run tomcat in debug mode from eclipse and see it hit the breakpoint. Report on what you see on the execution stack.

Tomcat in debug mode in eclipse below 

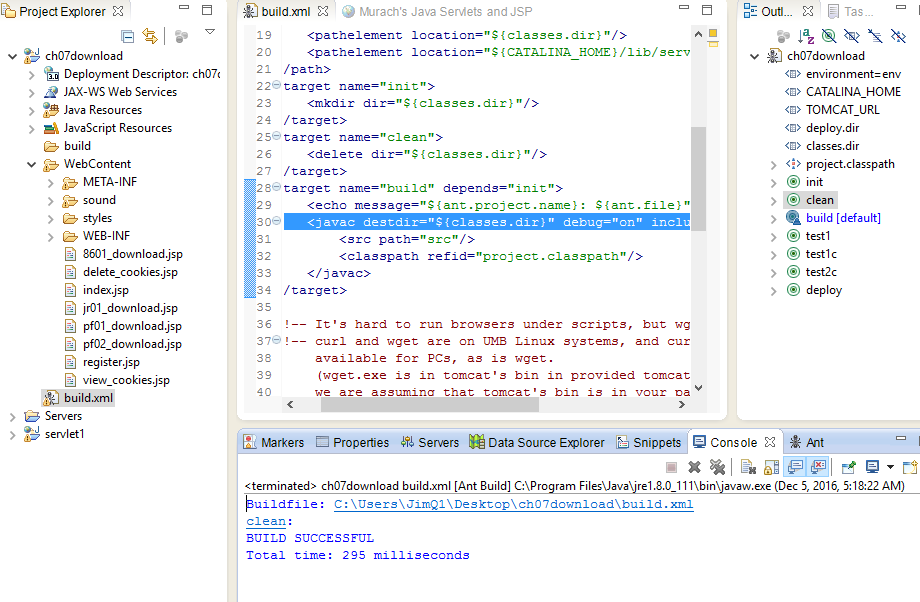
Running and setting a debug breakpoint in doGet() in debug mode



2.  Transfer the provided $cs636/ch07download project to your PC and start up a Dynamic Web project for it. The default choices for web-related options are fine. Note that this project does a part of project 2, but does not take advantage of our track table data.

a. Deploy this project to tomcat running in eclipse and use a browser to play a sample song for a CD, noting any problems.   


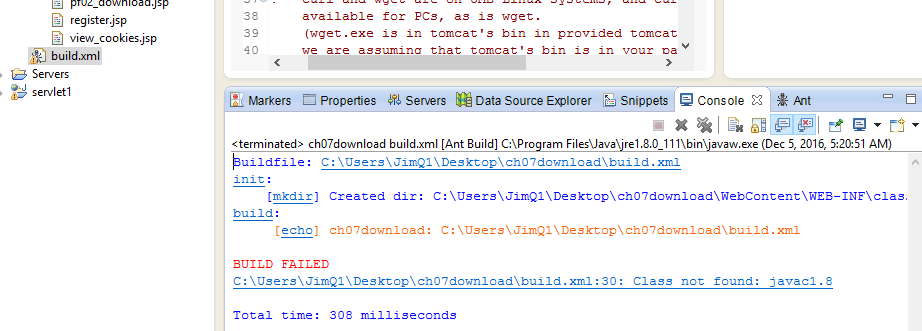
b. Note that this project comes with a build.xml.  On the command line or eclipse ant view on your development system, try ant clean, ant build, ant deploy, ant test1, and ant test1c if you have curl on your system. Note any problems. Explain what these tests are doing.

'ant clean'  


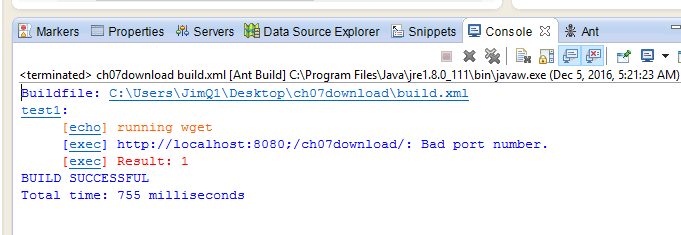
'ant build'

'ant deploy'

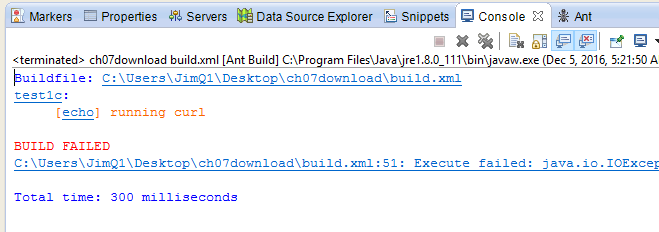
Build doesn’t seem to work on ant

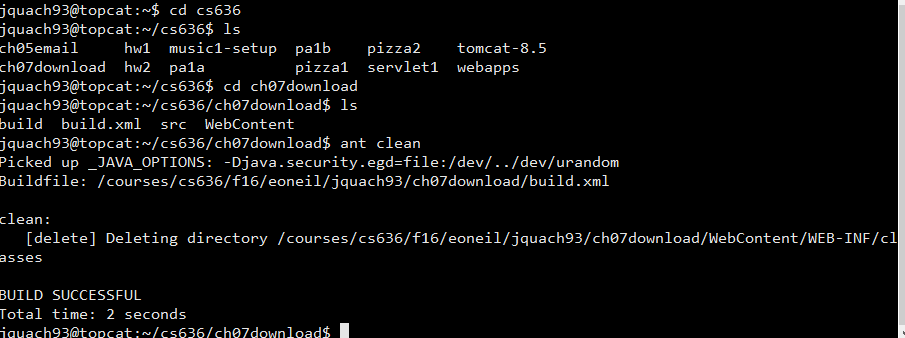


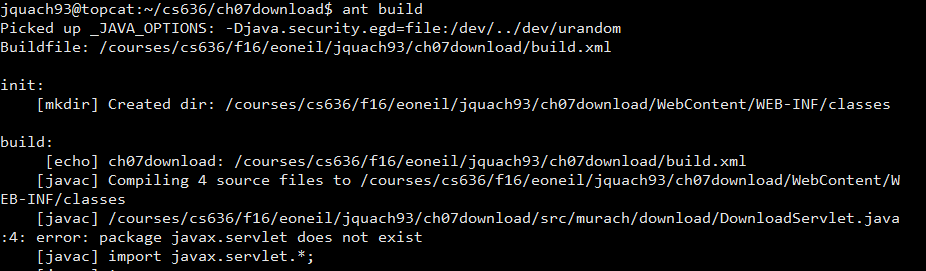
'ant test1'



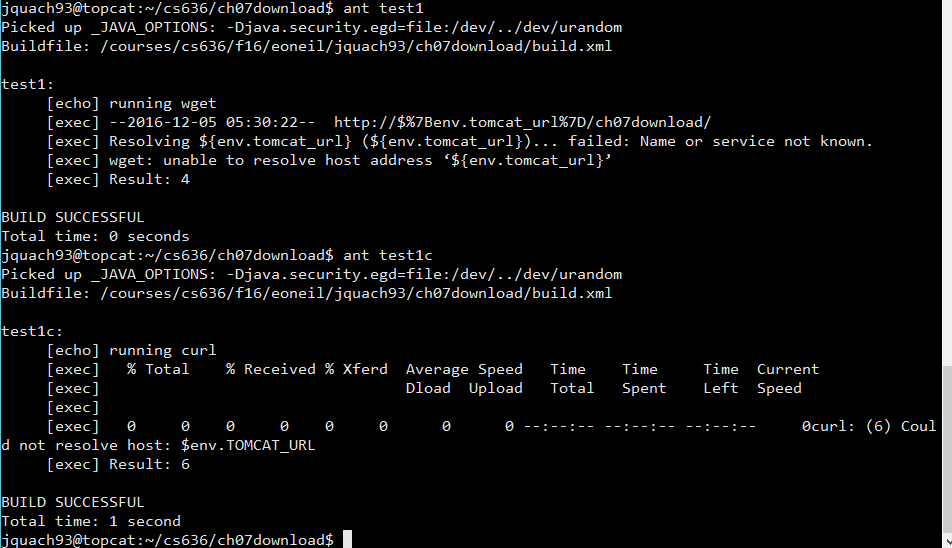
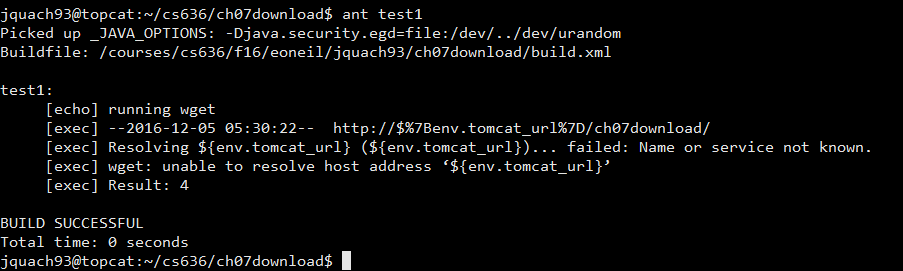
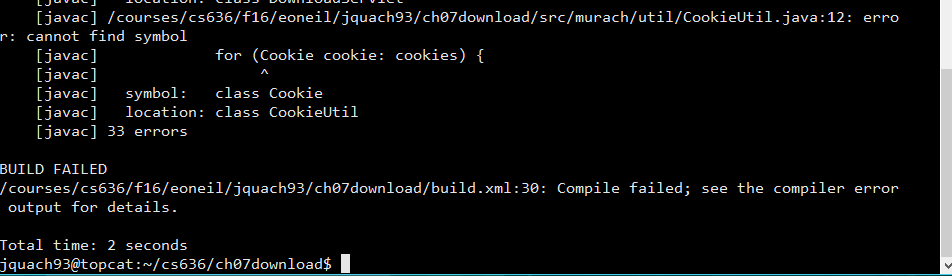
'ant test1c'

No curl on system

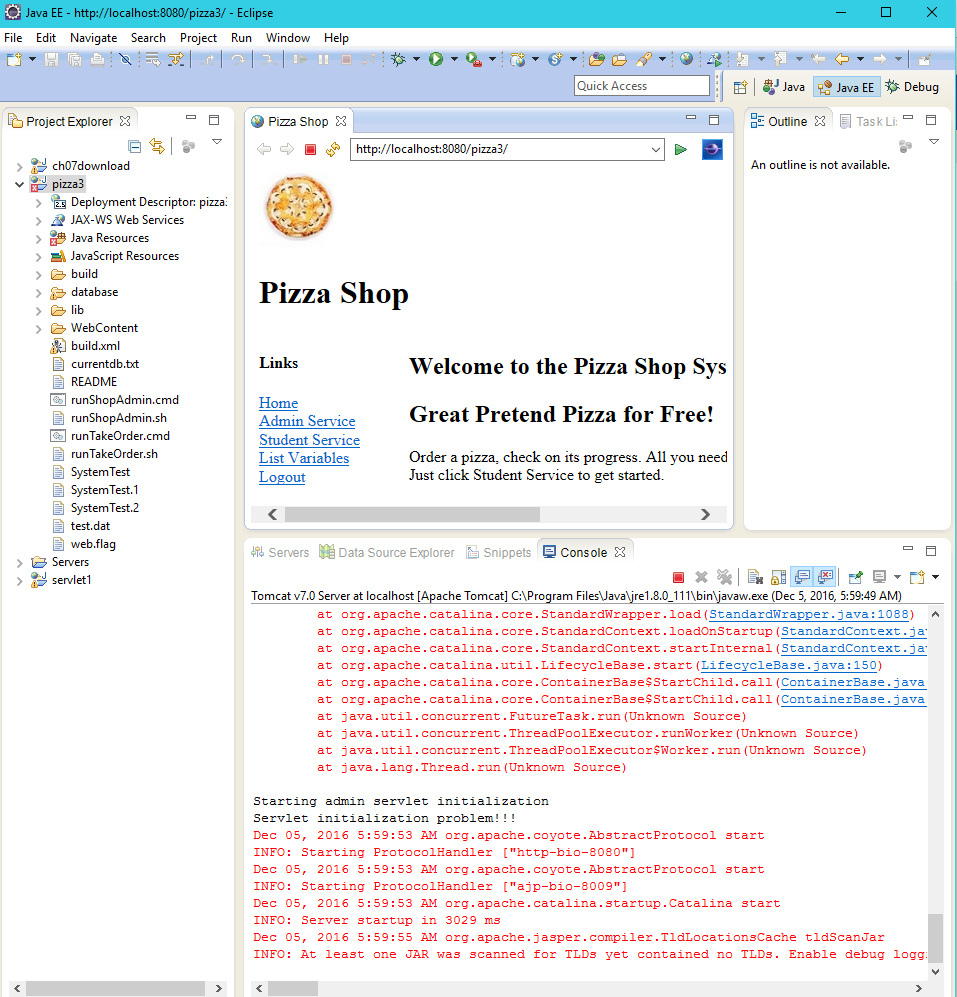
c. Do a. and b. on topcat too.  Note any problems.  




Hmm build doesn't build



3. Transfer $cs636/pizza3 to your development machine and set it up in a eclipse project, and deploy it with ant or eclipse.  See [handout](http://www.cs.umb.edu/cs636/UsingDBfromWebApp.html) about needed configuration of tomcat's conf/context.xml. Report on any problems. Bring down eclipse while you first set up tomcat's context.xml, and get pizza3 working with ant before running eclipse again: ant deploy, ant webTest1, ant webTest2. (Eclipse will try to manage context.xml itself, which confuses the initial configuration.) Then bring up eclipse again and set up a project for pizza3. Note that pizza3 needs a Dynamic Web Project.  



4.   Analyze the differences between the pizza2 and pizza3 projects by using the powerful and convenient eclipse compare capability. In Project Explorer, select two packages, for example, one from pizza2 and corresponding one from pizza3, and then right-click, choose Compare. Go through the diffs and report on each set of related edits, explaining why they are needed to change to a web app.  Don't include the diffs in your hw paper, just the discussion of them.

By doing a comparison of pizza2 and pizza3 there are obviously some differences between the two.

Within PizzaSystemConfig.java:

Pizza3 implemented import java.util.Map;.

'dbDAO' is now outside configureServices() and is a private static object.

'configureServices()' throws Exception as opposed to RuntimeException.

The exception throws 'new Exception("…...",e);'

Within 'testEMF(EntityManagerFactory emf)' in pizza3 throws an Exception. The exception handling has a try-catch with a Map object to get properies from the Entity Manager Factory. It has another try statement with a if statement to check for connection to the web.

Within DbDAO.java:

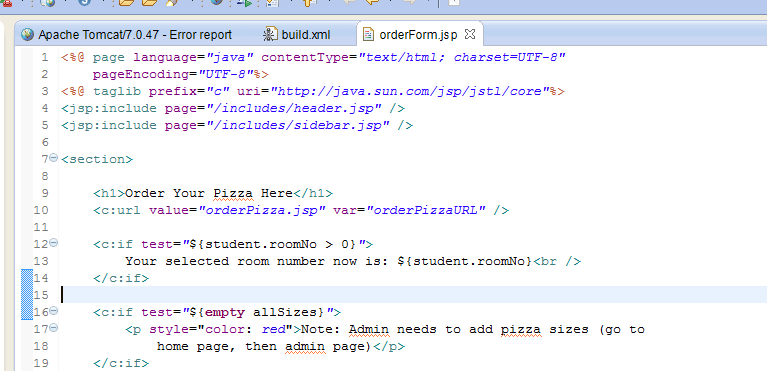
'em' is replaced with 'getEM()' in respective statements in pizza3.

A ThreadLocal<EntityManager> threadEM is implemented in pizza3 for thread handling for large sets of traffic in websites.

Within the presentation folder the clientserver folder, ShopAdmin.java, TakeOrder.java, and web folder is deleted since this project is meant to commicate with the database.

5.   In pizza3, the room number of the student is the essential Session variable. Trace how the room number gets set in the StudentBean in one request cycle and later used in another request cycle. Specifically, follow Student Service link to the Student Welcome page, then set room 3, and see the page redisplayed with "room=3" in the URL query string. Then follow the "Order a Pizza Now" link to the Order Form page. There your choice of 3 shows on the Order Form page without being in an incoming query string. How does the code know to display 3 as your room number?  It's because of the session variable, but can you be more specific?

Below in orderForm.jsp in pizza3 the statement that checks for the condition of 'student.roomNo>0' is where the session variable knows to display 3 as your room number.



Below within orderForm.jsp is where Room for delivery is indicated, and the check for the respective room to deliver the pizza.