

## CS320 Web and Internet Programming

### Handling HTTP Requests

Chengyu Sun  
California State University, Los Angeles

## Echo Server

- ◆ <http://cs.calstatela.edu:4040>
- ◆ Display HTTP requests

## HTTP Request Example

*http://cs.calstatela.edu:4040/whatever*

### GET /whatever HTTP/1.1

```
Host: cs.calstatela.edu:4040
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.3) ...
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,...
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: nxt/gateway.dll/uid=4B4CF072; SITESERVER=ID=f1675...
```

## HTTP Request

- ◆ Request line
  - Method
  - Request URI
  - Protocol
- ◆ Header
- ◆ [Message body]

## Request Methods

- ◆ Actions to be performed regarding the resource identified by the *Request URI*

### ◆ Browser

- GET
- POST

### ◆ Editor

- PUT
- DELETE

### ◆ Diagnosis

- HEAD
- OPTIONS
- TRACE

## HttpServlet Methods

- |           | → | service()     |
|-----------|---|---------------|
| ◆ GET     | → | ◆ doGet()     |
| ◆ POST    | → | ◆ doPost()    |
| ◆ PUT     | → | ◆ doPut()     |
| ◆ DELETE  | → | ◆ doDelete()  |
| ◆ HEAD    | → | ◆ doHead()    |
| ◆ OPTIONS | → | ◆ doOptions() |
| ◆ TRACE   | → | ◆ doTrace()   |

## Override HttpServlet Methods

- ◆ Default implementation returns an HTTP *Bad Request* error
- ◆ `doGet()`, `doPost()`
  - Simply call one method from the other

## Request Header Fields

- ◆ User-Agent
- ◆ Accept
- ◆ Accept-Charset
- ◆ Accept-Encoding
- ◆ Accept-Language
- ◆ Host
- ◆ Connection/Keep-alive
- ◆ Content-Length
- ◆ Cookie
- ◆ Referer

## HttpServletRequest

- ◆ `getThis()`, `getThat()`, ...
- ◆ <http://jakarta.apache.org/tomcat/tomcat-5.0-doc/servletapi/javax/servlet/http/HttpServletRequest.html>

## HttpServletRequest Exercises

- ◆ List all header fields
- ◆ Find the client address
- ◆ Is gzip compression supported?

## Request Parameters

- ◆ Query string
  - `?param1=value1&param2=value2&...`
- ◆ Form data

## Parameter Methods

- ◆ Inherited from *ServletRequest*
  - `String getParameter( String p )`
  - Enumeration `getParameterNames()`
  - `String[] getParameterValues()`
  - `Map getParameterMap()`

## Parameter Examples

- ◆  $A+B = C$
- ◆ File read

## HTML Forms

- ◆ `<form>`
  - action – URL of the server-side program that process the submitted form data
  - method – GET or POST
  - enctype
    - ◆ multipart/form-data
      - for file uploading
      - require POST

## `<input>`

- ◆ name
- ◆ value
- ◆ type
  - Text field ??
  - Password field ??
  - Submit button ??
  - Reset button ??
  - Check box ??
  - Radio button ??
- ◆ more types
  - File upload
  - Hidden field

## `<textarea>`

- ◆ name
- ◆ rows
- ◆ cols

## `<select>`

- ◆ name
- ◆ size
- ◆ multiple
- ◆ `<option value="something">some text`

## JavaScript Controls

- ◆ Button
- ◆ *onthis, onthat, ...*

## Form Example – Guestbook

### ◆ Filter HTML characters

- <: &lt;
- >: &gt;
- ": &quot;
- &: &amp;

## HTTP Request as User Input

### ◆ HTTP

- Request line and header fields
- Query string
- Form data

### ◆ Servlet classes

- ServletRequest and HttpServletRequest
- It's still Java!
  - ◆ Strings, streams, Enumerations, ArrayLists ...