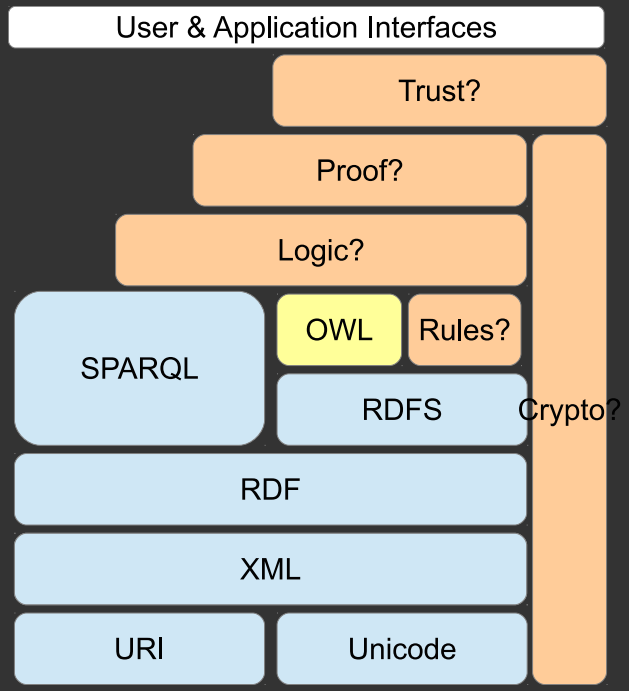
Ενότητα 1: HTTP, Clients και Servers

(Web, το θεμέλιο του Σημασιολογικού Ιστού)

Τα επίπεδα του Σημασιολογικού Ιστού

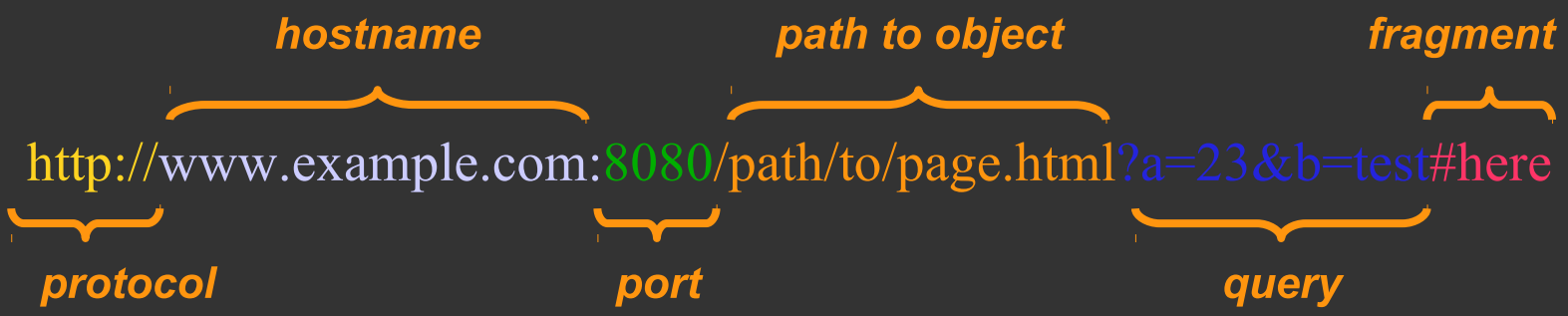


- Θα το βρούμε με πολλές μικρές παραλλαγές
- Θα ασχοληθούμε σε επόμενα εργαστήρια
- Ποια είναι όμως η βάση, στην οποία πατάει το “semantic web layer cake”;
  - Το “Web” στον όρο “Semantic Web”

Το θεμέλιο: Web

- Τα μέρη που συγκροτούν το Web
  - Το πρωτόκολλο HTTP για την ανταλλαγή δεδομένων
  - Web clients και Web Servers
  - Επιπλέον: HTML για ιστοσελίδες (εδώ όμως δεν μας ενδιαφέρει!)
- Κλασσική μορφή client-server
  - Ο client στέλνει μια αίτηση (request) προς τον server...
  - ...και ο server απαντά στον client με μία απόκριση (response)

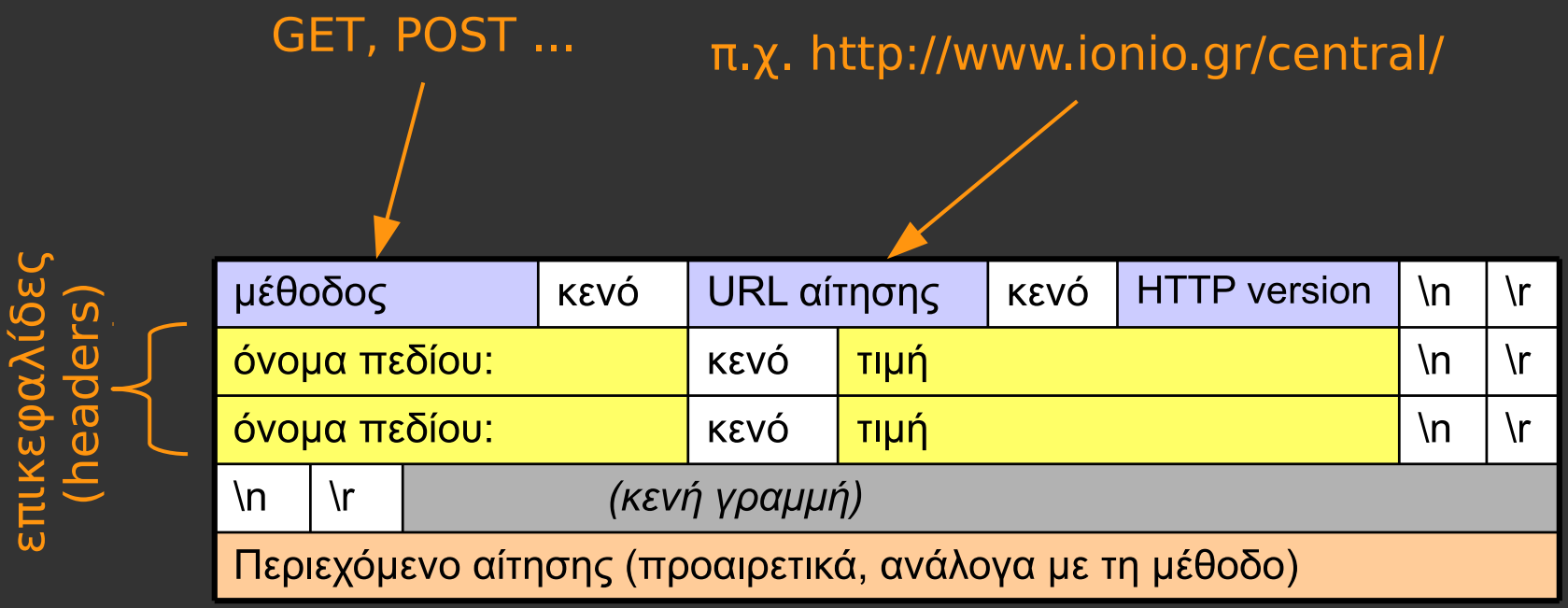
Uniform Resource Locator (URL)



Θα συναντήσουμε αργότερα το URL, αναβαθμισμένο σε URI:

- Uniform Resource Identifier
- Βασική έννοια στον Σημασιολογικό Ιστό

HTTP: Η Αίτηση (Request)



Τα συστατικά της Αίτησης HTTP

- Uniform Resource Locator (URL): επιλέγει “αντικείμενο” –πόρο (resource), οντότητα (entity)...
- Μέθοδος: προσδιορίζει επιθυμητή λειτουργία πάνω στο “αντικείμενο”
- Επικεφαλίδες: εκφράζουν πρόσθετη πληροφορία για εμάς και την πρόθεσή μας
- Περιεχόμενο: δεδομένα που στέλνουμε μαζί με αίτηση, για ορισμένες μεθόδους μόνο

Παράδειγμα αίτησης

```
GET /~mistral/tp/semweb/ HTTP/1.1
Host: di.ionio.gr
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:18.0) G
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
If-Modified-Since: Sun, 03 Feb 2013 18:31:41 GMT
If-None-Match: "b4c029-10b5-4d4d6300b9d40"
```

Μια αίτηση με περιεχόμενο

```
POST /central/gr/directory/19gr/1 HTTP/1.1
Host: www.ionio.gr
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:18.0) G
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.ionio.gr/central/gr/directory
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 14

selection=19gr
```

- Πρόκειται για ένα ζευγάρι τιμών (key:value)
  `key="selection", value="19gr"`
- Προσέξτε τη μέθοδο (POST)

Οι διάφορες μέθοδοι HTTP

Οι μέθοδοι που υποστηρίζονται από όλους!

- GET: ανακτούμε την πληροφορία που σχετίζεται με το αντικείμενο που περιγράφει το URL
  - Μία αίτηση GET σε ένα URL πρέπει να έχει το ίδιο αποτέλεσμα με πολλές αιτήσεις στο ίδιο URL –δεν πρέπει να προκαλεί αλλαγές στον server (idempotent)
  - Μπορούμε να στείλουμε δεδομένα προς τον server με την GET μέσω του URL:

    `https://www.google.com/search?q=http+rfc+2616&ie=`
  - Μία από τις δύο μεθόδους που χρησιμοποιούνται σε φόρμες HTML
- HEAD: όπως και η GET αλλά εδώ ανακτούμε μόνο τις επικεφαλίδες της απόκρισης

Οι διάφορες μέθοδοι HTTP (2)

Οι μέθοδοι που υποστηρίζονται από όλους (συνέχεια)

- POST: ζητάμε η πληροφορία που συνοδεύει την αίτηση να προστεθεί στον server, ο χειρισμός θα γίνει όπως προδιορίζει το URL
  - Πρακτικά χρησιμοποιείται για την αποστολή νέων δεδομένων για επεξεργασία ή την ενημέρωση μιας βάσης
  - Η δεύτερη μέθοδος που χρησιμοποιούν οι φόρμες HTML

Οι διάφορες μέθοδοι HTTP (3)

Μέθοδοι που δεν χρησιμοποιούνται για τη διακίνηση (τυπικών) ιστοσελίδων

- PUT: ζητάμε η πληροφορία που συνοδεύει την αίτηση να αποθηκευτεί στον server ως η νέα τιμή του αντικειμένου που προδιορίζει το URL
  - Idempotent!
- DELETE: ζητάμε ο server να διαγράψει το αντικείμενο που προδιορίζει το URL

Τα GET, POST, PUT και DELETE χρησιμοποιούνται στα διάφορα δημοφιλή web APIs (ανεξάρτητα από τον Σημασιολογικό Ιστό)

- Υπάρχουν και κάποιες άλλες μέθοδοι ακόμα

Οι επικεφαλίδες της Αίτησης, πρακτικά

- Οι περισσότερες επικεφαλίδες της αίτησης τίθενται αυτόματα από την εφαρμογή client που χρησιμοποιούμε
  - από τον web browser (σε κανονική χρήση και μέσω Javascript)
  - από τη βιβλιοθήκη web client που χρησιμοποιεί η εφαρμογή μας (σε οποιαδήποτε γλώσσα)
- Αξίζει όμως να προσέξουμε τις εξής επικεφαλίδες:
  - Accept: με αυτή μπορούμε να ζητήσουμε εναλλακτικές μορφές περιεχομένου (content negotiation)
  - User-Agent: κάποιες φορές θα χρειαστεί να υποκριθούμε ότι είμαστε browser (ενώ είμαστε κάτι διαφορετικό): υπάρχουν servers, οι οποίοι έχουν “αλλεργία” σε αιτήσεις από clients που δεν είναι browsers!

Τα εργαλεία της δουλειάς: Web Clients

- Χρειαζόμαστε Web Clients για
  - Να προσπελάσουμε δεδομένα στον Σημασιολογικό Ιστό
  - Debugging, όταν γράφουμε έναν (application) server
- Είδη Web Clients
  - Command-line tools: curl, wget (κυρίως για debugging και εξερεύνηση πηγών)
  - Βιβλιοθήκες: Για την εφαρμογή μας (θα δουλέψουμε κυρίως σε Python)
  - Browsers: Με τα κατάλληλα plugins (κατά την ανάπτυξη web applications)

curl και wget

```
# Απλή προσπέλαση
wget -q -O - http://www.example.com/page

curl http://www.example.com/page

# Αποστολή στοιχείων με τη μέθοδο POST:
wget -q -O - --post-data "a=1&b=3" http://www.example.com/page

curl --data "a=1&b=3" http://www.example.com/page

curl --data-urlencode 'a=&' --data b=3 http://www.example.com/

# Διαπραγμάτευση επιστρεφόμενου τύπου δεδομένων:
wget -q -O - --header="Accept: text/plain" http://www.example.

curl -H "Accept: text/plain" http://localhost:50007/page

# Εναλλακτικές μέθοδοι
curl -X DELETE http://www.example.com/page/1
```

Δοκιμάστε κι εσείς!

Δοκιμάστε να προσπελάσετε μέσω curl ή wget:

- τις σελίδες των δύο προηγούμενων παραδειγμάτων αίτησης
- τη διεύθυνση `http://dbtune.org/classical/resource/composer/giustini_lodovico`
- δοκιμάστε με το wget ξανά το προηγούμενο, ζητώντας εναλλακτικό περιεχόμενο
  - σε μορφή application/rdf+xml
  - σε μορφή text/rdf+n3

  (για να κάνετε το ίδιο με το curl, προσθέστε το όρισμα -L εκτός από τα άλλα ορίσματα)

Python urllib

```
import urllib

# Απλή προσπέλαση
page = urllib.urlopen("http://www.example.com/page")
text = page.read()
page.close()
print text

# Αποστολή παραμέτρων μέσω GET
# params sent to server
params = { 'a':23,'b':'testαβγδ','c':'+&=' }
# create appropriate params string
paramstr = urllib.urlencode(params)
page = urllib.urlopen("http://www.ex.com/page?"+paramstr)
text = page.read()
page.close()
print text
```

Python urllib (συνέχεια)

```
# Αποστολή παραμέτρων μέσω POST

import urllib

# params sent to server
params = { 'a':23,'b':'testαβγδ','c':'+&=' }

# create appropriate params string
paramstr = urllib.urlencode(params)


page = urllib.urlopen("http://www.ex.com/page",paramstr)
text = page.read()
page.close()

print text
```

Python urllib2

```
# Διαπραγμάτευση επιστρεφόμενου τύπου δεδομένων

import urllib2

req = urllib2.Request("http://www.ex.com/page")
req.add_header('Accept','text/plain')
page = urllib2.urlopen(req)
text = page.read()
page.close()

print text
```

Δοκιμάστε κι εσείς!

Δοκιμάστε να προσπελάσετε μέσω python τα παραδείγματα που δοκιμάσατε μέσω curl και wget!