

Ενότητα 2: HTTP, Clients και Servers (II)

(Web, το θεμέλιο του Σημασιολογικού Ιστού)

HTTP: Η Απόκριση (Response)

αριθμητικός κωδικός
(επιτυχία, σφάλμα, ανακατεύθυνση..)



επικεφαλίδες
(headers)

| | | | | | | | | |
|--------------------------------------|----|---------------|-------------|------|------|-----------------|----|----|
| HTTP version | | κενό | status code | | κενό | κείμενο st.code | \n | \r |
| όνομα πεδίου: | | | κενό | τιμή | | | \n | \r |
| όνομα πεδίου: | | | κενό | τιμή | | | \n | \r |
| \n | \r | (κενή γραμμή) | | | | | | |
| περιεχόμενο (ιστοσελίδα, εικόνα κλπ) | | | | | | | | |

HTTP Response: Status Codes

Οι πιο σημαντικοί κωδικοί:

| | |
|-----|------------------------|
| 200 | OK |
| 301 | Moved Permanently |
| 302 | Found |
| 303 | See Other |
| 304 | Not Modified |
| 400 | Bad Request |
| 403 | Forbidden |
| 404 | Not Found |
| 415 | Unsupported Media Type |
| 500 | Internal Server Error |
| 501 | Not Implemented |

303: Εκτεταμένη χρήση στον Σημασιολογικό Ιστό

Πώς θα δούμε τις επικεφαλίδες της απόκρισης

Λήψη (και) των επικεφαλίδων της απόκρισης

```
curl -s -i 'http://www.example.com/'
```

```
wget -q -O - --server-response 'http://www.example.com/'
```

Δοκιμάστε κι εσείς!

Δοκιμάστε να δείτε τις επικεφαλίδες απόκρισης:

- ▶ της διεύθυνσης <http://www.ionio.gr/central/>
- ▶ της διεύθυνσης <http://www.ionio.gr>
- ▶ της διεύθυνσης http://dbtune.org/classical/resource/composer/giustini_lodovico

Ανακατευθύνσεις (Redirects)

Οι κωδικοί απόκρισης **3xx** δηλώνουν ανακατεύθυνση

- ▶ Ο web client πρέπει να προσπελάσει κάτι άλλο!
- ▶ curl: πώς **θα ακολουθήσει** την ανακατεύθυνση
`curl -L 'http://www.example.com/'`
- ▶ wget: πώς **δεν θα ακολουθήσει** την ανακατεύθυνση

```
wget -q -O - 'http://ex.com' --max-redirect 0
```

Δοκιμάστε κι εσείς!

Δοκιμάστε να δείτε τις ανακατευθύνσεις μέσω curl ή wget για:

- ▶ τη διεύθυνση `http://dbpedia.org/resource/Lodovico_Giustini`
- ▶ την ίδια διεύθυνση με εναλλακτικό περιεχόμενο
 - ▶ σε μορφή `application/rdf+xml`
 - ▶ σε μορφή `text/rdf+n3`

Τα εργαλεία της δουλειάς: Web Servers

Για τη διάθεση δεδομένων του Σημασιολογικού Ιστού

- ▶ Χρήση ενός τυπικού web server (π.χ. Apache)
 - ▶ Εύκολη διάθεση σημασιολογικών δεδομένων
 - ▶ Με ορισμένες ρυθμίσεις η/και τη χρήση scripting
- ▶ Συγγραφή ειδικού application server
 - ▶ Μεγάλη ελευθερία χειρισμού δεδομένων
 - ▶ Συχνά “πίσω” από έναν συμβατικό web server

Τυπικές απαιτήσεις

- ▶ Προσθήκη επικεφαλίδων
- ▶ Ανακατευθύνσεις
- ▶ Διαπραγμάτευση περιεχομένου

CORS!

Cross-Origin Resource Sharing

- ▶ Πολλές εφαρμογές του σημασιολογικού ιστού είναι web applications:
 - ▶ Εκτελούνται στον web browser
 - ▶ Και καταναλώνουν σημασιολογικά δεδομένα από διάφορες πηγές
- ▶ Όμως...
 - ▶ Για λόγους ασφάλειας ο browser δεν μπορεί να προσπελάσει δεδομένα παρά μόνο από το ίδιο site που προέρχεται η web εφαρμογή...
 - ▶ Αυτό αντιβαίνει την ιδέα διασύνδεσης διαφορετικών πηγών του σημασιολογικού ιστού!
- ▶ Μια μοντέρνα αντιμετώπιση: CORS
 - ▶ Κάθε server με σημασιολογικά δεδομένα επιτρέπει τη χρήση τους από τρίτους
 - ▶ Μέσω ειδικής επικεφαλίδας απόκρισης:

Access-Control-Allow-Origin: *

Δοκιμάστε κι εσείς!

Θα χρειαστείτε δύο διαφορετικούς web servers για να δείτε το CORS σε δράση

- ▶ Βήμα 1^ο: Στο site δοκιμών του εργαστηρίου
 - ▶ Δημιουργήστε αρχείο **test.txt** με δοκιμαστικό κείμενο
 - ▶ Αυτό θα αποτελέσει την **πληροφορία από τρίτους**
- ▶ Δοκιμάστε να το προσπελάσετε μέσω curl ή wget

Web application

Τοποθετήστε στον υπολογιστή σας ([corstest.html](#))

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CORS Test</title>
    <script src="corstest.js"></script>
  </head>

  <body>
    <div id="puthere">
      before..
    </div>
  </body>
</html>
```

Web appliction

Τοποθετήστε στον υπολογιστή σας ([corstest.js](#))

```
window.onload = function() {  
    var url = "http://server/~you/test.txt";  
  
    var request = new XMLHttpRequest();  
    request.open("GET",url);  
    request.onload = function() {  
        if (request.status==200) {  
            displayResponse(request.responseText);  
        }  
    };  
    request.send(null);  
};  
  
function displayResponse(responseText) {  
    var putdiv = document.getElementById("puthere");  
    putdiv.appendChild(document.createTextNode(responseText));  
}
```

Προσωρινός web server (στον υπολογιστή σας)

Τοποθετήστε στον υπολογιστή σας ([simplserver.py](#))

```
import sys
import BaseHTTPServer
from SimpleHTTPServer import SimpleHTTPRequestHandler

HandlerClass = SimpleHTTPRequestHandler
ServerClass = BaseHTTPServer.HTTPServer
Protocol = "HTTP/1.0"

if sys.argv[1:]:
    port = int(sys.argv[1])
else:
    port = 8000
server_address = ('127.0.0.1', port)

HandlerClass.protocol_version = Protocol
httpd = ServerClass(server_address, HandlerClass)

sa = httpd.socket.getsockname()
print "Serving HTTP on", sa[0], "port", sa[1], "..."
httpd.serve_forever()
```

Δοκιμάστε κι εσείς!

- ▶ Βήμα 2^ο: Τρέξτε τον προσωρινό server:
`python simpleserver.py`
- ▶ Φορτώστε το `http://localhost:8000/corstest.html` στον browser
 - ▶ (αφού αλλάξετε το URL στο `corstest.js!`)
 - ▶ Βλέπετε το κείμενό σας από το αρχείο `test.txt`;

Δοκιμάστε κι εσείς!

- ▶ Βήμα 3^ο: Στο site δοκιμών, στον φάκελο που περιλαμβάνει το text.txt:
 - ▶ Στο αρχείο `.htaccess` προσθέστε:
`Header set Access-Control-Allow-Origin "*"`
 - ▶ Αυτό οδηγεί τον web server (Apache) να προσθέσει την επικεφαλίδα στην απόκριση
 - ▶ (ελέγξτε το μέσω curl ή wget!)
- ▶ Φορτώστε το `http://localhost:8000/corstest.html` ξανά στον browser
 - ▶ Η προσθήκη της επικεφαλίδας επιτρέπει τη χρήση πληροφορίας τρίτων!

Apache και .htaccess

- ▶ Μπορούμε να βάλουμε αρχείο **.htaccess** σε κάθε φάκελο του `public_html`
- ▶ Με τη βοήθεια του **.htaccess** μπορούμε
 - ▶ να αλλάξουμε το URL της αίτησης σε διαφορετικό URL
 - ▶ να ζητήσουμε ανακατεύθυνση σε διαφορετικό URL
- ▶ Θα πρέπει να το επιτρέπουν οι ρυθμίσεις του Apache

.htaccess και Αλλαγή URL

Μετατροπή π.χ. της διεύθυνσης

`http://www.example.com/~user/resource/test`

στη διεύθυνση

`http://www.example.com/other/script.php?q=test`

```
RewriteEngine On
```

```
RewriteBase /~user/resource
```

```
RewriteRule ^(.*) /other/script.php?q=$1
```

.htaccess και Ανακατεύθυνση

Ειδοποίηση του client για ανακατεύθυνση (303 See Other) από π.χ. τη διεύθυνση

`http://www.example.com/~user/resource/test`

στη διεύθυνση

`http://www.example.com/~user/data/test1.txt`

```
RewriteEngine On
```

```
RewriteBase /~user/resource
```

```
RewriteRule ^(.*) /~user/data/$1.txt [R=303]
```

Δοκιμάστε κι εσείς!

Στον server δοκιμών υπάρχει (από το παλιό εργαστήριο των Τεχνολογιών Διαδικτύου) web service στη διεύθυνση [/~hayman11/jsonsender.php?q=courseinfo&code=ΚΩΔ](http://~hayman11/jsonsender.php?q=courseinfo&code=ΚΩΔ) που δίνει πληροφορίες σχετικούς με μαθήματα

- ▶ ΚΩΔ = ΗΥ100, ΗΥ010, ΘΕ100, ΜΘ100 και ΜΘ110 (ελληνικοί χαρακτήρες)

Δοκιμάστε πρώτα με αλλαγή URL και μετά με ανακατεύθυνση να στέλνεται στη διεύθυνση αυτή τα URL της μορφής:

`/~you/courses/ΚΩΔ`

- ▶ Ελέγξτε μέσω wget ή curl. Ποια η διαφορά;

Ανακατεύθυνση και Διαπραγμάτευση Περιεχομένου

Μόνο αν ισχύει η συνθήκη **RewriteCond**!

```
RewriteEngine On
```

```
RewriteBase /~user/resource
```

```
RewriteCond %{HTTP_ACCEPT} text/plain
```

```
RewriteRule ^(.*) /~user/data/$1.txt [R=303]
```