

Java Server Extensions

Servlets and JSP

CSE 794R/ECE 694R

1

Other Server-Side Technologies

- CGI (Common Gateway Interface)
 - one of oldest server-side technologies
 - server passes request to external program (often Perl, but also C, Python, PHP, etc.)
 - output of program is sent to client
- Limitations:
 - scalability: new process is created for each request
 - inefficiency: e.g., reloading interpreter with each request
 - awkward resource sharing with server and other processes
 - weak security model

CSE 794R/ECE 694R

2

Other Server-Side Technologies

- FastCGI
 - single persistent process for each program
- PerlEx
 - improved performance for Perl CGI scripts on Windows NT web servers
- mod_perl
 - module for Apache web server
- Microsoft ASP (Active Server Pages)
 - embed VBScript or JScript in web pages
- Server-side Javascript
 - embed Javascript in web pages

CSE 794R/ECE 694R

3

Java Server Pages (JSP)

- Embed Java in web pages
- JSP page is transformed into a servlet as part of its execution
- We'll come back to JSP after discussing servlets first

CSE 794R/ECE 694R

4

Java Servlets

- Generic server extensions
- Run inside JVM on server
- Multiple threads within the same process or within multiple processes spread across multiple servers (i.e., efficient & scalable)
- Bidirectional communication with server
- Portable across OSs and web servers
- Good support for security, data base access, synchronization, etc.
- **But all html generated by the servlet with output statements**

CSE 794R/ECE 694R

5

Servlet API

- Packages from J2EE platform
 - javax.servlet
 - generic, protocol-independent servlets
 - javax.servlet.http
 - add HTTP-specific functionality
- Needed to compile servlets

CSE 794R/ECE 694R

6

Servlet Container

- Needed to test and deploy servlets
 - standalone
 - add-on
 - embeddable
- Apache Tomcat (<http://tomcat.apache.org/>)
 - the official reference implementation
 - can operate standalone or as an add-on (and even be embedded in other apps)
 - Eclipse Tomcat plugin (<http://www.sysdeo.com/eclipse/tomcatplugin>)

CSE 794R/ECE 694R

7

HTTP Basics

- Client (browser) and (web) server communicate with HTTP protocol
- HTTP is stateless: client makes request, server responds, transaction is done
- Client request: *method URL version*
 - GET /intro.html HTTP/1.0
 - followed by optional header, e.g., browser type, content types understood
 - followed by optional body

CSE 794R/ECE 694R

8

HTTP Basics cont.

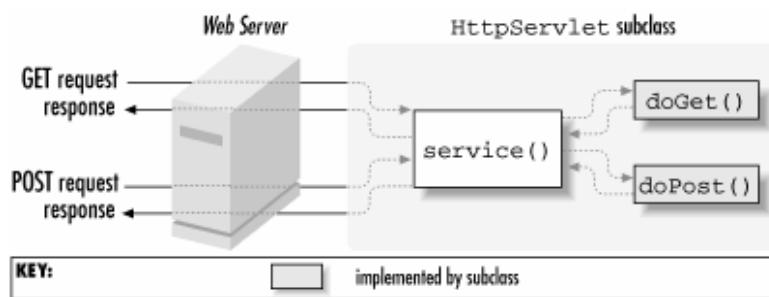
- Server response: *protocol status-code*
 - HTTP/1.0 200 OK
- followed by response headers
 - date, server, content type, length, etc.
- followed by the requested data

GET and POST

- GET designed to get information
 - can also be used to submit info by using *query strings*
 - URL length may be limited by server
- POST designed to post information
 - submits info as part of HTTP request body
 - unlimited length of submitted data

Servlet API

- Extend `javax.servlet.http.HttpServlet`
- Override `doGet()` and/or `doPost()`



CSE 794R/ECE 694R

11

Accessing Request and Response

- `javax.servlet.http.HttpServletRequest`
 - `getHeader()`, `getMethod()`, `getRequestURL()`, `getParameter()`, `getSession()`, etc.
- `javax.servlet.http.HttpServletResponse`
 - `setContentType()`, `getWriter()`, `setStatus()`, `sendError()`, etc.

CSE 794R/ECE 694R

12

The Servlet Lifecycle

- Lifecycle is highly flexible:
 1. Create and initialize the servlet
 2. Handle 0 or more service calls from clients
 3. Destroy the servlet, then garbage collect it

Servlet Advantage

- Common for all servlets to run in single JVM
 - efficiently share data
 - persistence between requests
 - even server can run in same JVM maximizing performance

Instance Persistence

- Servlets can persist between requests
- Server creates a single instance, which handles every request made to the servlet
- Memory footprint is small
- No object creation overhead (except first)
- Enables state persistence

- Beware of concurrency!!

CSE 794R/ECE 694R

15

Session Tracking

- HTTP is stateless
- How can we keep track of information from request to request
- Use Session Tracking API
 - `javax.servlet.http.HttpSession`
 - can be used to store and retrieve information about a user across HTTP requests
- Usually based on persistent cookies (or URL rewriting, if cookies are off)

CSE 794R/ECE 694R

16

HttpSession

- Get it from HttpServletRequest object with getSession()
- Can save any set of arbitrary (key, value) pairs in HttpSession
 - setAttribute()
 - getAttribute()
 - getAttributeNames()
 - removeAttribute()

CSE 794R/ECE 694R

17

URL Rewriting

- Web servers must support session tracking even for browsers that don't accept cookies (from servlet specs)
- HttpServletResponse.encodeURL(url)
- Modifies url to include session id
- All URLs emitted by a servlet should be run through this method

CSE 794R/ECE 694R

18

Cookies

- `javax.servlet.http.Cookie`
- Cookies contain:
 - name/value pair
 - `maxAge` that makes them expire
 - domain (i.e., when client should include in request)
- Get cookie information from `HttpServletRequest`
`request.getCookie()`
- Set cookies in `HttpServletResponse`
`response.addCookie(c)`

CSE 794R/ECE 694R

19

Initialization and Finalization

- `init()` – servlet initialization
 - parameters specified in `web.xml` file
 - `getInitParameter("parameter name")`,
parameters and values are Strings
 - guaranteed to be called once before first
request
- `destroy()` – servlet finalization
 - free resources, save persistent data, etc.
 - not guaranteed to be called

CSE 794R/ECE 694R

20

ServletConfig

- A servlet configuration object used by a servlet container to pass information to a servlet during initialization
- `javax.servlet.ServletConfig`
 - `getInitParameter()`
 - `getInitParameterNames()`
 - `getServletContext()`
 - `getServletName()`

CSE 794R/ECE 694R

21

ServletContext

- Defines a set of methods that a servlet uses to communicate with its servlet container
- There is one context per "web application" per Java Virtual Machine
- `javax.servlet.ServletContext`
 - `getAttribute()`
 - `getAttributeNames()`
 - `setAttribute()`
 - `removeAttribute()`

CSE 794R/ECE 694R

22

Web Application Deployment

- WebApplication
 - HTML docs, JSP pages, images, other resources
 - Other folders with application content
 - WEB-INF/
 - web.xml (deployment descriptor)
 - classes/
 - lib/

CSE 794R/ECE 694R

23

Deployment Descriptor

- WebApp/WEB-INF/web.xml
- XML document describing configuration
 - servlet aliases, mappings, initialization parameters
 - session timeout limits
 - global parameters to make available throughout the application
 - security configuration
 - mime types

CSE 794R/ECE 694R

24