

Session 8

Deployment Descriptor Http

HTTP/1.1 301 Moved Permanently	
GET http://www.cs.stonybrook.edu/~cse336/	
CONTENT-ENCODING	gzip
CONTENT-LENGTH	257
CONTENT-TYPE	text/html; charset=iso-8859-1
DATE	Thu, 02 Feb 2017 17:39:50 GMT
LOCATION	http://www3.cs.stonybrook.edu/~cse336/
SERVER	Apache/2.2.22 (Ubuntu)
VARY	Accept-Encoding
HTTP/1.1 200 OK	
GET http://www3.cs.stonybrook.edu/~cse336/	
ACCEPT-RANGES	bytes
CONNECTION	Keep-Alive
CONTENT-LENGTH	25719
CONTENT-TYPE	text/html
DATE	Wed, 22 Feb 2017 17:23:44 GMT
KEEP-ALIVE	timeout=15, max=500
SERVER	Apache

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

Lecture Objectives

- 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)

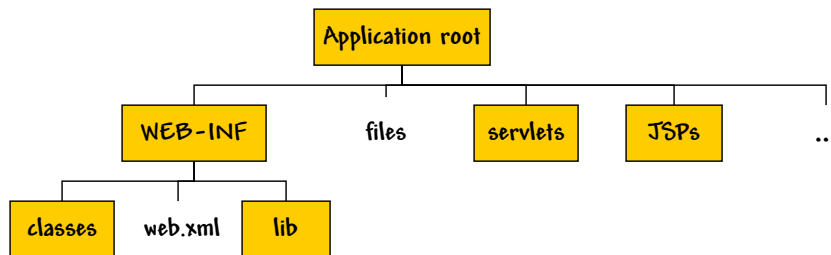
© Robert Kelly, 2018

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



© Robert Kelly, 2018

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

© Robert Kelly, 2018

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.

© Robert Kelly, 2018

6

Http

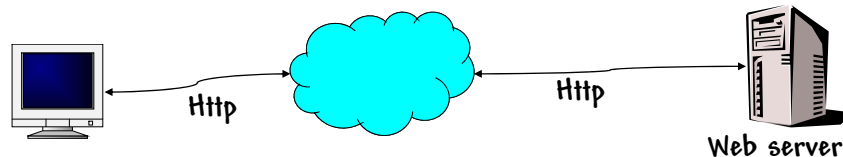
- HyperText Transfer Protocol defines communications between a browser and a server
- Defined in specs (HTTP 1.0, HTTP 1.1, and HTTP/2)
- Defines:
 - Types of messages exchanged (request and response)
 - Syntax of the messages
 - Semantics of the message content
 - Rules for determining how and when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

© Robert Kelly, 2018

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



© Robert Kelly, 2018

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 meta-information, and possible entity-body content.

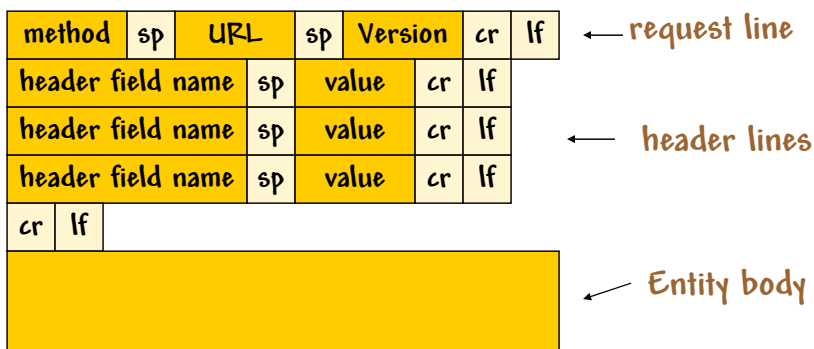
Stateless protocol is an
important part of
RESTful services

© Robert Kelly, 2018

9

Request Message Format

- The http request is specified by the request line, a variable number of header fields (each appended by a colon), and the entity body



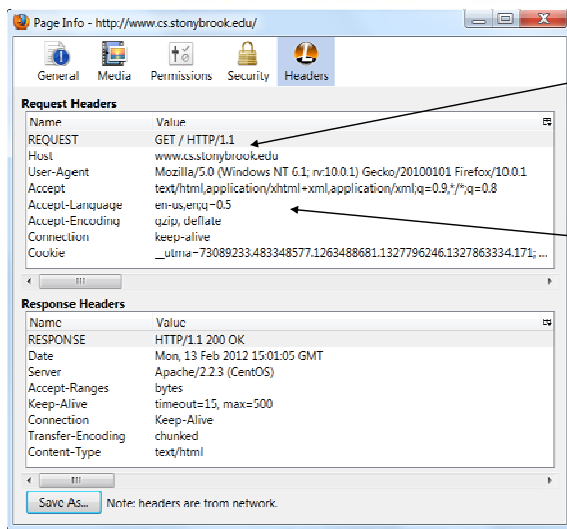
© Robert Kelly, 2018

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

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

Http Response From Server

The screenshot shows the 'Page Info' dialog for the URL `http://www.cs.toronto.edu/`. It displays two sections: 'Request Headers' and 'Response Headers'. The 'Request Headers' section includes fields like Name, Value, REQUEST (GET / HTTP/1.1), Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Connection, and Cookie. The 'Response Headers' section includes Name, Value, RESPONSE (HTTP/1.1 200 OK), Date, Server, Accept-Ranges, Keep-Alive, Connection, Transfer-Encoding, and Content-Type. Annotations with arrows point to specific parts: 'Status line contains version, code and code text' points to the 'REQUEST' line; 'Response Mime type' points to the 'Content-Type' header; and 'Http header info' points to the 'RESPONSE' line.

Status line contains version, code and code text

Response Mime type

Http header info

© Robert Kelly, 2018

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

© Robert Kelly, 2018

15

Browser Plugin

- Most browsers have one or more tools to inspect your http dialog
- Take a minute to check which tool is installed in your favorite browser

Live Http Headers icon in a Chrome browser

#	Method	Status	Url
8	GET	200	http://www.cs.stonybrook.edu/favicon.ico
7	GET	200	http://www.cs.stonybrook.edu/favicon.ico
6	GET	200	http://www.w3.org/conn/vaid XHTML?0
5	GET	200	http://www.cs.stonybrook.edu/~cae336/images
4	GET	200	http://www.cs.stonybrook.edu/~cae336/images
3	GET	200	http://www.cs.stonybrook.edu/~cae336/images
2	GET	200	http://www.cs.stonybrook.edu/~cae336/Cse332
1	GET	200	http://www.cs.stonybrook.edu/~cae332/

Header	Value
Accept	text/html,application/xhtml+xml,application/javascript
Accept-Encoding	gzip, deflate, sdch
Accept-Language	en-US,en;q=0.8
Cookie	__utmz=5365362.1426403836.608.CFD0271817.1760160963376632022; CFT008E10=1760160963376632022; __utmz=42881847.1391278185.1.1.4; __utmz=73089233.208026202.13796; __utmz=73089233.1393888166.83.1; __utmz=73089233.1393888166.83.1;
User Agent	Mozilla/5.0 (Windows NT 6.1; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36)
Response Headers	
Accept-Ranges	bytes
Connection	Keep-Alive
Content-Length	22892
Content-Type	text/html
Date	Mon, 10 Mar 2014 19:55:48 GMT
Keep-Alive	timeout=15, max=100
Server	Apache

© Robert Kelly, 2018

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

© Robert Kelly, 2018

17

Http Status Codes

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

Status codes become more important with RESTful services

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

© Robert Kelly, 2018

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

© Robert Kelly, 2018

19

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

© Robert Kelly, 2018

20

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)