STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 146: Intro to Web Programming and Project Development

Instructor: Iraklis Tsekourakis

Lieb 213

Email: itsekour@stevens.edu

HTTP

Objectives

Students will be able to:

- Understand the functionalities provided by HTTP
- Explain the break down of an HTTP URL

What is HTTP?

- ***H**yper**T**ext **T**ransfer **P**rotocol*
- An application protocol for distributed, collaborative, hypermedia information systems
- Hypermedia is an extension of hypertext
  - non linear medium of information including graphics, video, audio, plain text and hyperlinks
- Multimedia: non-interactive linear presentation of graphics, etc.

How it all started?

- First version was HTTP/0.9, created by early developers of the Web, which had its beginnings at CERN (1989)
- HTTP/1.1 was released in June 1999
- Protocol for client-server communication

HTTP vs HTML

- HTML: hypertext markup language
  - Definitions of tags that are added to Web documents to control their appearance
- HTTP: hypertext transfer protocol
  - The rules governing the conversation between a Web client and a Web server

Both were invented at the same time by the same person

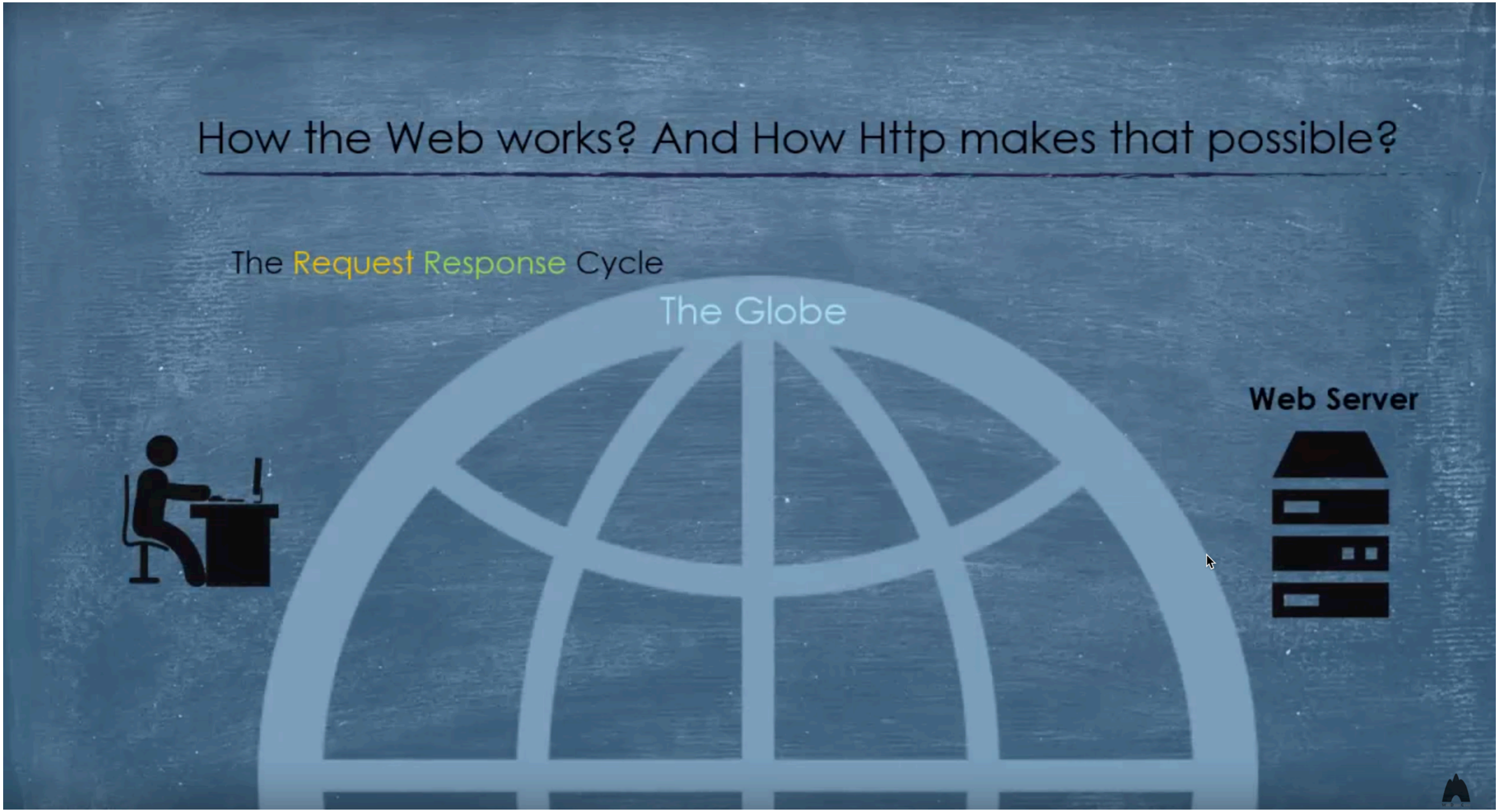
Application Layer Protocols

Protocol	Application
HTTP: Hypertext Transfer	Retrieve and view Web pages
FTP: File Transfer	Copy files from client to server or from server to client
SMTP: Simple Mail Transport	Send email
POP: Post Office	Read email

Three Important Facts about HTTP

- **Connectionless** protocol:
  - After making the request, the client disconnects from the server, then when the response is ready, the server re-establishes the connection and delivers the response
- HTTP can deliver any sort of data, as long as the two computers are able to read it
- **Stateless** protocol:
  - The client and server know about each other just during the current request
  - If it closes and the two computers want to connect again, they handle the connection as it was the first time

How does it work?



How the Web works? And How Http makes that possible?

HTTP URL

- URL stands for Uniform Resource Locator
- Syntax:
  - <scheme>://<user>:<password>@<host>:<port>/<urlpath>;<params>?<query>#<fragment>
  - For the Web, the scheme is HTTP

Breaking Down an HTTP URL

- Syntax:
  - http://**<user>:<password>**@<host>:<port>/<urlpath>?<query>#<bookmark>
  - <user> and <password> supply optional authentication information
  - Are rarely used

Breaking Down an HTTP URL

- Syntax:
  - http://<user>:<password>@**<host>:<port>**/<urlpath>?<query>#<bookmark>
  - <host> is the host name of the web server where the resource is located
  - May also be an IP address

Breaking Down an HTTP URL

- Syntax:
  - http://<user>:<password>@<host>:<port>/**<urlpath>**?<query>#<bookmark>
  - <urlpath> is the path pointing to the specific resource to be retrieved by HTTP
  - Usually a full directory posting
  - Is case-sensitive, even though DNS domain names are not

Breaking Down an HTTP URL

- Syntax:
  - http://<user>:<password>@<host>:<port>/<urlpath>?**<query>**#<bookmark>
  - <query> is an optional query or other information passed to the web server
  - Alternative method to get information there is via HTTP POST

Breaking Down an HTTP URL

- Syntax:
  - http://<user>:<password>@<host>:<port>/<urlpath>?<query>#**<bookmark>**
  - <bookmark> identifies a particular location within an HTML document
  - Allows users to click a hyperlink and scroll to a particular place in the document

HTTP Client Request Methods
(most common in bold)

GET
- **Retrieve a representation of the resource**

HEAD
- **Like GET but without content**

POST
- **Submit data to be processed, includes the data in the message**

PUT
- Upload a representation

DELETE
- Deletes the resource

TRACE
- Send the received request to see if it changed on the way

OPTIONS
- Request a list of active functions supported by the server

CONNECT
- Used to simplify SSL by translating communications to TCP/IP

PATCH
- To slightly modify a resource

HTTP Server Response

- HTTP response
  - Status line
    - HTTP-version   Status-code   Reason
    - Status-codes 1xx - Informational
      - Reserved for future use

HTTP Server Response

- HTTP response
  - Status line
    - Status-codes 2xx - Success
      - The action was successfully received, understood, and accepted

» 200	OK
» 201	POST command successful
» 202	Request accepted
» 203	GET or HEAD request fulfilled
» 204	No content

HTTP Server Response

- HTTP response
  - Status line
    - Status-codes 3xx - Redirection
      - Further action must be taken in order to complete request
        - » 300 Resource found at multiple locations
        - » 301 Resource moved permanently
        - » 302 Resource moved temporarily
        - » 304 Resource has not modified (since date)

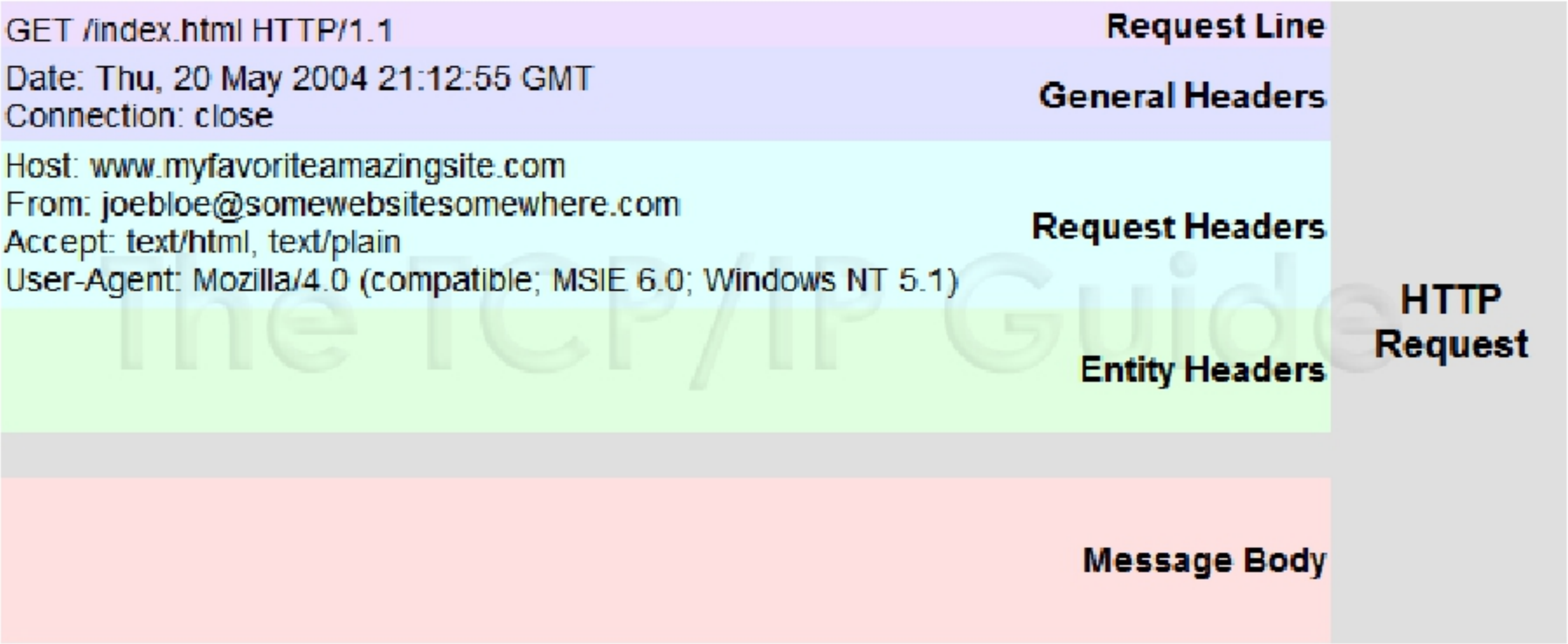
HTTP Server Response

- HTTP response
  - Status line
    - Status-codes 4xx - Client error
      - The request contains bad syntax or cannot be fulfilled
        - 400 Bad request from client
        - 401 Unauthorized request
        - 402 Payment required for request
        - 403 Resource access forbidden
        - **404 Resource not found**
        - 405 Method not allowed for resource
        - 406 Resource type not acceptable

HTTP Server Response

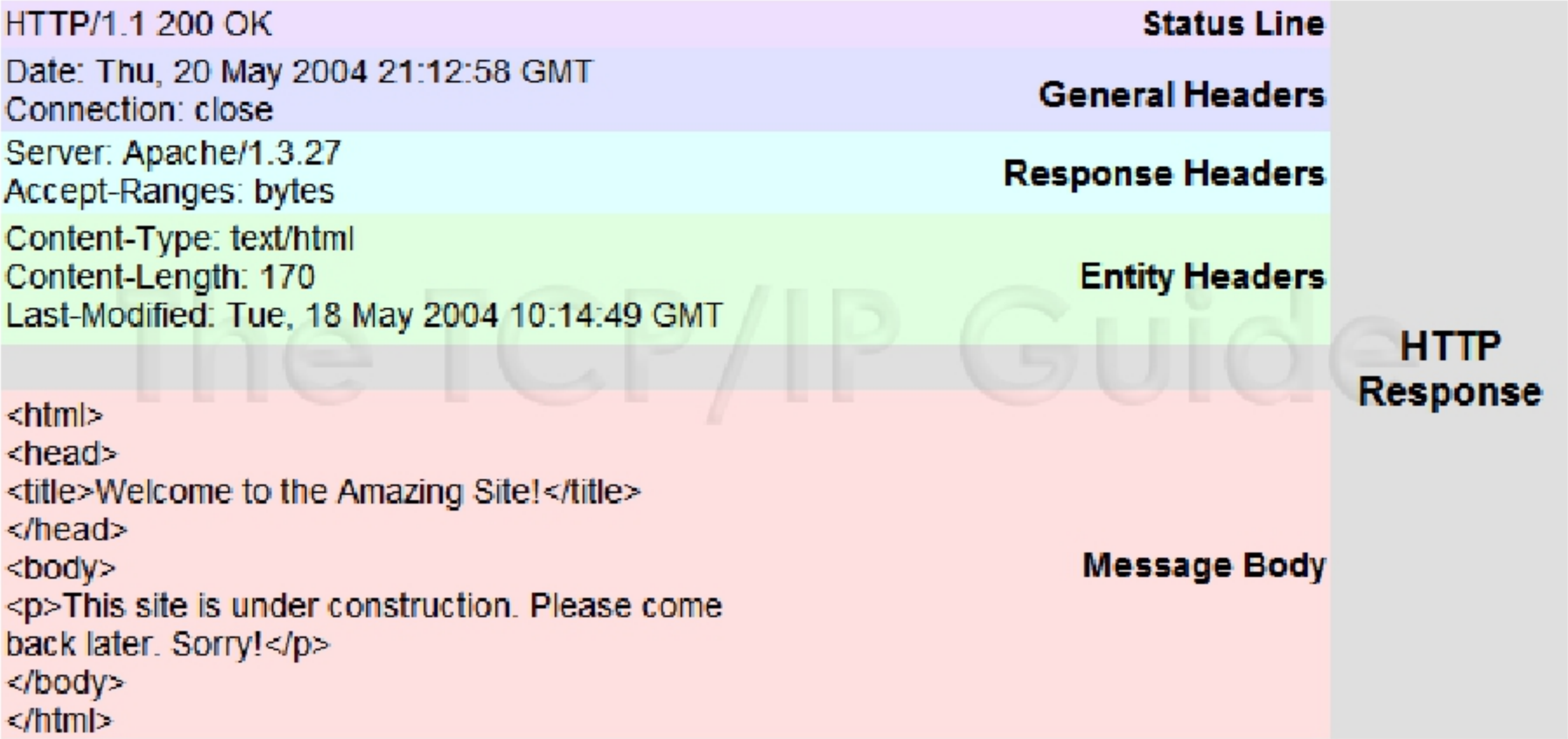
- HTTP response
  - Status line
    - Status-codes 5xx - Server error
      - The server failed to fulfill an apparently valid request
        - » 500 Internal server error
        - » 501 Method not implemented
        - » 502 Bad gateway or server overload
        - » 503 Service unavailable / gateway timeout
        - » 504 Secondary gateway / server timeout

HTTP Request



```
GET /index.html HTTP/1.1
Date: Thu, 20 May 2004 21:12:55 GMT
Connection: close
Host: www.myfavoriteamazingsite.com
From: joebloe@somewebsitesomewhere.com
Accept: text/html, text/plain
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

HTTP Response



```
HTTP/1.1 200 OK
Date: Thu, 20 May 2004 21:12:58 GMT
Connection: close
Server: Apache/1.3.27
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 170
Last-Modified: Tue, 18 May 2004 10:14:49 GMT

<html>
<head>
<title>Welcome to the Amazing Site!</title>
</head>
<body>
<p>This site is under construction. Please come
back later. Sorry!</p>
</body>
</html>
```

HTTPS (HTTP Secure)

- HTTPS is HTTP-within-SSL/TLS
- SSL (TLS) establishes a secured bidirectional tunnel for arbitrary binary data between two hosts
- HTTP is meant to run over a bidirectional tunnel for arbitrary binary data; when that tunnel is a SSL/TLS connection, then the whole is called "HTTPS”

HTTPS

- Uses TLS (Transport Layer Security) / SSL (Secure Sockets Layer) to encrypt the transmission of data
- Syntactically identical to HTTP
- Only needs one party to be authenticated
- Relies on certificates issued by several companies
- Usually runs on port 443

Play time!

- Open a terminal (Mac/Linux) or a command line tool(MS)
  - type: “telnet google.com 80”
  - wait for the response and then type: “GET / HTTP/1.1”, and press return twice…

Tools to View HTTP Traffic

- Chrome/Webkit Inspector (fav. amongst web-devs)
- Fiddler (Windows), Charles Proxy (Mac)
  - web debugging proxies (not only inspect, but offer expanded capabilities to interact with your site)
- Command line tools:
  - curl, tcpdump, tshark (wireshark with GUI)