

Networking with TCP and HTTP

Module 2 Week 5

Notes Repo: <https://github.com/C-Shi/lhl-flex-lecture>



Learning Objectives

Network and Internet

Layer of Internet (TCP/IP model)

HTTP, Request and Response

What is Internet

A global system of interconnected computer networks



TCP / IP Model

APPLICATION

Application
Presentation
Session



SMTP



HTTP



FTP

- Runs your application

TRANSPORT



TCP / UDP

- Assign port number
- Segmentation

NETWORK



IP, Router, Gateway

- Origin & Destination IP
- Packets

DATA LINK



Ethernet



WiFi

- Link physical infrastructure
- MAC Address
- Error checking

PHYSICAL



Copper Cables



WiFi Card

- Convert data to signal

100 *SECONDS OF* 



computer
NETWORKING

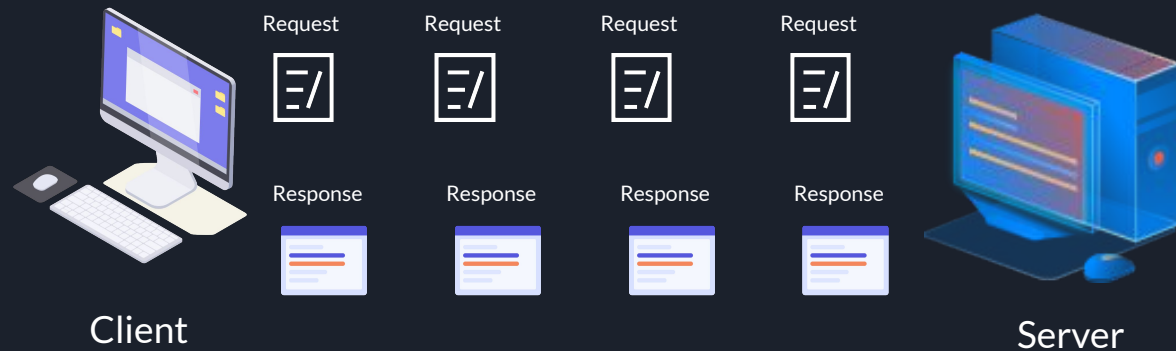
What is HTTP

Hypertext Transfer Protocol

Utilizes TCP for client-server communication

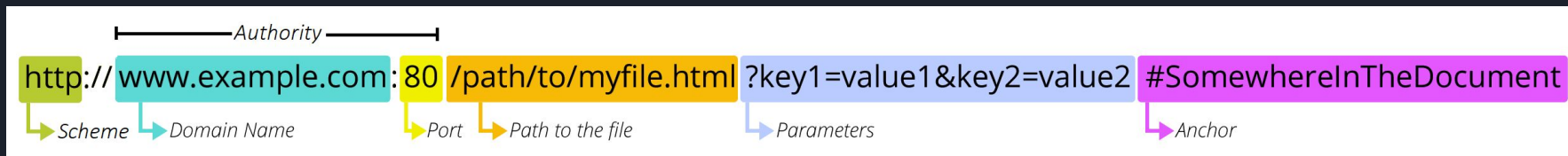
HTTP is **stateless**

HTTP operates based **request/response cycle**



WHERE is the URL

- Uniform Resource Locator
- The address of a given unique resource on the Web





HOW to use Method

Method	Description	Example
GET	To read resources	You search a video on Youtube
POST	To write resources	You post a question on Stackoverflow
PUT/PATCH	To update resources	You edit your comment on Facebook
DELETE	To delete resources	You delete picture from cloud

Using correct HTTP method is a standard, good practice, and rules to follow



Quizzes

You visit CNN website to see today's news	GET
You login to your google account	POST
You download your monthly statement from online banking	GET
You pay your amazon order and now the order is completed	PUT/PATCH
You delete your post on twitter	DELETE
You delete your cloud pictures so now it is in the recycle bin	PUT/PATCH
You logout of your github account	POST

WHAT is Request Headers

HTTP headers allow us to specify additional information about the request

▼ Request Headers

:authority: flex-web.compass.lighthouselabs.ca

:method: GET specify what method is this request

:path: /profile/edit

:scheme: https

accept: text/html, application/xhtml+xml Specify what type of response data is expected

accept-encoding: gzip, deflate, br

accept-language: en-US,en;q=0.9 specify additional information about the request such as who you have

cookie: _ga=GA1.2.2040925388.1647716735; user_id=MjA4--b588e508842ea554ca6bb3e521581a3fe0cb2f64; _compass_session=c25CMi9hN1RMV3RQ0FhTUjBXQnZ0NHF4MvdMQ3UyaXk1TlFERDRJRzlhbVZlNXVtWFBmT0NWQXc2SHpRV3lPQ2tFSytXenBxeWFyWCTjcExDZkU4RnQyV0RWFjR4MnRCRGpid2lkOE5HUW9XWxhuSXRueUg0SHlLN3VxZkxyMi9UK1FLTzdCQ1VsbWtNZlhjLzV0SVkrckNlYm5HYXZTTmk0eThrbHBF3K292eEVCSnh3ckw5YkpNY3A1N1dzU090LS1xaGtiZ050bGRlMFFqNEloeUpBeXFhPT0%3D--d7364d2acd4fda54829f99887e20e4eb23af861a; _dd_s=logs=1&id=2502d28d-15fc-4843-80b5-1915e694a48c&created=1649262658659&expire=1649263696417

referer: https://flex-web.compass.lighthouselabs.ca/activities/361?workbook_id=flex-m01w3

sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="99", "Google Chrome";v="99"

sec-ch-ua-mobile: ?0

sec-ch-ua-platform: "macOS"

sec-fetch-dest: empty

sec-fetch-mode: cors

sec-fetch-site: same-origin

turboinks-referrer: https://flex-web.compass.lighthouselabs.ca/activities/361?workbook_id=flex-m01w3 Specify how you send the request

user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36



WHAT is Request Body

1. Body is optional. Usually used with POST/PUT/PATCH
2. Contains information that you want to write on the server
3. Payload can be:
 - a. Form Data: default data type when using HTML form
 - b. JSON: usually sent through AJAX
 - c. Multipart/form-data: for carrying BLOB

HTTP Response

Header: like request header. Contains status

Status: Information about the state of the request

Body: Information that is requested by the client.

```
HTTP/1.1 200 OK
```

```
Cache-Control: max-age=604800
```

```
Content-Type: text/html
```

```
Date: Fri, 30 Mar 2018 00:24:55 GMT
```

```
Etag: "1541025663+gzip+ident"
```

```
Expires: Fri, 06 Apr 2018 00:24:55 GMT
```

```
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
```

```
Server: ECS (dfw/279D)
```

```
Vary: Accept-Encoding
```

```
X-Cache: HIT
```

```
Content-Length: 1270
```

```
<!doctype html>
```

```
<html>
```

```
  <!-- more html content >
```

```
</html>
```

HTTP Status

- **200 range:** success
 - 200 OK
 - 201 CREATED
 - 202 ACCEPTED
- **300 range:** redirect
 - 301 MOVED PERMANENTLY
 - 302 FOUND
- **400 range:** client error
 - 400 BAD REQUEST
 - 401 UNAUTHORIZED
 - 404 NOT FOUND
- **500 range:** server error
 - 500 INTERNAL SERVER ERROR

