

CS3226

WEB PROGRAMMING AND APPLICATIONS

LECTURE 02 - HTTP

DR STEVEN HALIM

STEVENHALIM@GMAIL.COM

OUTLINE

- Client-Server Paradigm
- URL and Domain Name
- HTTP Examples - Live Demo
- HTTP Basic Authentication

CLIENT-SERVER PARADIGM (HTTP)

Client

- Initiates contact with server (speaks first)
- Typically requests service from server
- The client is implemented in a web browser

Server

- Typically waits for requests from clients (standby 24/7)
- Provides the requested services to client
- Web server sends the requested web pages

For more details, GIYF, e.g. read [World Wide Web Consortium \(W3C\) explanation](#)

PROTOCOLS

For two parties to communicate,
they must use the same standard (protocol):

A protocol defines:

- Message format
- Order of messages sent and received
- Actions taken on message transmission and receipt
- Etc

Examples: Ethernet, TCP, IP (v4 and v6), HTTP (we will jump into HTTP as the rest are the topics of [CS2105 - Introduction to Computer Networks](#))

HYPertext TRAnSFER PROTOCOL (HTTP)

HTTP is a connectionless, stateless protocol that defines how a web browser and a web server communicate*

Both the request and the response look like this:

```
INITIAL LINE  
HEADERS  
<CRLF> (an empty line)  
BODY
```

Reference: [RFC2068](#)

FOR AN HTTP REQUEST:

1. The INITIAL LINE can be GET or HEAD or POST (there are others) and it includes the URL encoded resource as well as the HTTP protocol version number
2. The HEADER specifies things like Host: ... Cookie: ...
3. The BODY of a request contains any payload data, e.g. a web form (discussed later) which uses HTTP POST request will have the URL encoded data in the body of the HTTP request

FOR AN HTTP RESPONSE:

1. The INITIAL LINE has the status of the request

Status	Meaning
HTTP/1.1 200	OK
HTTP/1.1 404	Not Found
HTTP/1.1 500	Internal Server Error

2. The HEADER contains things like Set-cookie:, Last-Modified:, Content-Type:, ..., etc
3. The BODY contains the payload of the response, e.g. the HTML file for the webpage, the image file, ..., etc

URL: UNIFORM RESOURCE LOCATOR

The definition of the HTTP URL:

```
httpurl      = "http://" hostport [ "/" path [ "?" query ] ]
hostport     = host [ ":" port ]
host         = hostname | hostnumber
hostname     = *[ domainlabel "." ] toplabel
hostnumber   = digits "." digits "." digits "." digits
port         = digits
path         = segment *[ "/" segment ]
hsegment     = *[ uchar | ";" | ":" | "@" | "&" | "=" ]
query       = *[ uchar | ";" | ":" | "@" | "&" | "=" ]
```

Usually for HTTP, the default port number* is 80

URL, CONTINUED

Absolute HTTP reference:

[SoC website](#), [SoC website \(index.html\)](#)

Relative HTTP reference:

[Go up one folder and access VisuAlgo index page](#)

Some characters has to be encoded before it can be placed in a URL, e.g. a space is replaced with a '%20'

Reference: [Request for Comments RFC1738](#),
[Percent encoding](#)

QUERY STRING

Query string is important for client-server communications,
both for HTTP GET and HTTP POST methods

Typical syntax:

```
?key1=value1&key2=value2&key3=value3
```

Example: [https://nusmods.com/timetable/2015-2016/sem2?
CS3226\[LAB\]=4&CS3226\[LEC\]=1&CS3233\[LEC\]=1](https://nusmods.com/timetable/2015-2016/sem2?CS3226[LAB]=4&CS3226[LEC]=1&CS3233[LEC]=1)

In this course, we will frequently use such URL query string

DOMAIN NAME

The host of a URL can be a hostname or a hostnumber

The hostnumber is also the Internet Protocol/IP address:
"http://" digits "." digits "." digits "." digits (IPv4 version)

<http://137.132.80.57> is the IP address of SoC website

But that IP address is surely very hard to remember

DNS SERVER

Domain Name System (DNS) Server provides a mapping between easy-to-remember Domain Name to an IP address

That is, DNS maps `http://" (*[domainlabel "."] toplabel)` to
`"http://" digits "." digits "." digits "." digits`

e.g. `http://www.comp.nus.edu.sg` to `http://137.132.80.57`

Note: We can check* who owns that domain name using `who.is` tool (and it's corresponding Alexa ranking*)

WHAT'S IN A NAME?

"What's in a name?

*That which we call a rose by any other
name would smell as sweet."*

- William Shakespeare

An easy-to-remember Domain Name will increase the traffic to your web application and therefore good Domain Names are heavily commercialized...

Quick sharing (details nearing the end of this course):
Experience with <http://visualgo.net>

HTTP EXAMPLES - LIVE DEMO

Example 1: A successful **GET** request*, OK (200) response

We use valid URLs, e.g. this webpage itself

Example 2: A **GET** request for a file that the server cannot find along with the server's Not Found (404) response

We use random non-existing URLs (of a valid server address)

Example 3: A **POST** request of a web-form along with the server's OK (200) response

We use our own [simple form](#) (revisited later in PHP lecture)

Example 4: A **GET** request that will return Unauthorized (401) response if credentials are wrong (next slide)

HTTP BASIC AUTHENTICATION (APACHE)

1. Create a password file "**passwd**" with uids and passwords using `htpasswd` tool

```
joe:$apr1$Q2bZd5p9$fdnfilagGx92ZK3r/WbhE1
```

2. Restrict access to resource in a directory by placing a special file "**.htaccess**" in it, e.g.

```
AuthType Basic  
AuthName "Enter 'joe' as uid and 'student' as pwd to bypass this"  
AuthUserFile "path...to...passwd"  
Require user joe
```

Now try accessing this [link](#)

THE END