HTTP & Server-Side Programming
In Raft, each node is in one of three states. These state are:

A. zero, one, two
B. Term 1, Term 2, Term 3
C. heartbeat, election, replicated log
D. follower, candidate, leader
E. senator, president, judge
In Raft, if a leader fails, then the other nodes will miss the heartbeat. The next node to become a candidate is selected:

A. by a majority
B. at random
C. by its node number – higher nodes are selected first
D. by a super majority of 2/3
E. by the previous leader
What type of Java objects serve the M, V, and C roles in InterestingPicture?

A. JSP, POJO, HttpServlet
B. POJO, JSP, HttpServlet
C. POJO, HttpServlet, JSP
D. HttpServlet, JSP, POJO
Select all of the following that are true:

A. Screen scraping is no longer needed now that web sites have APIs
B. Screen scraping is illegal
C. Screen scraping can be brittle and break whenever the web site owner changes their html
D. Screen scraping is the preferred approach to accessing data from web services
Goals

• Understand the HTTP application protocol
  – Request and response messages
  – Methods / safety / idempotence

• Understand Java server-side web application terminology
  – Servlet
  – jsp
  – Web container
  – HttpServlet

• Understand the flow of control
  – Web server, web container, servlet

• Understand flow of request to response
  – HttpServletRequest
  – HttpServletResponse

• Understand MVC in servlets and jsp.
The genesis of the WWW

- Sir Tim Berners Lee
  - 2016 Turing Award winner
- Defined 3 standards:
  1. Hypertext Transfer Protocol (HTTP)
  2. Uniform Resource Identifiers (URI)
     - UR Locators (URL) – e.g. http://cmu.edu/heinz
     - scheme://domain:port/path?query_string#fragment_id
     - UR Names (URN) e.g. urn:isbn:0132143011
       - urn:namespace_identifier:namespace_specific_string
  3. Hypertext Markup Language (HTML)
- Then implemented to these standards:
  - A server program
  - A browser program

Who maintains the standards?

The Internet Engineering Task Force (IETF) of the Internet Society, in cooperation with the World Wide Web Consortium (W3C)
• If you have not used HTML before, you should learn the basics.

• W3Schools is a good HTML tutorial resource
  – Do the HTML Tutorial from start through the "HTML Attributes" page (6 pages)
    • http://www.w3schools.com/html
  – Do all the "HTML Forms" pages (4 pages)
    • http://www.w3schools.com/html/html_forms.asp

• Lynda is another good resource:
  – http://www.cmu.edu/lynda/
Cascading Style Sheets

- Cascading Style Sheets (CSS) provide web-page styling
  - e.g. layout, fonts, colors
- **We do not require any CSS in this class.**
- But you can add some if you like to make your small apps look better.
- W3Schools CSS Tutorial:
  - http://www.w3schools.com/css
Basic HTML pages are static, and sometimes you want to include dynamic content
  – E.g. include values from a database query in a table.
As you saw in the lab HelloWorld servlet example, you can write HTML directly in a servlet.
  – e.g. out.println("<html>"); out.println("<head>");
  – Getting the HTML structure right can be troublesome.
What would be nice, would be to have the HTML structure (as a template), but be able to replace pieces programmatically.
JSP is a templating language
  – Allows you to write HTML templates and fill in values using Java code
Predecessor to newer templating engines
  – E.g. erb (Ruby) and ejs (JavaScript)
HTTP Interaction Pattern

- Uses Request / Response interaction pattern
  - Client
    - Establishes a connection with the server
    - Sends a request message and waits
  - Server
    - Processes the request as it is received
    - Sends a response message over the same client connection

- Coupled in time:
  - Both client and server must be available at time of the interaction

- Coupled in space:
  - Both client and server must know the address of each other, and communicate directly point-to-point

- Can also be used for Request / Acknowledge
HTTP Request

General Format

- <method> <resource identifier> <HTTP Version> <crlf>
- [<Header>: <value>] <crlf>
- ...
- [<Header>: <value>] <crlf>
- a blank line
- [entity body]

Example

GET /course/95-702/ HTTP/1.1
Host: www.andrew.cmu.edu
User-Agent: Joe typing
Accept: text/html
This line intentionally left blank

• **Method**
  - GET, PUT, DELETE, HEAD, POST, etc.

• **Resource identifier** specifies the name of the target resource;
  - i.e. it's the URL stripped of the protocol and the server domain name.
  - When using the GET **method**,
    - this field will also contain a series of name=value pairs separated by ‘&’.
  - When using a POST **method**,
    - the **entity body** contains these pairs.

• **HTTP version** identifies the protocol used by the client.
HTTP Response

<table>
<thead>
<tr>
<th>General Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;HTTP Version&gt; &lt;Status&gt; &lt;crlf&gt;</code></td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td><code>[&lt;Header&gt;: &lt;value&gt;] &lt;crlf&gt;</code></td>
<td>Date: Mon, 13 Jan 2014 15:43:08 GMT</td>
</tr>
<tr>
<td><code>...</code></td>
<td>Server: Apache/1.3.39 (Unix) mod_throttle/3.1.2 ...</td>
</tr>
<tr>
<td><code>[&lt;Header&gt;: &lt;value&gt;] &lt;crlf&gt;</code></td>
<td>Set-Cookie: webstats-cmu=cmu128.2.87.50.8400; ...</td>
</tr>
<tr>
<td><code>a blank line</code></td>
<td>Last-Modified: Sun, 12 Jan 2014 21:46:30 GMT</td>
</tr>
<tr>
<td><code>[response body]</code></td>
<td>Accept-Ranges: bytes</td>
</tr>
<tr>
<td></td>
<td>Content-Length: 9014</td>
</tr>
<tr>
<td></td>
<td>Content-Type: text/html</td>
</tr>
</tbody>
</table>

- **HTTP version** identifies the protocol used by the client.
- **Status** indicates the result of the request
Some Common HTTP Status Codes

- 200 OK
- 301 Moved Permanently
- 400 Bad Request
- 401 Unauthorized
- 404 Not found
- 500 Internal Server Error
  - You get this when your server program throws an uncaught exception.
  - You are likely to see this frequently this semester!

- For more information, see the standard:
  - http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
Show example

• Show with curl

• Show in Chrome
  – Intro:
    • https://developers.google.com/web/tools/chrome-devtools
  – Show Network tab
  – Inspect a particular request
    • Request and Response Headers
    • Response data
HTTP Methods

• Most frequently used:
  – GET – retrieve information
  – HEAD – query if there is information to retrieve
  – POST – add or modify a resource's information
    • if the resource identity is not known
  – PUT – add or replace a resource's information
    • if the resource identity is known (add) or can be predicted (replace)
  – DELETE – delete information

• When choosing which HTTP methods to implement, consider:
  – Safety
  – Idempotence
HTTP Methods

• You should study and know:
  – What *safe* means
  – What *idempotent* means
    • Why idempotent is useful (hint: it has to do with failure)
  – The common HTTP methods (previous slide)
    • What they are used for
    • Whether they defined as being safe
    • Whether they defined as being idempotent

• Good resource to study:
• Interesting test question:
  – Should HTTP DELETE be safe?
  – Should HTTP DELETE be idempotent?
  – How would you handle receiving an HTTP DELETE referring to a resource that has already been deleted?
<table>
<thead>
<tr>
<th>Method</th>
<th>Purpose?</th>
<th>Safe?</th>
<th>Idempotent?</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Retrieve a resource</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PUT</td>
<td>Insert or replace a resource</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>DELETE</td>
<td>Remove a resource</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>HEAD</td>
<td>Get header information only of a resource</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>POST</td>
<td>Add or modify a resource</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
HTTP standard

• HTTP is a standard.
  – How it is actually implemented depends on the developer.
  – A developer could choose to use HTTP DELETE to add records to a database.
    • Good question to ponder: What benefit is there to following standards?

• Misuse is common

• E.g. Flickr uses GET to do everything, including removing a photo from your favorites list
  – http://api.flickr.com/services/rest/?method=flickr.favorites.remove&...
HTTP/2

• A new version of HTTP is being adopted.
• Read this good HTTP/2 FAQ
  – https://http2.github.io/faq/#general-questions
• Know for the exam, at a high level, what are the key differences from HTTP/1.x?
  – (There is a bullet list in the FAQ.)
Java Servlets

- What does a server do when it receives an HTTP request?
- Java Enterprise Edition (JEE) is the platform (i.e. set of classes) for building distributed systems in Java.
- HttpServlet (commonly called a "Servlet") is the class designed to process HTTP requests.

- Great Reference:
  - Head First Servlets & JSP
    by Bryan Basham; Kathy Sierra; Bert Bates
  - Ch 1 – 3
In init() you might:
• Initialize a database
• Open file(s)
• Register with other objects
• Connect (sockets) to other services
You must overwrite one of:
- doGet
- doPost
- doPut
- doDelete
- doHead
In `destroy()` you might:
- Close a database or file
- Clean up relationships to other objects
- Close (sockets)

Source: Head First Servlets and JSP by Basham, Sierra, & Bates
You implement, e.g., `doGet(HttpServletRequest request, HttpServletResponse response)`
Model – View - Controller

MVC is discussed in the Lab 2 Video 3
• See handout, only available in class.