



Tehnologii Internet

CURSUL 02 – ARHITECTURA WEB

Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare
Specializarea Tehnologia informației

© 2017-2018 Adrian ALEXANDRESCU



Cuprins

1. Arhitectura web
2. Protocolul HTTP
3. Implementarea unui server HTTP



1. Arhitectura web

1.1. Definiții

1.2. Identificarea resurselor web

1.3. Principii ale arhitecturii web

1.4. Tipuri media

1.5. Browser-e web



1.1. Arhitectura web - definiții

World Wide Web - WWW, W3, Web

- Sistem de documente hipertext interconectate care sunt accesate prin intermediul Internetului

Hipertext (en., hypertext)

- Text afișat pe un dispozitiv electronic care conține referințe către alte date ce pot fi accesate de utilizator

Hiperlink (en., hyperlink)

- Referință la date care pot fi accesate de utilizator (în contextul unui hipertext)



1.2. Identificarea resurselor web

Uniform Resource Identifier (URI)

- Șir de caractere folosit pentru a identifica o resursă
- O resursă poate reprezenta orice:
 - un document electronic,
 - o imagine,
 - un serviciu,
 - o colecție de alte resurse,
 - oameni și corporații,
 - concepte abstracte (operatorii unei ecuații matematice)



1.2. Identificarea resurselor web

Exemple de URI-uri

- `http://www.google.com`
- `ftp://192.168.0.100/ti/laborator`
- `ldap://[2001:db8::7]/c=GB?objectClass?one`
- `mailto:aalexandrescu@tuiasi.ro`
- `news:sci.math.www.servers.unix`
- `urn:isbn:9780307743657`
- `telnet://192.0.1.16:80/`



1.2. Identificarea resurselor web

Sintaxa unui URI

`http://www.ace.tuiasi.ro/index.php?page=678#about`

scheme : [*authority*] *path* [*?query*] [*#fragment*]

`urn:isbn:9780307743657`



1.2. Identificarea resurselor web

URI, URL și URN

- URI (Uniform Resource Identifier)
 - Identifică o resursă
- URL (Uniform Resource Locator)
 - Localizează o resursă prin descrierea modalității de accesare a acesteia
- URN (Uniform Resource Name)
 - Definește identitatea (numele) unei resurse
- $URI = URL \mid URN$



1.2. Identificarea resurselor web

Uniform Resource Locator (URL)

- Șiruri de caractere US-ASCII (litere, cifre, caractere speciale, caractere rezervate)

- Caractere speciale:

\$ - _ . + ! * ' () ,

- Caractere rezervate:

; / ? : @ = &

- Caractere unsafe ("nesigure"):

spațiu < > " # % { } | \ ^ ~ [] `



1.2. Identificarea resurselor web

Uniform Resource Locator (URL)

- Exemple de scheme:
 - http - HyperText Transfer Protocol
 - ftp - File Transfer Protocol
 - mailto - adresa de mail
 - telnet - protocol client-server orientat pe text
 - ldap - Lightweight Directory Access Protocol
 - file - locația unui fișier local



3.2. Identificarea resurselor web

Sintaxa unui URL

scheme://user:password@host:port/url-path

- HTTP

http://host[:port][/path][?query][#fragment]

http://www.ace.tuiasi.ro/index.php?page=678#about

- FTP

ftp://[user[:password]@]host[:port]/[url-path]

ftp://aalexandrescu@192.168.243.80/cursuri/ti



1.2. Identificarea resurselor web

Internationalized Resource Identifier (IRI)

- Șir de caractere folosit pentru a identifica o resursă
- Caracterele sunt din Universal Character Set (Unicode/ISO 10646)
- Sintaxa unui IRI este similară cu cea a unui URL
- Avantaj: o adresă poate fi afișată într-o anumită limbă



1.2. Identificarea resurselor web

Internationalized Domain Name (IDN)

- Numele unui domeniu care conține caractere non-ASCII din alfabetul unei limbi
- Astfel de nume de domenii sunt prefixate cu `xn--`
- Este folosită metoda Punycode de a converti un șir unicode la un șir de caractere dintr-un set mai restrictive (ASCII)
- Exemplu:

`http://www.müller.de/`

`http://www.xn--mller-kva.de/`



1.3. Principii ale arhitecturii web

- Arhitectura web are două niveluri:
 - Un client web (e.g., browser-ul) care utilizează / afișează informația
 - Un server web care transferă informația la client
- Tehnologiile utilizate:
 - URL / URI
 - HTML (HyperText Markup Language)
 - HTTP (HyperText Transfer Protocol)



1.3. Principii ale arhitecturii web

- Principii:
 - Un URI trebuie să identifice o singură resursă
 - Nu trebuie ca mai mult de un URI să identifice o anumită resursă
 - Schemele URI trebuie reutilizate (în loc de a se crea scheme noi) dacă oferă proprietățile necesare
- Un agent nu trebuie neapărat să întrerupă utilizatorul pentru a obține acceptul.



1.4. Tipuri media

- Specifică tipul unei resurse (ce conține o anumită resursă)
- Erau denumite Multipurpose Internet Mail Extensions (MIME)
- Sunt folosite ca valori pentru header-ul HTTP `Content-Type`
- Sintaxa: *top-level-type/subtype*
- Sunt case-insensitive



1.4. Tipuri media

- Text
 - text/[plain](#), text/[html](#), text/[css](#)
- Imagini
 - image/[gif](#), image/[jpeg](#), image/[png](#), image/[svg+xml](#)
- Audio
 - audio/[basic](#), audio/[mp4](#), audio/[mpeg](#)
- Video
 - video/[mp4](#), video/[mpeg](#), video/[x-matroska](#)
- Aplicații
 - application/[json](#), application/[javascript](#), application/[pdf](#), application/[octet-stream](#), application/[x-shockwave-flash](#), application/[xml](#), application/[zip](#)
- Alte tipuri: example, message, model, multipart



1.5. Browser-e web

- WorldWideWeb – primul web browser (1991)
- Internet Explorer (1995)
- Opera (1996)
- Safari (2003)
- Firefox (2004)
- Chrome (2008)
- Alte browser-e: Torch, Maxthon, SeaMonkey Avant Browser, Deepnet Explorer
- Alte browser-e (Linux): Konqueror, Epiphany, Qupzilla, Midori, Dillo, Arora; în mod text: ELinks, Lynx



2. Protocolul HTTP

2.1. Concepte

2.2. Istoric

2.3. Caracteristici

2.4. Comunicarea

2.5. Header-ul mesajului

2.6. Cererea HTTP

2.7. Răspunsul HTTP



2.1. Protocolul HTTP - concepte

HyperText Transfer Protocol (HTTP)

- "Este un protocol folosit în sistemele distribuite, colaborative și hipermedia"
- Specifică modul de comunicare pe web (World Wide Web)
- Este folosit pentru a transmite resurse (fișiere html, imagini, rezultatele unor interogări, șiruri de octeți)



2.2. Protocolul HTTP - istoric

Scurt istoric

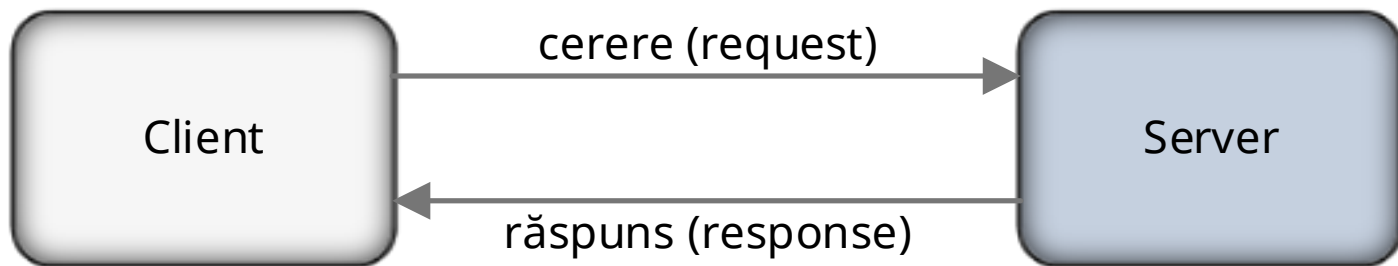
- În 1990 apare prima versiune, HTTP/0.9
- În 1996 apare o versiune semnificativ îmbunătățită, HTTP/1.0
- Se permite specificarea tipului datelor transmise și specificarea unor informații legate de cerere și răspuns
- În 1999 apare versiunea HTTP/1.1
- În 2015 apare versiunea curentă, HTTP/2



2.3. Protocolul HTTP - caracteristici

Caracteristici

- Paradigma client-server



- De obicei comunicarea are loc prin TCP/IP, iar portul la care este serverul este 80.
- Serverul trebuie să suporte mai mulți clienți



2.3. Protocolul HTTP - caracteristici

Caracteristici

- Protocolul este stateless ("fără stare")

Nivelul Aplicație	HTTP	Stateless
Nivelul Transport	TCP	Stateful
Nivelul Rețea	IP	Stateless
Nivelul Acces la rețea	BGP	Stateful



2.4. Protocolul HTTP - comunicarea

Pașii comunicării client-server

1. Utilizatorul introduce o adresă (URL) în clientul HTTP (e.g., browser-ul web)
2. Clientul inițializează o conexiune TCP pe portul 80 (de obicei)
3. Serverul HTTP acceptă conexiunea
4. Clientul trimite cererea
5. Serverul procesează cererea
6. Serverul trimite răspunsul
7. Conexiunea este închisă



2.4. Protocolul HTTP - comunicarea

Cerere – Răspuns (Request – Response)

- Un mesaj HTTP (cerere sau răspuns) trebuie să aibă următoarea formă:

```
linie_start ← Request-Line | Status-Line
*(header_mesaj CRLF) ← cheie : valoare
CRLF
[corp_mesaj]
```

- CR (Carriage Return) – 0x0D (13 decimal) – \r
- LF (Line Feed) – 0x0A (10 decimal) – \n
- SP (Space) – 0x20 (32 decimal)



2.5. Header-ul mesajului HTTP

Header-ul mesajului

- 4 tipuri: general, de entitate, specific cererii, specific răspunsului
- Header general (general-header)
 - Se referă la header-ele comune cererii și răspunsului
 - E.g., Cache-Control, Connection, Pragma, Trailer, Transfer-Encoding
- Header de entitate (entity-header)
 - Oferă informații cu privire la corpul mesajului
 - E.g., Allow, Content-Encoding, Content-Length, Content-Type, Last-Modified



2.6. Cererea HTTP

Cererea HTTP (HTTP Request)

<http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii>

GET /descopera-tuiasi/istoricul-universitatii HTTP/1.1

Host: www.tuiasi.ro

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:27.0)
Gecko/20100101 Firefox/27.0

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

Connection: keep-alive



2.6. Cererea HTTP

Cererea HTTP (HTTP Request)

1. Linia de start = Request-Line

2. Header-ele mesajului = *(Request-header CRLF)

CRLF

3. [Corpul mesajului = Message-body]



2.6. Cererea HTTP

1. *Linia de start = Request-Line*

method SP request-URI SP HTTP-version CRLF

- Method = acțiunea aplicată asupra URI-ului specificat (request-uri)
- Request-URI = * | absolutePath | absoluteURI | authority
- HTTP-Version = HTTP/1.1



2.6. Cererea HTTP

http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii

GET /descopera-tuiasi/istoricul-universitatii HTTP/1.1

Host: www.tuiasi.ro

method

request-URI

HTTP-version

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

Connection: keep-alive



2.6. Cererea HTTP

Metode

GET*	cererea unei anumite resurse
------	------------------------------

HEAD*	similar cu GET doar că serverul nu transmite și corpul mesajului
-------	--

POST	trimiterea unor date resursei identificate prin request-URI
------	---

PUT	trimiterea unor date care să fie stocate pe server
-----	--

* metode sigure (en, "safe") – care nu modifică starea serverului



2.6. Cererea HTTP

Metode (2)

DELETE	ștergerea unei anumite resurse
TRACE*	serverul va răspunde cu cererea trimisă de client (ecou)
OPTIONS*	serverul va răspunde cu informații despre opțiunile de comunicare disponibile
CONNECT	folosită la tuneluri
PATCH	aplică modificări parțiale asupra unei resurse



2.6. Cererea HTTP

2. *Header-ul mesajului= Request-header*

- Permite clientului să transmită informații suplimentare la server referitoare la cerere și la clientul care face cererea
- Conține mai multe perechi:

cheie: valoare

- Trebuie neapărat să conțină cheia: `Host` (în HTTP/1.0 nu era necesară)



2.6. Cererea HTTP

<http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii>

GET /descopera-tuiasi/istoricul-universitatii HTTP/1.1

Host: www.tuiasi.ro

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:27.0)
Gecko/20100101 Firefox/27.0

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

Connection: keep-alive



2.6. Cererea HTTP

Request-headers

Accept	Host	Proxy-Authorization
Accept-Charset	If-Match	Range
Accept-Encoding	If-Modified-Since	Referer
Accept-Language	If-None-Match	TE
Authorization	If-Range	User-Agent
Expect	If-Unmodified-Since	
From	Max-Forwards	



2.6. Cererea HTTP

3. *Corpul mesajului = Message-body*

- Șir de octeți sau caractere care este transmis de la client la server
- De obicei, dacă metoda folosită la transmiterea cererii este GET, atunci corpul mesajului este vid
- Există un corp al mesajului când utilizatorul transmite prin POST un formular completat sau când trimite o imagine la server



2.6. Cererea HTTP

Exemplu – formular de login

POST /students/confirmLogin.jsp HTTP/1.1

...

Content-Type: application/x-www-form-urlencoded

Content-Length: 35

Referer: https://aatuiasi.appspot.com/students/access.html

...

username=adrian&password=anaaremere

Username

adrian

Password

●●●●●●●●

Login



2.6. Cererea HTTP

<http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii>

GET /descopera-tuiasi/istoricul-universitatii HTTP/1.1

Host: www.tuiasi.ro

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:27.0)
Gecko/20100101 Firefox/27.0

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

Connection: keep-alive



2.7. Răspunsul HTTP

<http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii>

HTTP/1.1 200 OK

Date: Thu, 13 Feb 2014 10:35:58 GMT

Server: Apache/2.2.15 (CentOS)

Pragma: no-cache

Last-Modified: Thu, 13 Feb 2014 10:35:58 GMT

Connection: close

Transfer-Encoding: chunked

Content-Type: text/html; charset=UTF-8

[corpul-mesajului]



2.7. Răspunsul HTTP

Răspunsul HTTP (HTTP Response)

1. Linia de start = Status-Line

2. Header-ele mesajului = *(Request-header CRLF)

CRLF

3. [Corpul mesajului = Message-body]



2.7. Răspunsul HTTP

1. *Linia de start = Status-Line*

HTTP-Version SP Status-Code SP Reason CRLF

- HTTP-Version = HTTP/1.1
- Status-Code = număr de trei cifre care specifică modul în care a fost îndeplinită cererea
- Reason-Phrase = descriere scurtă a statusului



2.7. Răspunsul HTTP

<http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii>

HTTP/1.1 200 OK

Date: Thu, 13 Feb 2014 10:35:58 GMT

Server: status-code 15 (C reason-phrase

HTTP-version

Last-Modified: Thu, 13 Feb 2014 10:35:58 GMT

Connection: close

Transfer-Encoding: chunked

Content-Type: text/html; charset=UTF-8

[corpul-mesajului]



2.7. Răspunsul HTTP

Clase de coduri de stare (Status-Code)

1xx	Informațional	Cererea a fost primită și urmează a fi procesată (răspuns provizoriu)
2xx	Succes	Cererea a fost procesată cu succes
3xx	Redirectare	Clientul trebuie să ia acțiuni suplimentare pentru a îndeplini cererea
4xx	Eroare client	Sintaxa cererii este eronată sau cererea nu poate fi îndeplinită de server
5xx	Eroare server	Serverul nu poate îndeplini o cerere aparent validă



2.7. Răspunsul HTTP

Coduri de stare uzuale

200	OK	Cererea a fost procesată cu succes
302	Found	Resursa cerută se găsește temporar la un alt URI
304	Not Modified	Resursa cerută nu a fost modificată (se găsește în cache-ul browser-ului)
404	Not Found	Resursa nu a fost găsită la server
405	Method Not Allowed	Metoda (GET, POST, ...) specificată nu este permisă pentru accesarea resursei
500	Internal Server Error	La server a apărut o problemă în încercarea procesării cererii



2.7. Răspunsul HTTP

2. *Header-ul mesajului= Response-header*

- Permite serverului să transmită informații suplimentare la client referitoare la răspuns și la serverul care răspunde
- Conține mai multe perechi:

`cheie: valoare`



2.7. Răspunsul HTTP

<http://www.tuiasi.ro/descopera-tuiasi/istoricul-universitatii>

HTTP/1.1 200 OK

Date: Thu, 13 Feb 2014 10:35:58 GMT

Server: Apache/2.2.15 (CentOS)

Pragma: no-cache

Last-Modified: Thu, 13 Feb 2014 10:35:58 GMT

Connection: close

Transfer-Encoding: chunked

Content-Type: text/html; charset=UTF-8

[corpul-mesajului]



2.7. Răspunsul HTTP

Response-headers

Accept-Ranges

Age

ETag

Location

Proxy-Authenticate

Server

Vary

WWW-Authenticate



2.7. Răspunsul HTTP

3. Corpul mesajului = Message-body

- Șir de octeți sau caractere care este transmis de la server la client
- Corpul mesajului poate să lipsească
- Modalitatea de interpretare a corpului mesajului (text, imagine, ...) este dată de header-ul Content-Type



3. Implementarea unui server HTTP

1. Serverul așteaptă conexiuni pe un anumit port
2. Când s-a conectat un client, se creează un fir de execuție în care se realizează comunicarea cu respectivul client (pentru a permite conectarea și a altor clienți)
3. Serverul primește mesajul (HTTP) de la client
4. Cererea este procesată și este trimis răspunsul
5. Conexiunea cu clientul respectiv se închide



3. Implementarea unui server HTTP

Server HTTP care să permită vizualizarea conținutului unui director de pe harddisk

- De exemplu: <http://localhost:5678/Windows> va afișa conținutul directorului C:\Windows
- Prima liniei a cererii HTTP va arăta de forma:

GET /Windows HTTP/1.1

- Dacă este un director valid, atunci prima linie a răspunsului va fi:

HTTP/1.1 200 OK

- Iar dacă nu este un director valid:

HTTP/1.1 404 Not Found



3. Implementarea unui server HTTP

Server HTTP care să permită vizualizarea conținutului unui director de pe harddisk

- Liniile header din răspuns ar trebui să conțină măcar:
 - Content-Length
 - Content-Type
 - Content-Encoding (dacă răspunsul este arhivat)
 - Server
- Corpul mesajului de răspuns va conține numele fiecărui director și fișier din directorul pentru care se face cererea
- Înainte de corpul mesajului trebuie lăsată o linie goală (CRLF)



Bibliografie

- <http://tools.ietf.org/html/rfc1122>
- <http://www.networkworld.com/article/2228449/microsoft-subnet/ipv6-addressing--subnets--private-addresses.html>
- <https://tools.ietf.org/html/rfc1350>
- T. Berners-Lee. (2005) Uniform Resource Identifier (URI): Generic Syntax. [Online]. <http://tools.ietf.org/html/rfc3986>
- T. Berners-Lee. (1994) Uniform Resource Locators (URL). [Online]. <http://tools.ietf.org/html/rfc1738>
- M. Duerst. (2005) Internationalized Resource Identifiers (IRIs). [Online]. <http://tools.ietf.org/html/rfc3987>
- R. Fielding, et al. (1999) Hypertext Transfer Protocol – HTTP/1.1. [Online]. <http://tools.ietf.org/html/rfc2616>
- N. Freed. (1996) Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. [Online]. <http://tools.ietf.org/html/rfc2046>
- N. Freed. (2013) Media Type Specifications and Registration Procedures. [Online]. <http://tools.ietf.org/html/rfc6838>
- N. Freed, et al. (2014) Media Types. [Online]. <http://www.iana.org/assignments/media-types/media-types.xhtml>
- <http://www.w3.org/TR/2004/REC-webarch-20041215/>