CS375 - Networks
Ch 8 - Servlets
(tomcat)

Fall 2007
Thu, Nov 1
Tue, Nov 6
Tue, Nov 13
Marc L. Smith

Servlets (environment)

- To create and run servlets without Netbeans, we must set up some environment variables.
- Login to an Asprey Lab machine and edit your .bashrc file
  (if you don't have one, create it at this time)
- The .bashrc file executes whenever you login or open a bash terminal

Servlets (environment)

- Here are the contents of my .bashrc file (make the appropriate changes for your account):

```
# .bashrc

export JAVA_HOME="/usr/lib/jvm/java-1.5.0-sun"

export CATALINA_HOME="/usr/local/share/netbeans-5.5.1/
enterprise3/apache-tomcat-5.5.17"

export CATALINA_BASE="/home/mlsmith/.netbeans/5.5.1/apache-
tomcat-5.5.17_base"

export CLASSPATH="/usr/local/share/netbeans-5.5.1/enterprise3/
apache-tomcat-5.5.17/common/lib/servlet-api.jar"

export PATH=$PATH:$CATALINA_HOME/bin
```

Servlets (environment)

- Test that your environment is correct by starting and stopping Tomcat.
- Open a new terminal, which will cause your new .bashrc to execute, and set your environment variables
- Issue the command: startup.sh
- Issue the command: shutdown.sh
- You should see something similar to transcript on the next slide

Servlets (environment)

Transcript of startup and shutdown commands:

```
mlsmith@martha:~$ startup.sh
Using CATALINA_BASE:   /home/mlsmith/.netbeans/5.5.1/apache-
tomcat-5.5.17_base
Using CATALINA_HOME:   /usr/local/share/netbeans-5.5.1/
enterprise3/apache-tomcat-5.5.17
Using CATALINA_TMPDIR: /home/mlsmith/.netbeans/5.5.1/apache-
tomcat-5.5.17_base/temp
Using JRE_HOME:       /usr/lib/jvm/java-1.5.0-sun
mlsmith@martha:~$
mlsmith@martha:~$ shutdown.sh
Using CATALINA_BASE:   /home/mlsmith/.netbeans/5.5.1/apache-
tomcat-5.5.17_base
Using CATALINA_HOME:   /usr/local/share/netbeans-5.5.1/
enterprise3/apache-tomcat-5.5.17
Using CATALINA_TMPDIR: /home/mlsmith/.netbeans/5.5.1/apache-
tomcat-5.5.17_base/temp
Using JRE_HOME:       /usr/lib/jvm/java-1.5.0-sun
mlsmith@martha:~$
```

Servlets (environment)

- Startup Tomcat again: $ startup.sh
- Test whether you can display the default Tomcat homepage
  - ours uses different port from the book (the book references port 8080)
  - point your browser to: http://localhost:8084
- Shut down Tomcat: $ shutdown.sh

FirstServlet

- Create a web application by creating a directory where Tomcat knows to look
- under $CATALINA_BASE there should be a webapps directory. In that directory, create:
  - mywebapp (naming consistent with Ch 8)
  - mywebapp/WEB-INF
  - mywebapp/WEB-INF/classes

FirstServlet

- What files go where?
  - Under mywebapp: HTML files and JSPs
  - Under mywebapp/WEB-INF/classes: servlets (subdirectories if using packages)
  - Under mywebapp/WEB-INF:
    - web.xml (The deployment descriptor)
    - contains <servlet> and <servlet-mapping> tags for each servlet
    - maps URLs to .class files

FirstServlet

- URL for our servlet:
  http://localhost:8084/mywebapp/FirstServlet.html
- Not just accessible from localhost:
  - start tomcat from your server (al69 - al77)
  - access from another machine in the Asprey Lab
    e.g.,
    http://al69:8084/mywebapp/FirstServlet.html

Servlet Structure (a closer look)

- servlets are compiled Java .class files located in the `WEB-INF/classes` subdirectory of your webapp
- servlets are invoked from .html files, via `<FORM METHOD=GET ACTION="FirstServlet">`
- servlets must import `javax.servlet` and `javax.servlet.http` (in addition to `java.io`)
- servlet classes must extend HttpServlet

A servlet's four basic steps

1. Execute the setContentType method:
   ```
   response.setContentType("text/HTML");
   ```
2. Execute the getWriter method:
   ```
   PrintWriter out = response.getWriter();
   ```
3. Retrieve any parameter(s) from initial Web page: e.g.,
   ```
   String name = request.getParameter("FirstName");
   ```
4. Use println's to create elements of the web page to be "served up" by Tomcat, e.g.,:
   ```
   out.println("<HTML>");
   ```

Homework

- Get the following examples up and running on your server in the Asprey Lab
  - FirstServlet
  - PersonalServlet
  - SimpleAdder
- Enhance SimpleAdder to support: +,-,*,/
- Due: Fri, Nov. 9. 5pm
- I will test by accessing your web server...

8.10 Accessing MySQL via a Servlet

- Create a table to access
  $ mysql -h yoursql -u mlsmith -p
  (enter password when prompted)
  mysql> use marc (substitute your db name)
  mysql> create table PhoneNums
  mysql> (phnum_id int auto_increment primary key,
  mysql> Surname varchar(25),
  mysql> Forenames varchar(25),
  mysql> PhoneNum varchar(13));

Server/DB access

- Copy JDBCServletTest.html to $CATALINA_BASE/webapps/mywebapp
- Copy DBServlet.java to $CATALINA_BASE/webapps/mywebapp/WEB-INF/classes
- Edit web.xml to include <servlet> and <servlet-mapping> tags for DbServlet

Server/DB access

- Edit DbServlet.java
  - change URL to our MySQL server and your database: "jdbc:mysql://yoursql:3306/marc"
  - change Class.forName() to reference our database driver: "com.mysql.jdbc.Driver"
  - compile DbServlet.java
- Restart Tomcat

Server/DB access

- Point Firefox browser to: http://localhost:8084/mywebapp/JDBCServletTest.html
- Enter names and phone number
- click Commit
- debug! :-)