WEB BASICS

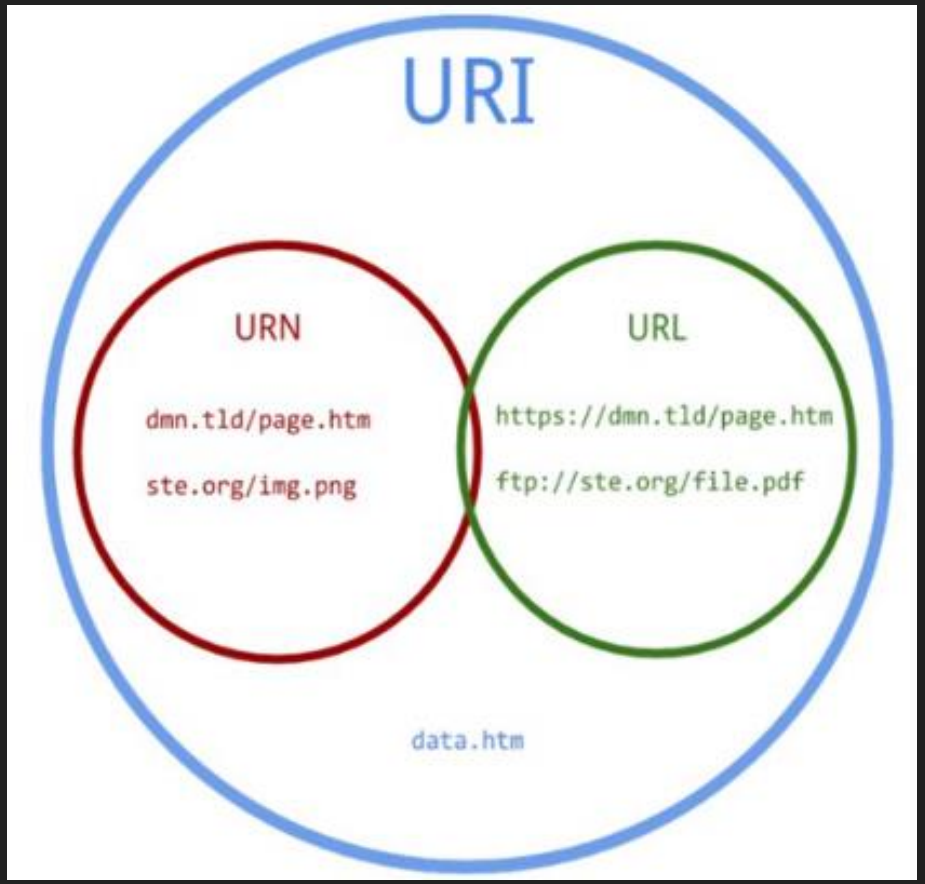
<img src="rick_roll.png"></img>

HTTP

- Πρωτόκολλο στρώματος εφαρμογής
- Κατά σύμβαση ακούει στη θύρα 80
- Είναι stateless (ο server δε χρειάζεται να διατηρεί πληροφορία σχετική με τα λαμβανόμενα requests)
- Μη ασφαλές, τα δεδομένα δεν είναι κρυπτογραφημένα, γι’ αυτό έχει γίνει μετάβαση στο HTTPS (port 443) που είναι το HTTP αλλά με encryption και digital signatures (TLS/SSL)

URI

- Χαρακτηρίζεται ένα string το οποίο ταυτοποιεί μία δικτυακή πηγή χρησιμοποιώντας ένα όνομα, μία τοποθεσία ή και τα δύο
- Περιλαμβάνει δύο υποκατηγορίες, το URN και το γνωστό URL, με το οποίο συχνά συγχέεται


URI
URN
dmn.tld/page.htm
ste.org/img.png
URL
https://dmn.tld/page.htm
ftp://ste.org/file.pdf
data.htm

URL

- Συνιστά το μέσο πρόσβασης σε δικτυακές σελίδες
- Προσδιορίζει το πρωτόκολλο που θα χρησιμοποιηθεί κατά το request (HTTP, HTTPS, FTP, …)
- Προσδιορίζει το domain name ή καλύτερα hostname στο οποίο επιθυμούμε να συνδεθούμε (πχ. w3bs4cks.ntua.gr). Στο σημείο αυτό τονίζουμε πως αντί για domain μπορεί να χρησιμοποιηθεί και η αντίστοιχη IP
- Μπορεί να περιλαμβάνει το port number (σε πρωτόκολλα με default port συνήθως παραλείπεται)
- Περιλαμβάνει το path (web location on a server) που ζητάμε κατά το request
- Μπορεί να περιέχει query strings (πχ ?cmd=sh)

https://helios.ntua.gr/course/view.php?id=867

https: protocol name

helios.ntua.gr: hostname

port: omitted

/course/view.php: path εντός του domain που εξετάζουμε

?id=867: query string

HTTP methods

- Η μέθοδος GET ζητάει από τον server μία αναπαράσταση της εκάστοτε σελίδας (URL) που επιθυμούμε. Με άλλα λόγια τα GET requests ανακτούν (retrieve) μόνο δεδομένα.
- Από την άλλη, η μέθοδος POST υποβάλει μία οντότητα (πχ. ένα password) στο αντίστοιχο URL, συχνά προκαλώντας κάποια αλλαγή στον server και την κατάστασή του (πχ. render a new url).
- Οι παραπάνω είναι οι σημαντικότερες μέθοδοι, ωστόσο υπάρχουν και άλλες, μερικές εκ των οποίων είναι: HEAD, PUT, DELETE, OPTIONS
- Δεν υλοποιούν όλα τα URLs όλες τις μεθόδους!!

Status codes

Τα status codes αποτελούν απάντηση ενός server στο εκάστοτε request που του πραγματοποιούμε, παρέχοντας σημαντική πληροφορία σχετικά με την προσβασιμότητα του URL που ζητάμε. Τα κάτωθι είναι τα σημαντικότερα:

- **200 OK**: επιτυχές request
- **301 Moved Permanently**: Το ζητούμενο URL έχει αλλάξει σε αυτό που κάνει redirect ο browser
- **401 Unauthorized**: Απαιτείται user authentication
- **403 Forbidden**: Απαγορεύεται η προβολή της σελίδας
- **404 Not Found**: Το requested URI δεν εντοπίζεται (και δεν υποστηρίζεται) από τον server
- **500 Internal Server Error**: Πρόκειται συνήθως για unhandled exception από μεριάς server (δηλαδή ο σέρβερ έπαθε ντουβουρτζά)

HTTP Headers

Τα HTTP Headers επιτρέπουν στους client/server να εισαγάγουν περισσότερες πληροφορίες στο αίτημα/απάντηση και είναι ενίοτε απαραίτητα. Ακολουθούν κάποια από τα σημαντικότερα headers:

- Host: host και port προς τον οποίο θα γίνει το request
- Cookie: Ένα cookie που συμπεριλαμβάνεται με το request
- User-Agent: string που ταυτοποιεί όποιον πραγματοποιεί το αίτημα (συνήθως browser)
- Referer: Ουσιαστικά η διεύθυνση (συνήθως URL) από την οποία έγινε ένα request
- Content-Type: Καθορίζει τον τύπο του περιεχομένου σε περίπτωση πχ Post Request ή Response (e.g. application/json, text/html etc.)
- Content-Length: Το μήκος του περιεχομένου σε bytes

Example HTTP Request

```
GET /home.html HTTP/1.1
Host: developer.mozilla.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://developer.mozilla.org/testpage.html
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Mon, 18 Jul 2016 02:36:04 GMT
If-None-Match: "c561c68d0ba92bbeb8b0fff2a9199f722e3a621a"
Cache-Control: max-age=0
```

Source: https://developer.mozilla.org/en-US/docs/Glossary/Request_header

Example HTTP Response

HTTP

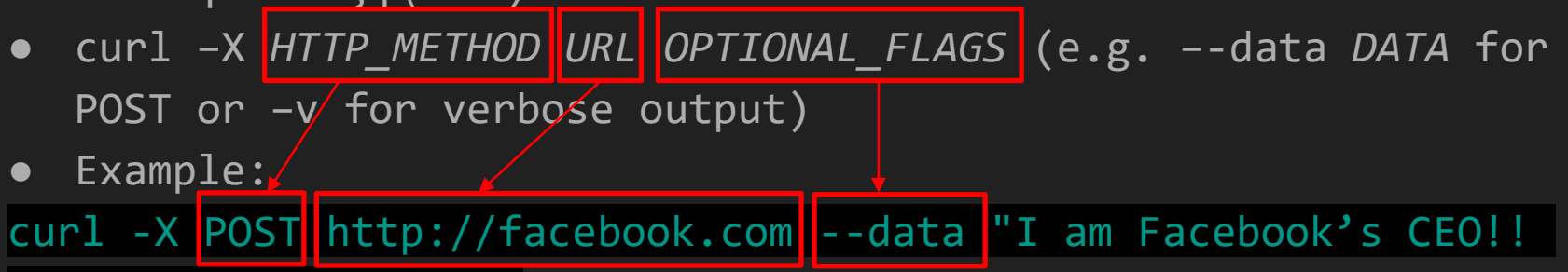
```
200 OK
Access-Control-Allow-Origin: *
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Mon, 18 Jul 2016 16:06:00 GMT
Etag: "c561c68d0ba92bbeb8b0f612a9199f722e3a621a"
Keep-Alive: timeout=5, max=997
Last-Modified: Mon, 18 Jul 2016 02:36:04 GMT
Server: Apache
Set-Cookie: mykey=myvalue; expires=Mon, 17-Jul-2017 16:06:00 GMT; Max-Age=31449600; Path=/;
secure
Transfer-Encoding: chunked
Vary: Cookie, Accept-Encoding
X-Backend-Server: developer2.webapp.scl3.mozilla.com
X-Cache-Info: not cacheable; meta data too large
X-kuma-revision: 1085259
x-frame-options: DENY
```

Source:

https://developer.mozilla.org/en-US/docs/Glossary/Response_header

Το εργαλείο curl

- Απλό και εύχρηστο (είναι όντως) εργαλείο για πραγματοποίηση (http) requests
- Υπάρχει και στα windows και στα linux (`sudo apt install curl`)
- Βασική σύνταξη (linux)
- `curl -X` *`HTTP_METHOD`* *`URL`* *`OPTIONAL_FLAGS`* `(e.g. --data` *`DATA`* `for POST or -v for verbose output)`
- `Example:`



GET request with curl

POST Request with curl

```
┌──(kali㉿kali)-[~]
└─$ curl -X POST http://facebook.com --data "I am Facebook's CEO! Please give me money" -v
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 157.240.9.35:80 ...
* Connected to facebook.com (157.240.9.35) port 80 (#0)
> POST / HTTP/1.1
> Host: facebook.com
> User-Agent: curl/7.88.1
> Accept: */*
> Content-Length: 41
> Content-Type: application/x-www-form-urlencoded
>
< HTTP/1.1 301 Moved Permanently
< Location: https://facebook.com/
< Content-Type: text/plain
< Server: proxygen-bolt
< Date: Tue, 28 Nov 2023 22:54:02 GMT
< Connection: keep-alive
< Content-Length: 0
<
* Connection #0 to host facebook.com left intact
```

Request made by curl

Response from server

Useful Links

- https://tryhackme.com/room/httpindetail→ TryHackMe course on HTTP
- https://curl.se/docs/ → Curl documentation
- https://developer.mozilla.org/en-US/docs/Web/HTTP → Mozilla HTTP web docs
- https://www.freecodecamp.org/news/http-and-everything-you-need-to-know-about-it/ → Freecodecamp’s HTTP guide