CS320 Web and Internet Programming

Introduction to Java Servlets

Chengyu Sun
California State University, Los Angeles

Servlet Hello World

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
   public void doGet( HttpServletRequest request,
                      HttpServletResponse response )
      throws ServletExceptoin, IOException
   {
      PrintWriter out = response.getWriter();
      out.println( "Hello World" );
   }
}
```

Some Simple Observations

- Inherits from HttpServlet
- There's no `main()` method
  - Who "runs" HelloWorld?
- doGet()
  - HttpServletRequest
  - HttpServletResponse sent back to the client browser

Compilation – The Hard Way

- `javac` HelloWorld.java
  - Where is the servlet library?
  - How to include it in compilation?
- Copy HelloWorld.class to the right directory

Compilation – The Slightly Smarter Way

- Set environment variable
  - .bashrc
- Use -d option of javac

Compilation – The Ant Way …

- Download build.xml from http://sun.calstatela.edu/~cysun/www/teaching/cs320/extras/build.xml
- Modify the properties to match your setup
  - `tomcat.home`
  - `install.dir`
- Create necessary directories
  - `src`
  - `web`

… Compilation – The Ant Way

- Put files in the right directories
  - Java source code: `src`
  - JSP and HTML files: `web`
- Compile (and more)
  - `ant`
  - `ant install`
  - `ant clean`
  - `ant uninstall`

Generate HTML

- Set content type to "text/html"
- Output a HTML page using println()
  - `<html>`
  - `<head>, <title>`
  - `<body>`
- Validate
  - validator.w3c.org
  - Sometimes useful for debugging

Package

- The advantages of using packages
- The necessity of using packages
  - A Class without a package qualifier cannot be used in a class that belongs to a package (enforced in JDK 1.4)
  - All servlets generated from JSPs are in packages

Package Example

- Package and directory structure
  - cs320.stu31.HelloWorld
- `package` statement
- URL to access a packaged servlet

Servlet Life Cycle

- Loaded for the first time – `init()`
- Per request – `service()`
  - dispatch to doXxx()
- Unloaded – `destroy()`
  - When requested by the manager user
  - When the context is *reloadable* and a new version of the servlet is found

I/O Re-examined

- Java application
  - Input
    - Console, GUI
    - File
    - CLP, Resource
  - Output
    - Console, GUI
    - File
- Servlet
  - Input ??
  - Output ??

File I/O Example

- Read `quotes.txt`
- Display the stocks which have gone up in a HTML table
  - `<table>`
  - `<tr>`
  - `<td>`

More About Files

- Paths
  - Absolute path
  - Relative path
  - *File path on the CS server??*
- Permissions
  - Who owns the file?
  - Who can read/write the file?

Servlet Debugging Tips

- Debugging servlets could be rather hard
- Section 3.8
  - Use print statements
  - Invest some time in a good IDE
    - Netbeans - http://www.netbeans.org/
    - Eclipse (WTP) - http://www.eclipse.org/
  - Follow good programming practice
  - Check out HTML source
  - Restart server