



# 703128 PS/2 Web Services

HTTP

Zaenal Akbar

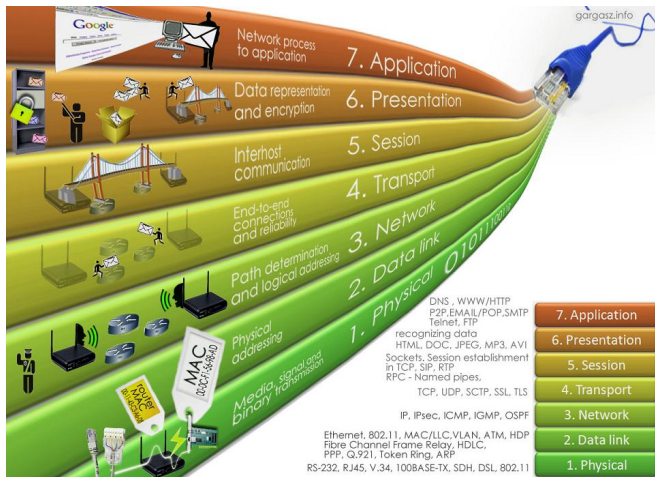
Friday, 2017-10-06

# Outline

- HTTP Overview
  - HTTP Protocol
  - HTTP Request/Response
  - HTTP Response Codes
  
- Bi-weekly Exercise

# HTTP

# OSI Reference Model

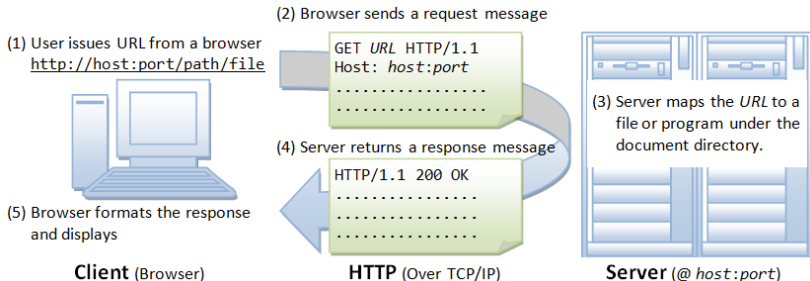


(\*) <http://www.gargasz.info/osi-model-how-internet-works/>

# The Basics: HTTP Protocol

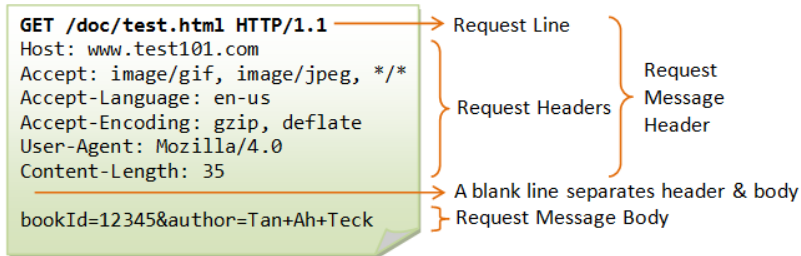
- The Hypertext Transfer Protocol (HTTP) is **an application-level** TCP/IP based protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems (internet)
- Defines how to interact with Web servers:
  - Request-reply interactions (stateless)
  - Content types
  - One resource per request
  - Simple access control

# HTTP Request/Response



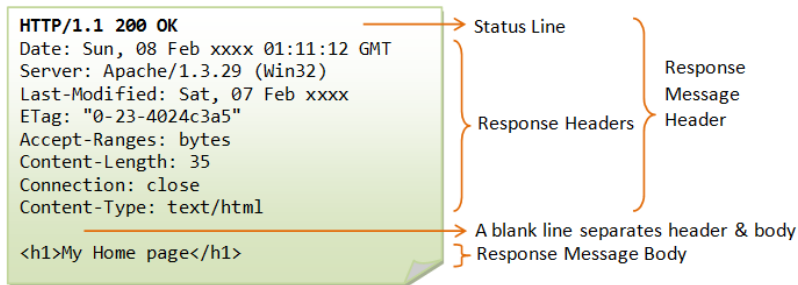
(\*) [http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)

# HTTP Request Anatomy



(\*) [http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)

# HTTP Response Anatomy



(\*) [http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](http://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)



# HTTP Request Methods

- GET → retrieve a web resource from the server
- HEAD → same as GET but returns no data, only information about it
- POST → send data to a resource on the server
- PUT → store the data passed in the request with the URL as identifier
- DELETE → delete the resource identified by the URL
- OPTIONS → request the list of methods the server allows to be applied
- TRACE → sends back the request (diagnostic)

# HTTP Response Codes

- 1xx: Information
- 2xx: Successful
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

# Exercises

# Send Request to Server

Use telnet<sup>1</sup>

```
$ telnet www.innsbruck.at 80
```

```
GET / HTTP/1.1
```

```
Host: www.innsbruck.at
```

Use curl<sup>2</sup>

```
$ curl http://www.innsbruck.at
```

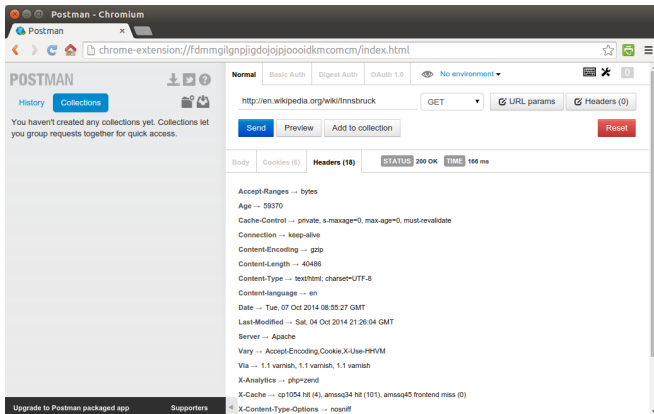
---

<sup>1</sup><http://en.wikipedia.org/wiki/Telnet>

<sup>2</sup><http://curl.haxx.se/>

# Send Request to Server

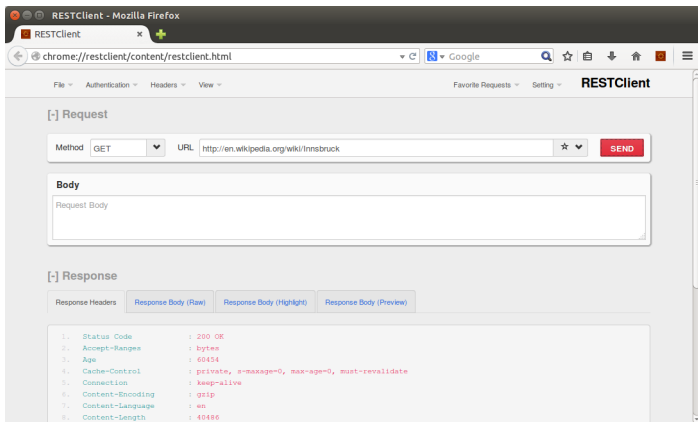
- Chrome Extension: Postman - REST Client



(\*) <https://github.com/a85/POSTMan-Chrome-Extension/wiki>

# Send Request to Server

- Firefox Extension: RESTClient



(\*) <http://restclient.net/>

# Examples

Try and see the response headers:

```
$ curl -I https://en.wikipedia.org/wiki/Innsbruck
HTTP/1.1 200 OK
Content-language: en
...
```

```
$ curl -I -0 https://de.wikipedia.org/wiki/Innsbruck
HTTP/1.1 200 OK
Content-language: de
Connection: close
...
```

```
$ curl -I https://en.wikipedia.org/Innsbruckk
HTTP/1.1 404 Not Found
...
```

# Examples

```
$ curl -I https://en.wikipedia.org/wiki/innsbruck
HTTP/1.1 301 Moved Permanently
Location: http://en.wikipedia.org/wiki/Innsbruck
...
```

```
$ curl -I -A "Mobile" http://en.wikipedia.org/wiki/Innsbruck
HTTP/1.1 302 Found
Location: http://en.m.wikipedia.org/wiki/Innsbruck
...
```

```
$ curl -I -A "Mobile" https://www.twitter.com
HTTP/1.1 302 Found
Location: https://mobile.twitter.com/
...
```



# Task

## Assignment:

1. Download the worksheet form the course webpage
2. Complete the exercises in the worksheet
3. Send it to the tutor before the next two sessions

# Thank You