

Τεχνολογίες Εφαρμογών Διαδικτύου

Πρωτόκολλο HTTP (Hypertext Transfer Protocol)
Η λειτουργία του Web Server και του Web Browser

Δρ. Ι. Χαμόδρακας
Μέλος του Εργαστηριακού Διδακτικού Προσωπικού
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Παγκόσμιος Ιστός: Ιστορικά Στοιχεία

- ▶ Ο Παγκόσμιος Ιστός προτάθηκε στο ινστιτούτο CERN το 1989 από τον Tim Berners-Lee.
- ▶ Μέχρι το τέλος του 1990 ο Berners-Lee υλοποίησε τον πρώτο server και έναν browser γραμμής εντολών. Ο server και ο browser επικοινωνούσαν με την αρχική έκδοση του πρωτοκόλλου HTTP που σχεδίασε για αυτό το σκοπό.
- ▶ Μέχρι το 1993 υπήρχαν περίπου 50 διαφορετικοί δικτυακοί τόποι με εγκατεστημένους HTTP servers...
- ▶ Η ανάπτυξη που ακολούθησε ήταν εκρηκτική.



Δομικά στοιχεία του Ιστού

- ▶ Γλώσσα σήμανσης εγγράφων υπερκειμένου (HTML: HyperText Markup Language)
 - Μια γλώσσα για τη μορφοποίηση εγγράφων υπερκειμένου. Το υπερκείμενο είναι κείμενο που περιέχει συνδέσμους σε άλλα κείμενα.
- ▶ Σχήμα ενιαίας διευθυνσιοδότησης των πόρων (URI: Uniform Resource Identifier / URL: Uniform Resource Locator)
 - Κοινώς η διεύθυνση παγκόσμιου ιστού
- ▶ Πρωτόκολλο μεταφοράς υπερκειμένου (HTTP: HyperText Transfer Protocol)
 - Δικτυακό πρωτόκολλο στο επίπεδο εφαρμογής

URI: Ενιαίο Αναγνωριστικό Πόρων

`scheme://host[:port]/path/.../[;url-params]/[?query-string][#anchor]`

- ▶ **scheme:** ορίζει το πρωτόκολλο
- ▶ **host:** η διεύθυνση IP (αριθμητική ή domain name)
- ▶ **port:** η θύρα TCP (default:80) όπου “ακούει” ο Web Server
- ▶ **path:** μονοπάτι εντός του Web Server που αντιστοιχεί σε μονοπάτι στο σύστημα αρχείων
- ▶ **url-params:** (σπάνια χρησιμοποιούμενες) παράμετροι (ζεύγος ονόματος-τιμής)
- ▶ **query-string:** παράμετροι (ζεύγη ονομάτων-τιμών) που χωρίζονται με το ampersand
- ▶ **anchor:** αναφορά σε συγκεκριμένη θέση του εγγράφου που ζητείται μέσω κατάλληλης ετικέτας

Παράδειγμα:

`http://www.di.uoa.gr/ted/lecture;number=2?title=http&subtitle=uri#label`

Το πρωτόκολλο HTTP

- ▶ Πρωτόκολλο επικοινωνίας των Web Servers με τους Web Browsers (HTTP Server / Client αντίστοιχα). Τρέχον πρότυπο: HTTP/1.1 (σύντομα HTTP/2)
- ▶ Βρίσκεται στο επίπεδο εφαρμογής στη σουίτα TCP/IP και χρησιμοποιεί το TCP για την μετάδοση μηνυμάτων στο δίκτυο
- ▶ **Λειτουργία Request-Response**
 - Ο Browser στέλνει ένα μήνυμα HTTP request (αίτηση) στον Server, ο οποίος απαντά με μήνυμα HTTP response (απόκριση)
 - Τα μηνύματα αποτελούνται από έναν αριθμό γραμμών επικεφαλίδας (message headers, που ακολουθείται από μια κενή γραμμή και στη συνέχεια από το σώμα του μηνύματος (message body)

Χαρακτηριστικά HTTP

- ▶ Ανάμεσα στον browser και τον server μπορεί να υπάρχουν ενδιάμεσοι κόμβοι που ονομάζονται proxies.
- ▶ Το πρωτόκολλο HTTP δεν διατηρεί πληροφορίες κατάστασης, δηλαδή είναι stateless. Κάθε αλληλεπίδραση client-server είναι ξεχωριστή.
- ▶ Το πρωτόκολλο FTP είναι stateful δηλαδή ο server διατηρεί την κατάσταση της αλληλεπίδρασης με τον client καθ' όλη τη διάρκεια της επικοινωνίας. Επομένως, μια σειρά request-response αλληλεπιδράσεων ομαδοποιείται σε μία συνεδρία (session).
- ▶ Στις εφαρμογές Web χρησιμοποιούνται τα cookies ως εναλλακτικός μηχανισμός διατήρησης κατάστασης και διαχείρισης συνεδριών. (+ url-rewriting)

Δομή HTTP request

```
METHOD /path-to-resource HTTP/version-number
```

```
Header-Name-1: value
```

```
Header-Name-2: value
```

```
[προαιρετικό σώμα request]
```

- ▶ Παράδειγμα HTTP request για την επίσκεψη σε μία ιστοσελίδα

```
GET /~ihamod/ted/course_description.html HTTP/1.1
```

```
Host: users.uoa.gr
```

- ▶ Στην πρώτη γραμμή αναγράφεται η μέθοδος αίτησης (συνήθως GET ή POST), το μονοπάτι από το URI και η έκδοση του πρωτοκόλλου HTTP. Στη συνέχεια ακολουθούν γραμμές επικεφαλίδας π.χ. προσδιορισμός του host και μετά από μια κενή γραμμή (<CR> <LF>) προαιρετικά το σώμα του μηνύματος.

Δομή HTTP response (1)

HTTP/version-number status-code explanation

Header-Name-1: value

Header-Name-2: value

[σώμα response]

- ▶ Παράδειγμα απόκρισης στην αίτηση της προηγούμενης διαφάνειας

HTTP/1.1 200 OK

Content-Type: text/html

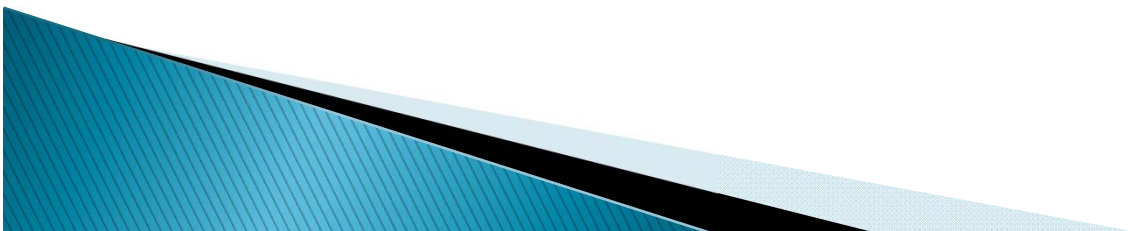
Date: Tue, 24 Feb 2015 17:03:39 GMT

. . .

```
<html> <head> <title> ΥΣ14 - Τεχνολογίες Εφαρμογών Διαδικτύου  
</title> </head> <body> <p> . . . </html>
```


Δομή HTTP response (2)

- ▶ Στην πρώτη γραμμή περιέχεται η έκδοση του HTTP ακολουθούμενη από ένα τριψήφιο κωδικό κατάστασης και μια σύντομη εξήγηση του κωδικού κατάστασης
- ▶ Ακολουθούν γραμμές επικεφαλίδας και μετά την κενή γραμμή ο κώδικας HTML που αποτελεί το σώμα του μηνύματος response.



Μέθοδοι HTTP request / GET request

- ▶ GET, HEAD και POST. Σπανιότερα: PUT, DELETE, TRACE, OPTIONS και CONNECT
- ▶ Μέθοδος GET:
 - Η απλούστερη μέθοδος αίτησης. Π.χ. όταν πληκτρολογούμε τη διεύθυνση του Τμήματος στον browser δημιουργείται μια αίτηση GET που αποστέλλεται στον server του Τμήματος.
 - Το μήνυμα της αίτησης GET δεν έχει σώμα
 - Αν χρειάζεται να περάσουν παράμετροι με την αίτηση GET αυτό γίνεται μέσω του query string κομματιού του URI.

Π.χ.: GET /search?q=test HTTP/1.1

Host: www.google.gr

User-Agent: Mozilla/5.0 (Windows NT 5.2; WOW64; rv:35.0)

Μέθοδοι HTTP request / POST request

- ▶ Η αίτηση POST περιέχει σώμα μηνύματος όπου αναγράφονται τυχόν παράμετροι της αίτησης. Οι παράμετροι δεν φαίνονται στο URI.

```
POST /search HTTP/1.1  
Host: www.google.gr  
User-Agent: Mozilla/5.0
```

```
q=test
```

- ▶ Συνήθως η απόκριση παραμένει ίδια, αν και αυτό δεν είναι απαραίτητο.



Μέθοδοι HTTP request / HEAD request

- ▶ HEAD: Όπως η μέθοδος GET με τη διαφορά ότι το μήνυμα response δεν περιλαμβάνει το σώμα του (δηλαδή τον κώδικα HTML).
- ▶ Η μέθοδος HEAD χρησιμοποιήθηκε ιστορικά για την υλοποίηση της προσωρινής αποθήκευσης (caching) ιστοσελίδων.
 - Ο browser συγκρίνει το χρόνο αποθήκευσης του αντιγράφου μιας ιστοσελίδας που βρίσκεται στην προσωρινή μνήμη με την ημερομηνία τελευταίας αλλαγής της ιστοσελίδας.
 - Αν είναι προγενέστερη τότε ζητείται το νέο περιεχόμενο της σελίδας μέσω π.χ. μιας GET αίτησης.
- ▶ Σήμερα χρησιμοποιούνται πιο προηγμένες μέθοδοι προσωρινής αποθήκευσης

Κωδικοί κατάστασης (Status Codes)

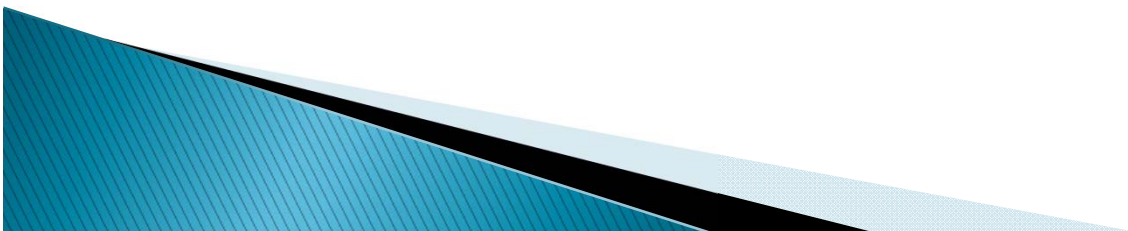
1xx: παρέχουν πληροφορία για τη συνέχιση της επεξεργασίας μιας αίτησης (π.χ. όταν μια αίτηση έχει αποσταλεί μερικώς, 100 Continue)

2xx: ενδεικτικοί επιτυχούς εξυπηρέτησης της αίτησης (π.χ. 200 OK)

3xx: δίνουν εντολή στον browser να εκτελέσει κάποια ενέργεια (π.χ. ανακατεύθυνση σε μια άλλη διεύθυνση)

4xx: αναπαριστούν σφάλματα κατά την αίτηση (π.χ. 401 Not Authorized ή 400 Bad Request)

5xx: αναπαριστούν σφάλματα στην πλευρά του server (π.χ. 500 Internal Server Error ή 501 Not Implemented)



Επικεφαλίδες μηνυμάτων HTTP

(1)

- ▶ **Γενικές επικεφαλίδες** (εφαρμόζονται τόσο στο request όσο και στο response):
 - **Date**: προσδιορίζει ημερομηνία και ώρα δημιουργίας του μηνύματος
 - **Connection**: προσδιορίζει αν ο client ή ο server που δημιούργησε το μήνυμα σκοπεύει να κρατήσει την TCP σύνδεση ανοιχτή
 - **Warning**: προσδίδει πληροφορίες για την ανίχνευση προβλημάτων σε μορφή αναγνώσιμη από τον άνθρωπο
- ▶ **Επικεφαλίδες Request**:
 - **User-Agent**: προσδιορίζει το λογισμικό του browser που έκανε την αίτηση
 - **Host**: προσδιορίζει το όνομα του domain ώστε να είναι δυνατό ένας server να υποστηρίζει πολλαπλά ονόματα domain
 - **Referer**: συνήθως προσδιορίζει την ιστοσελίδα από την οποία ξεκίνησε η αίτηση με την ενεργοποίηση ενός συνδέσμου (link) ως URI
 - **Authorization**: περιλαμβάνει δεδομένα για την επαλήθευση της ταυτότητας και την αδειοδότηση ενός χρήστη όταν συγκεκριμένοι πόροι στον server το απαιτούν

Επικεφαλίδες μηνυμάτων HTTP

(2)

► Επικεφαλίδες Response:

- Location: προσδιορίζει ένα URI στο οποίο πρέπει να ανακατευθυνθεί ο browser και συνοδεύει πάντοτε τους κωδικούς κατάστασης 301 και 302
- WWW-Authenticate: υποδεικνύει την προστατευμένη περιοχή για την οποία ο χρήστης στον browser πρέπει να δώσει στοιχεία επαλήθευσης ταυτότητας (username και password)
- Server: προσδιορίζει το λογισμικό του server

► Επικεφαλίδες Οντότητας (entity headers, περιγράφουν το σώμα του μηνύματος ή τον στόχο της αίτησης)

- Content-Type: ο τύπος MIME του περιεχομένου του σώματος π.χ. text/plain, text/html, application/octet-stream
- Content-Length: το μέγεθος του σώματος σε bytes
- Last-Modified: προσδιορίζει την ημερομηνία τελευταίας τροποποίησης του περιεχομένου που μεταδίδεται στο σώμα του μηνύματος

Υποστήριξη συνεδριών με cookies (1)

- ▶ Οι εφαρμογές Ιστού απαιτούν τη δημιουργία συνεδριών (π.χ. για το καλάθι αγορών σε μια ηλεκτρονική αγορά).
- ▶ Το πρωτόκολλο HTTP είναι stateless και επομένως δεν το υποστηρίζει.
- ▶ **Cookies**
 - Το πρωτόκολλο HTTP/1.1. παρέχει τις επικεφαλίδες Set-Cookie και Cookie
 - Set-Cookie: επικεφαλίδα response που στέλνει ο server στον browser και στην οποία προσδιορίζεται το αναγνωριστικό της συνεδρίας (session id) / άλλες πληροφορίες κατάστασης.

```
Set-Cookie: <name>=<value>  
           [; Max-Age=<value>] [; Expires=<value>]  
           [; Path=<path>] [; Domain=<domain name>]  
           [; Version=<version>]
```

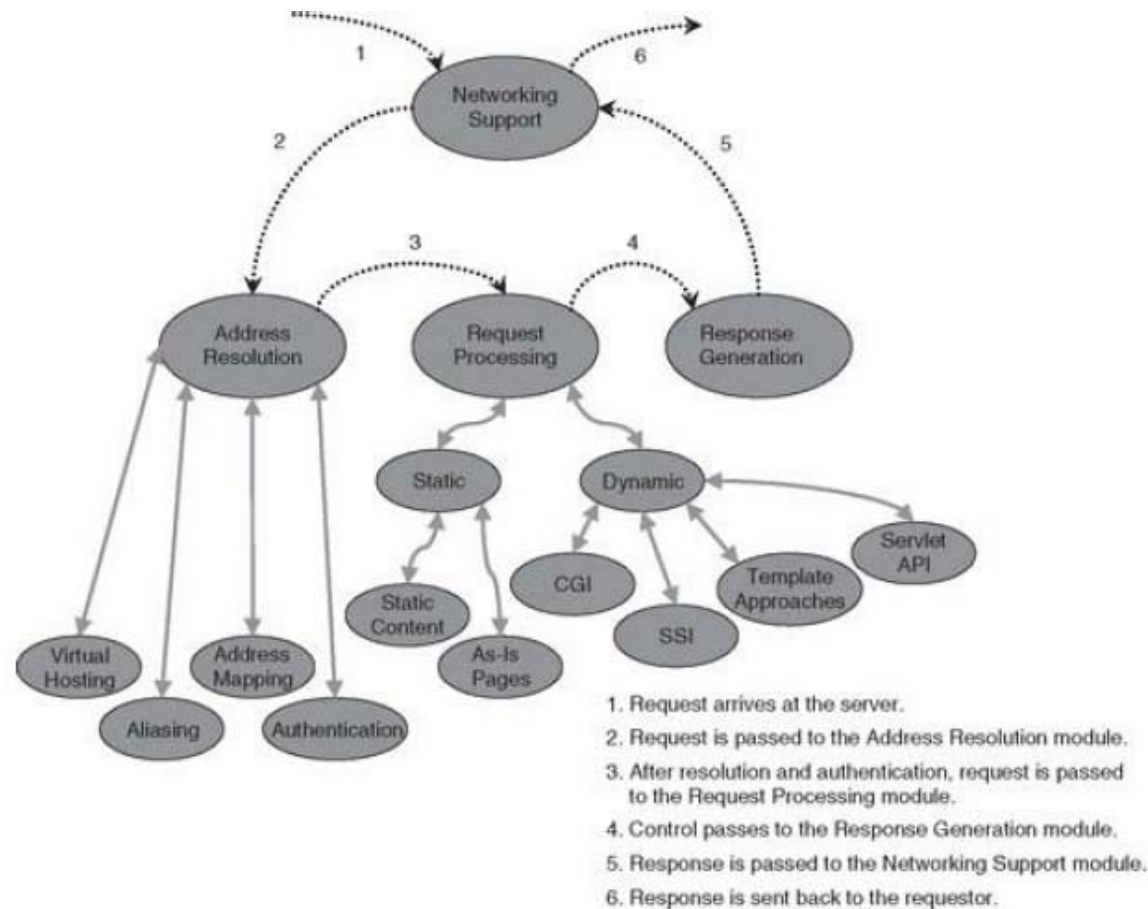
Υποστήριξη συνεδριών με cookies (2)

- Το πεδίο `Max-Age` προσδιορίζει το χρόνο ζωής του cookie σε δευτερόλεπτα, το πεδίο `Expires` αποτελεί έναν παλιότερο μηχανισμό για το ίδιο πράγμα προσδιορίζοντας την ημερομηνία λήξης του cookie. Το πεδίο `Version` προσδιορίζει ποιο από τα 2 ακολουθείται (αν δεν υπάρχει default είναι το `Expires`)
- Τα πεδία `Path` και `Domain` προσδιορίζουν το URI με το οποίο σχετίζεται το cookie.
- **Cookie**: επικεφαλίδα request που στέλνει ο browser στην οποία προσδιορίζεται το session id και άλλες πληροφορίες κατάστασης ώστε να συσχετιστούν οι αιτήσεις με συγκεκριμένες συνεδρίες
- Αφού σταλεί το μήνυμα που περιέχει την επικεφαλίδα `Set-Cookie` από τον server, ο browser δημιουργεί και αποθηκεύει το cookie είτε στη μνήμη (αν δεν έχει οριστεί χρόνος ζωής οπότε το cookie ισχύει όσο χρόνο ο browser είναι ανοιχτός, τη λεγόμενη συνεδρία browser) ή διαφορετικά σε κατάλληλη θέση στο δίσκο.
- **Poison cookies**: μπορούν να χρησιμοποιηθούν για την καταγραφή της κίνησης των χρηστών, την παράνομη συλλογή και τη διανομή προσωπικών δεδομένων και την εκμετάλλευση κενών ασφαλείας των browser.

Μόνιμες συνδέσεις (persistent) και HTTP

- ▶ Το πρωτόκολλο HTTP μπορεί να χρησιμοποιήσει τόσο μόνιμες όσο και μη-μόνιμες συνδέσεις TCP.
- ▶ Πρόκειται περί stateless πρωτοκόλλου επομένως δεν είναι απαραίτητες οι μόνιμες συνδέσεις TCP.
- ▶ Ωστόσο το κόστος και η καθυστέρηση εγκαθίδρυσης σύνδεσης TCP είναι μεγάλα και γι' αυτό διατηρείται μία σύνδεση για περισσότερα από ένα ζεύγη αιτήσεων-αποκρίσεων.
- ▶ Στο πρωτόκολλο HTTP/1.1 οι συνδέσεις είναι μόνιμες εκτός αν προσδιοριστεί διαφορετικά από την επικεφαλίδα `Connection: close`
- ▶ Υποστηρίζεται το pipelining των αιτήσεων: ο browser στέλνει πολλαπλές αιτήσεις χωρίς να περιμένει την απόκριση του server. Ο server στέλνει τις αποκρίσεις με τη σειρά που τις παρέλαβε. Ο browser θα πρέπει να διατηρεί ουρές αιτήσεων, να παρακολουθεί τις αποκρίσεις και να ξαναστέλνει τις αιτήσεις που βρίσκονται στην ουρά αν διακοπεί η σύνδεση.

Web Servers: Λειτουργία (1)



Διάγραμμα ροής λειτουργίας του Web Server
(από το βιβλίο Shklar & Rosen, Web Application Architecture)

Web Servers: Λειτουργία (2)

1. Η αίτηση φτάνει στον server
2. Η αίτηση προωθείται στη μονάδα διευθυνσιοδότησης που εκτελεί λειτουργίες προ-επεξεργασίας (virtual hosting, απεικόνιση διεύθυνσης, επαλήθευση ταυτότητας)
3. Μετά την προ-επεξεργασία η αίτηση προωθείται στη μονάδα επεξεργασίας
4. Ο έλεγχος περνά στη μονάδα δημιουργίας της απόκρισης
5. Η απόκριση προωθείται στη μονάδα δικτυακής υποστήριξης
6. Η απόκριση αποστέλλεται πίσω στον client.

Web Servers: προ-επεξεργασία αίτησης

- ▶ Virtual hosting: εξετάζεται η επικεφαλίδα `Host` της αίτησης για να προσδιοριστεί το `domain` στο οποίο απευθύνεται αν ο server υποστηρίζει πολλαπλά ονόματα `domain`.
- ▶ Απεικόνιση διεύθυνσης: προσδιορίζεται αν η αίτηση απευθύνεται σε στατικό ή δυναμικό περιεχόμενο ανάλογα με το `URI` και τις ρυθμίσεις του server καθώς και γίνεται αντιστοίχιση με κώδικα που βρίσκεται στο σύστημα αρχείων (HTML, Java Servlet, CGI, κλπ)
- ▶ Επαλήθευση ταυτότητας: αν η αίτηση ζητά έναν προστατευμένο πόρο πραγματοποιείται επαλήθευση ταυτότητας βάσει των διαπιστευτηρίων που περιλαμβάνονται (username, password)

Web Servers: επεξεργασία αίτησης

- ▶ Εξετάζονται οι επικεφαλίδες του μηνύματος της αίτησης:
 - Διατήρηση μόνιμης σύνδεσης / μη-μόνιμη σύνδεση
 - Μόνιμη σύνδεση: ουρά αιτήσεων / αποκρίσεων ανά client (FIFO)
 - Γίνεται ανάκτηση του στατικού περιεχομένου (HTML, XML, κείμενο, εικόνες, κλπ) από το σύστημα αρχείων ή εκτελείται ο κώδικας για την παραγωγή του δυναμικού περιεχομένου στον οποίο προωθείται το μήνυμα της αίτησης (CGI, Java Servlet, ASP, κλπ) ανάλογα με το URI.

Web Servers: δημιουργία απόκρισης

- ▶ Δημιουργείται το μήνυμα απόκρισης (επικεφαλίδες, σώμα του μηνύματος)
- ▶ Συχνά το δυναμικό περιεχόμενο είναι αποθηκευμένο σε προσωρινή μνήμη (caching)

```
HTTP/1.1 200 OK
Date: Thu, 26 Feb 2015, 17:00:01 GMT
Content-Type: text/html
Content-Length: 265
Server: Apache/2.2
```

```
<HTML>
<HEAD>
. . .
</HTML>
```

Web Servers: εκτέλεση / ρυθμίσεις

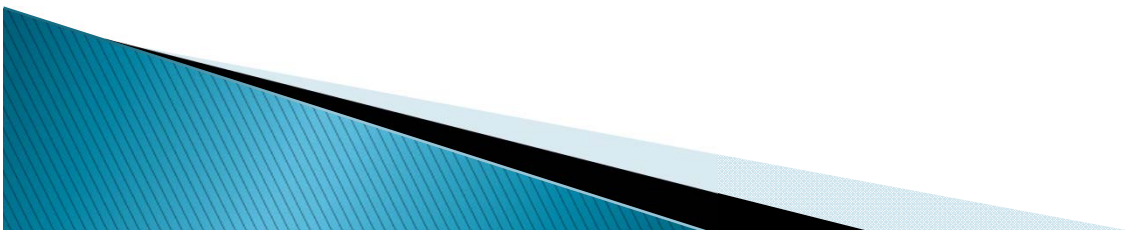
- ▶ Ο Web Server είναι ένα σύνολο από νήματα / διεργασίες.
- ▶ Ορισμένα νήματα «ακούνε» συγκεκριμένες θύρες για την παραλαβή των αιτήσεων ενώ άλλα αναλαμβάνουν την επεξεργασία των αιτήσεων.
- ▶ Η ρύθμιση του server περιλαμβάνει τον ορισμό της θύρας TCP όπου αποστέλλονται οι αιτήσεις, το μέγιστο χρονικό διάστημα για τη διατήρηση μίας σύνδεσης TCP (keep-alive), το μέγιστο αριθμό των ταυτόχρονων αιτήσεων, τον αριθμό αιτήσεων ανά σύνδεση, τη ρύθμιση των εικονικών κόμβων (virtual hosts), κλπ.
- ▶ Η ρύθμιση των virtual hosts απαιτεί την αντίστοιχη ρύθμιση του Domain Name Server.

Web Browsers: Λειτουργία (1)

- ▶ Οι web browsers (φυλλομετρητές ιστού) είναι εφαρμογές που παίζουν το ρόλο του web client.
- ▶ Web clients: browsers / ενδιάμεσοι κόμβοι (proxies) / πράκτορες (agents) που εκτελούν αυτοματοποιημένες λειτουργίες.
- ▶ Βασικές λειτουργίες:
 - Δημιουργία και υποβολή αιτήσεων στους servers μέσω ενεργοποίησης υπερσυνδέσμων, πληκτρολόγησης URI, υποβολής στοιχείων από φόρμες, κλπ.
 - Αποδοχή αποκρίσεων και επεξεργασία τους
 - Εμφάνιση της οπτικής αναπαράστασης της απόκρισης σε περιβάλλον γραφικής διεπαφής

Web Browsers: Λειτουργία (2)

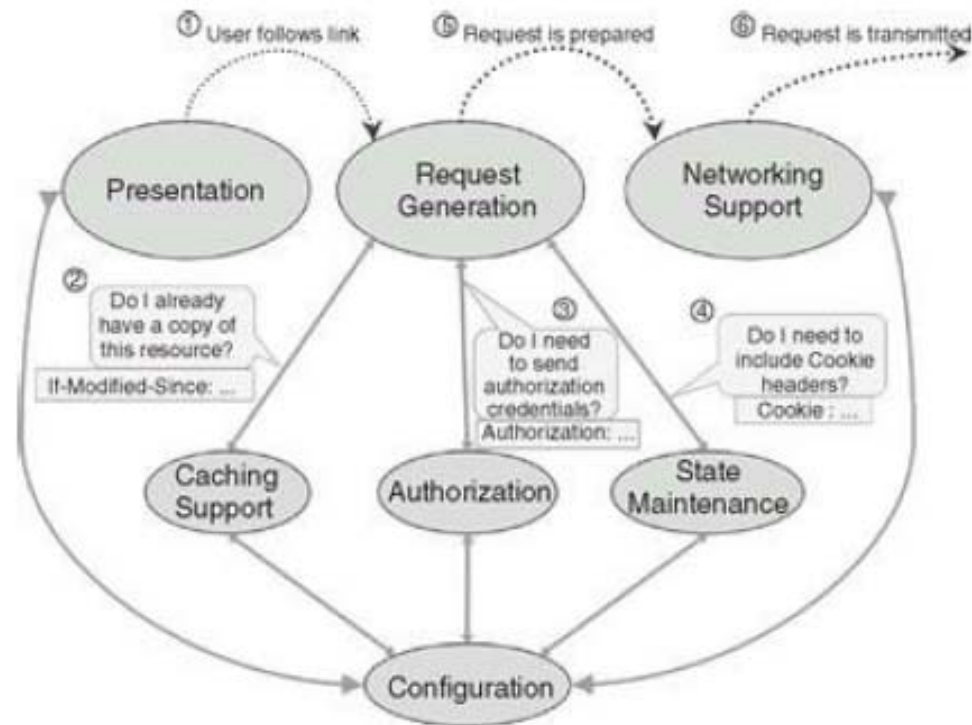
- ▶ Άλλες λειτουργίες:
 - Αποθήκευση ιστοσελίδων στην προσωρινή μνήμη (caching)
 - Επαλήθευση ταυτότητας και αδειοδότηση για την πρόσβαση σε πόρους ιστού (authentication & authorization)
 - Διατήρηση κατάστασης / συνεδριών μέσω των cookies
 - Ανάληψη ενεργειών βάσει των επικεφαλίδων / των κωδικών κατάστασης των μηνυμάτων response (π.χ. ανακατεύθυνση)
 - Απεικόνιση σύνθετων αντικειμένων (π.χ. image/jpg, application/pdf). Η απεικόνιση μπορεί να γίνεται με τη χρήση εξωτερικών βοηθητικών εφαρμογών / plug-ins.
 - Αντιμετώπιση σφαλμάτων



Web browsers: λειτουργικές μονάδες

- ▶ **Μονάδα διεπαφής χρήστη:** παρουσίαση, απεικόνιση της απόκρισης και αλληλεπίδραση με το χρήστη.
- ▶ **Μονάδα δημιουργίας αιτήσεων:** κατασκευή μηνύματων request.
- ▶ **Μονάδα επεξεργασίας αποκρίσεων:** ανάλυση και διερμηνεία της απόκρισης, προώθηση της εξόδου στη μονάδα διερμηνείας περιεχομένου.
- ▶ **Μονάδα δικτυακής επικοινωνίας**
- ▶ **Μονάδα προσωρινής μνήμης (caching):** αποθήκευση ιστοσελίδων στην προσωρινή μνήμη.
- ▶ **Μονάδα διατήρησης κατάστασης:** διαχείριση / αποθήκευση cookies.
- ▶ **Μονάδα επαλήθευσης ταυτότητας / αδειοδότησης:** αποθήκευση διαπιστευτηρίων για την επαναχρησιμοποίηση σε πολλαπλές αιτήσεις.
- ▶ **Μονάδα διερμηνείας περιεχομένου:** εξετάζει τις επικεφαλίδες κωδικοποίησης της απόκρισης και αναλόγως την προωθεί σε κατάλληλη μονάδα επεξεργασίας ή σε βοηθητική εφαρμογή. Αν το περιεχόμενο αναφέρεται σε άλλους πόρους ιστού, προωθεί τις πληροφορίες στη μονάδα δημιουργίας αιτήσεων για την ανάκτηση τους. Προωθεί τους επεξεργασμένους πόρους στη μονάδα διεπαφής.
- ▶ **Μονάδα ρυθμίσεων**

Web browsers: ροή δημιουργίας αιτήσεων (1)



Διάγραμμα ροής δημιουργίας αιτήσεων
(από το βιβλίο Shklar & Rosen, Web Application Architecture)

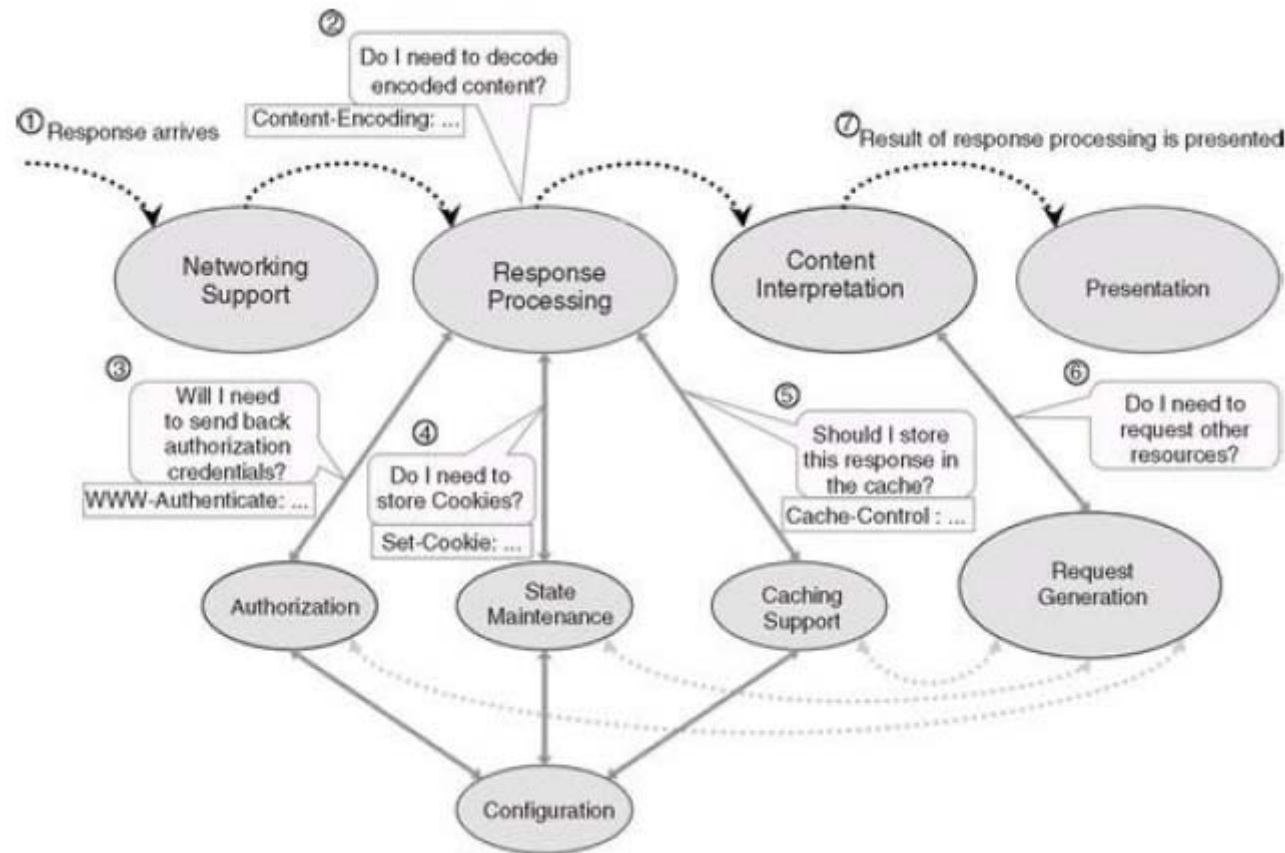
Web browsers: ροή δημιουργίας αιτήσεων (2)

- ▶ Η επεξεργασία ξεκινά από τη γραφική διεπαφή χρήστη (presentation) που λειτουργεί βάσει ενός *μοντέλου γεγονότων* (π.χ. πληκτρολόγηση URI, συμπλήρωση φόρμας σε ανοιχτή σελίδα, επιλογή υπερσυνδέσμου) (1).
- ▶ Ο έλεγχος και οι πληροφορίες για τη δημιουργία της αίτησης περνά στη *μονάδα δημιουργίας αιτήσεων* η οποία επικοινωνεί με τη *μονάδα προσωρινής μνήμης* (2), με τη *μονάδα αδειοδότησης* (3) και με τη *μονάδα διαχείρισης κατάστασης* (4).
- ▶ Κατά το βήμα (2), η μονάδα προσωρινής μνήμης ελέγχει αν υπάρχει αποθηκευμένη εγγραφή που δεν έχει λήξει και την προωθεί στη διεπαφή χρήστη.
- ▶ Αν έχει λήξει προσθέτει στην αίτηση την επικεφαλίδα *If-Modified-Since* σύμφωνα με την οποία ο server θα στείλει μήνυμα απόκρισης που θα περιέχει είτε τη νέα έκδοση του πόρου (200) είτε μια αίτηση με τον κωδικό κατάστασης 304 Not Modified βάσει του οποίου η αποθηκευμένη εγγραφή θα προωθηθεί στη διεπαφή χρήστη.(HTTP/1.1)

Web browsers: ροή δημιουργίας αιτήσεων (3)

- ▶ Κατά το βήμα (3) εν συνεχεία η μονάδα αδειοδότησης ελέγχει αν ο ζητούμενος πόρος ανήκει σε περιοχή για την οποία ο χρήστης είχε στο παρελθόν παράσχει διαπιστευτήρια επαλήθευσης ταυτότητας και αν αυτά είναι αποθηκευμένα προσθέτει τις αντίστοιχες επικεφαλίδες στην αίτηση. Αν δεν είναι αποθηκευμένα, ο έλεγχος περνά στη διεπαφή χρήστη που ζητά την παροχή τους.
- ▶ Κατά το βήμα (4) η μονάδα διαχείρισης κατάστασης ελέγχει αν η αίτηση ανήκει σε συνεδρία ή αν υπάρχουν πληροφορίες κατάστασης για τον πόρο βάσει των αποθηκευμένων cookies και αποφασίζει αν οι κατάλληλες πληροφορίες των cookies θα συμπεριληφθούν στην αίτηση μέσω των κατάλληλων επικεφαλίδων.
- ▶ Τέλος, ολοκληρώνεται η δημιουργία της αίτησης από τη μονάδα δημιουργίας αιτήσεων και προωθείται στη μονάδα δικτυακής επικοινωνίας (5), η οποία την αποστέλλει στον server (6).

Web browsers: ροή επεξεργασίας αποκρίσεων (1)



Διάγραμμα ροής επεξεργασίας αποκρίσεων
(από το βιβλίο Shklar & Rosen, Web Application Architecture)

Web browsers: ροή επεξεργασίας αποκρίσεων (2)

- ▶ Η απόκριση παραλαμβάνεται από τη μονάδα δικτυακής επικοινωνίας (1) και προωθείται στη μονάδα επεξεργασίας αποκρίσεων (2).
- ▶ Αν ο κωδικός κατάστασης της απόκρισης είναι 401 Not Authorized προωθείται ο έλεγχος στη μονάδα αδειοδότησης (3) η οποία ενεργοποιεί τη διεπαφή χρήστη που ζητά διαπιστευτήρια από το χρήστη και γίνεται εκ νέου υποβολή της αρχικής αίτησης στην οποία έχουν προστεθεί οι επικεφαλίδες αδειοδότησης (Authorization).
- ▶ Αν η απόκριση περιέχει επικεφαλίδες Set-Cookie (ή Set-Cookie2) καλείται η μονάδα διατήρησης κατάστασης για την αποθήκευση του cookie στη μνήμη ή το δίσκο (4).
- ▶ Αν ο κωδικός κατάστασης της απόκρισης είναι 301/302/307 (ανακατεύθυνση) γίνεται υποβολή νέας αίτησης στο URI που προσδιορίζεται από την κατάλληλη επικεφαλίδα (Location).

Web browsers: ροή επεξεργασίας αποκρίσεων (3)

- ▶ Η αίτηση προωθείται στη μονάδα διερμηνείας περιεχομένου (5) η οποία εξετάζει τις επικεφαλίδες κωδικοποίησης (π.χ. Content-Encoding) και τις επικεφαλίδες προσωρινής μνήμης (π.χ. Expires. Cache-Control).
- ▶ Αν υπάρχουν επικεφαλίδες caching καλείται η μονάδα προσωρινής μνήμης ώστε να δημιουργηθεί νέα εγγραφή ή να τροποποιηθεί η υπάρχουσα (π.χ. η χρονοσφραγίδα της)
- ▶ Βάσει των επικεφαλίδων κωδικοποίησης προσδιορίζεται ο τύπος MIME του σώματος της απόκρισης. Ανάλογα με τον τύπο γίνεται η επεξεργασία του περιεχομένου που υποστηρίζεται από τον browser (π.χ. text/html, text/css, image/jpeg, application/javascript,) ή διαφορετικά χρησιμοποιείται η κατάλληλη εξωτερική βοηθητική εφαρμογή ή το κατάλληλο plug-in.
- ▶ Αν το περιεχόμενο του σώματος αναφέρεται σε πόρους που βρίσκονται σε διαφορετικά URI καλείται η μονάδα δημιουργίας αιτήσεων (6) για την ανάκτησή τους.

Web browsers: ροή επεξεργασίας αποκρίσεων (4)

- ▶ Τέλος, η μονάδα διερμηνείας περιεχομένου στέλνει το αποτέλεσμα της επεξεργασίας της απόκρισης στη διεπαφή χρήστη η οποία το εμφανίζει στον χρήστη (7).

