Session 7

Deployment Descriptor
Http

1

Reading and Reference

- Reading
  - en.wikipedia.org/wiki/HTTP
- Reference
  - http headers
    - en.wikipedia.org/wiki/List_of_HTTP_headers
  - http status codes
    - en.wikipedia.org/wiki/Http_status_codes
  - http spec
    - www.ietf.org/rfc/rfc2616.txt?number=2616

2

Lecture Objectives

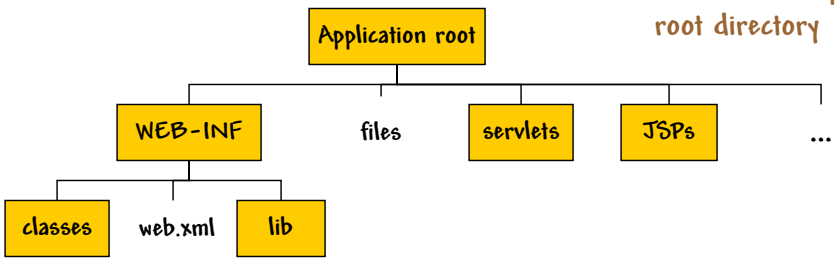
- Understand the directory structure of a Web application
- Understand that Http is a stateless, request/response protocol
- Understand the structure of HTTP messages
- Recognize the kinds of information that can be transmitted in Http headers (both request and response)

3

Web Application

- Collection of servlets, JSPs, HTML, images, etc.
- Can be portably deployed to any servlet-enabled web server
- Usually packaged in a war file

The server maps the application name in the URL to the web app root directory



4

WEB-INF Directory

- Does not contain files served directly to the client
- Contains classes and configuration information for the web app
  - WEB-INF/classes - contains class files for servlets
  - WEB-INF/lib - contains library classes - stored in jar files

Many Web Containers support a notation /servlet/xyz for locating the servlet class - but use of web.xml is better

5

Deployment Descriptor (web.xml)

- web.xml file is the deployment descriptor - allows Web applications to be deployed
  - An xml file (50+ defined elements)
  - Contains configuration information
  - Provides url string mapping, servlet name/class mapping, security, etc.

6

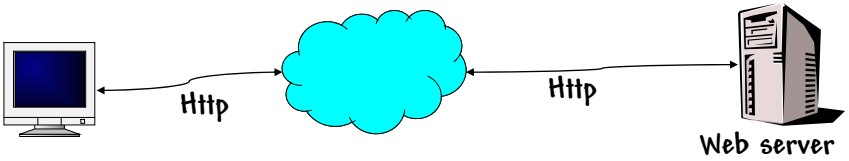
Http

- HyperText Transfer Protocol defines communications between a browser and a server
- Defined in 2 specs (http 1.0 and http 1.1)
- Defines:
  - Types of messages exchanged (request and response)
  - Syntax of the messages
  - Semantics of the message content
  - Rules for determining how and when a process sends and responds to a message

7

Http

- Hypertext Transfer Protocol
- Primary Web application layer protocol - uses TCP
- Implemented as
  - Client program - in browser (request message formatting)
  - Server program - in Web server (parsing the request method and preparing the response message)
- Http defines the structure of messages sent between the client and the server



8

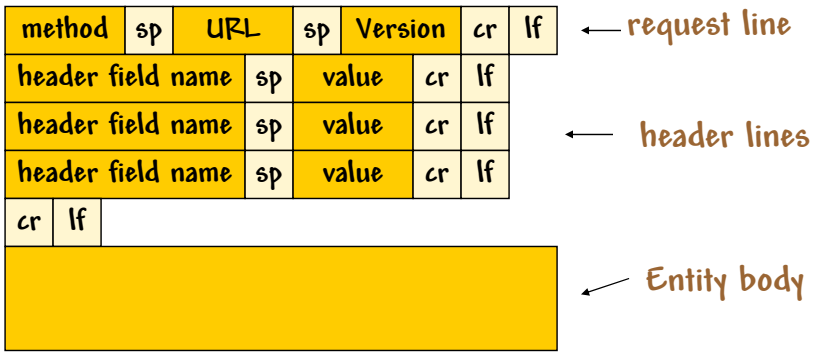
Http Protocol

- HTTP is a request/response (stateless) protocol
  - A client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content
  - The server responds with a status line, including the message's protocol version and a success (or error) code, followed by a MIME-like message containing server information, entity metainformation, and possible entity-body content.

9

Request Message Format

- The http request is specified by the request line, a variable number of header fields, and the entity body



10

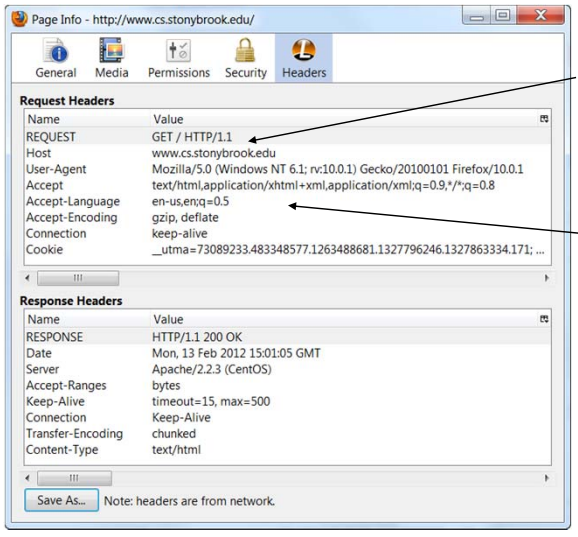
Http Methods

- OPTIONS - request for information concerning communications options (e.g., support of http 1.1)
- GET - retrieve information
- HEAD - identical to GET, except the server does not return a message body
- POST - modify a server resource
- PUT - store the enclosed entity
- DELETE - request that the resource be deleted
- TRACE - response contains the entire message request in the response body
- CONNECT - used in SSL tunneling

You will use get and post

11

Http Request From Browser



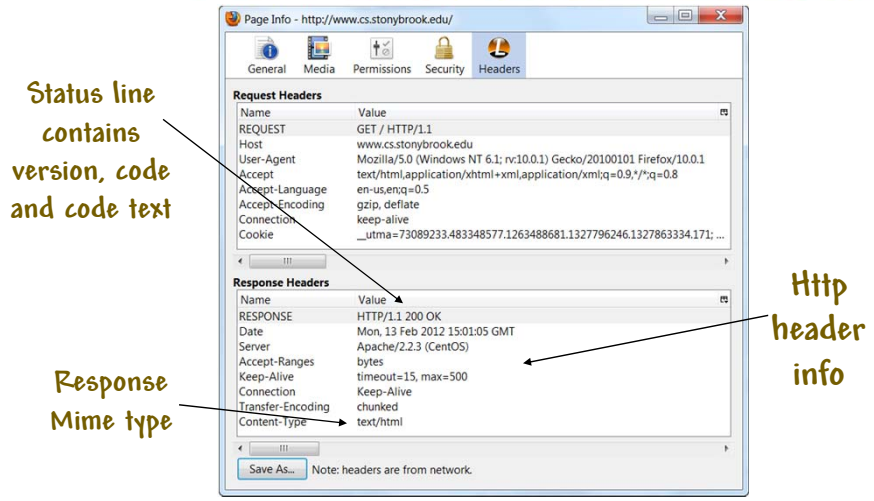
Request line contains the method, URL, and http version

Header lines contain http data

For the POST method, the form data set is transmitted in the Http entity body, not in the URL

12

Http Response From Server



Http Request Message

- Http messages (other than the body) are written in ASCII text
- Http request messages consist of:
  - Request line (method, URL, version)
  - Header lines (connection, user-agent, accept-language, etc)
  - Entity body
    - Not used for GET requests
    - Used for uploading files (as in WDG HTML validator)

Http Request Headers

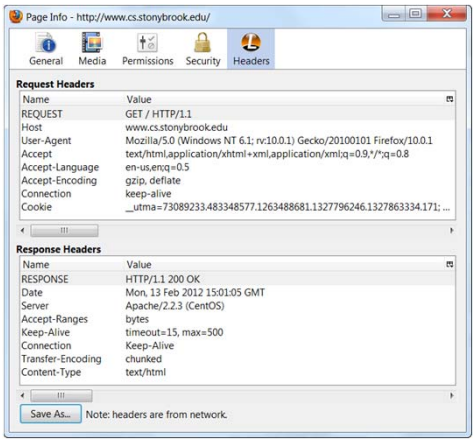
- Accept
- Accept-charset
- Accept-encoding
- Accept-language
- Authorization
- Cache-control
- Connection
- Content-length
- Content-type
- Cookie
- Expect
- From
- Host
- If-match
- If-modified-since
- If-none-match
- If-range
- If-unmodified-since
- Pragma
- Proxy-authorization
- Range
- Referer
- Upgrade
- User-agent
- Via

15

FireFox Plugin

- Install the Live Headers plugin for FireFox
- Displays http headers

Try this out on your system now. Do you see any http 1.0 transactions?



16

Http Response Message

- Http response messages consist of:
  - Status line (protocol version, status code, status message)
  - Header lines (date, server, last-modified, content-length, content-type)
  - Entity body

17

Http Status Codes

- Examples:
  - 200 - OK
  - 100 - Continue
  - 404 - Not found

You will see this code in your browser if the Web Application cannot find your servlet

18

Http Response Headers

- Accept-Ranges
- Age
- Allow
- Cache-Control
- Connection
- Content-Encoding
- Content-Language
- Content-Length
- Content-MD5
- Content-Type
- Date
- Etag
- Expires
- Last-Modified
- Location
- Refresh
- Server
- Set-Cookie
- Via
- Warning

19

Non-Persistent Connections

- Used in Http 1.0
- For each object on the same server identified in an html page (e.g., image files),
  - Client initiates a TCP connection with the server
  - Client sends an http request message to the server
  - Server sends the http response to the client
  - Server closes the TCP connection
  - Client receives the response message
- Browsers sometimes lessen the delays of the above by opening multiple simultaneous connections to the server

20

Http 1.1

- Most servers and browsers now use Version 1.1 (previous version was 1.0)
- In HTTP/1.1, the default is that a connection may be used for more than one request/response exchange - (persistent connection)
- Persistent connections can be pipelined (default) in which there are multiple outstanding request over the same connection

21

Setting the Response Header

- setHeader method in HttpServletResponse specifies the header name and the header value

```
resp.setHeader("Refresh", "5");
```

Header name

- Convenience methods (correspond to equivalent setHeader/parameter combinations)
  - setContentType
  - setContentLength
  - addCookie
  - sendRedirect

22

Have You Satisfied the Lecture Objectives?

- Understand the directory structure of a Web application
- Understand that Http is a stateless, request/response protocol
- Understand the structure of HTTP messages
- Recognize the kinds of information that can be transmitted in Http headers (both request and response)

23