

# Crearea unui Browser Web

Acest **laborator** prezinta **Crearea unui Browser Web**.

In acest PDF poti vizualiza cuprinsul si bibliografia (daca sunt disponibile) si aproximativ doua pagini din documentul original.

Arhiva completa de pe site contine un fisier, intr-un numar total de **6 pagini**.

Fisierele documentului original au urmatoarele extensii: doc.

## Cuprins

Sarcina lucrării  
Extras teoretic  
Protocolul HTTP. Descriere.  
Modul de funcționare.  
Mesajele de cerere.  
Metode.  
Coduri de stare.  
Securitate.  
Concluzie  
Anexe

## Extras

Sarcina lucrării: de realizat un browser web utilizând C#.

Extras teoretic:

Protocolul HTTP.Descriere și definiție.

HTTP (Hypertext Transfer Protocol) este un protocol de comunicare pentru transferul informației în internet care sunt păstrate pe servere World Wide Web. Este protocolul implicit al WWW, care conform modelului de referință OSI este de nivel aplicație. Realizarea și evoluția acestui protocol este coordonată de W3C (WWW Consortium).

Modul de funcționare: Când se scrie o adresă în browser aceasta este convertită într-o adresă IP (internet protocol) de către protocolul DNS (domain name system). După această operație are loc transferul de date utilizând protocolul TCP pe portul 80 standard al serverului HTTP, acest transfer se face ca răspuns al cererii HTTP-GET.

Mesajele de cerere: Un mesaj HTTP este divizat într-o parte de antet și o parte corp. Antetul cuprinde o serie de câmpuri (unele dintre ele obligatorii) oferind informații despre versiunea de protocol folosit, codificarea datelor, tipul de medii, lungimea și tipul mesajului etc.

Orice mesaj HTTP trebuie să debuteze cu un câmp indicând versiunea protocolului în prima linie a mesajului:

```
HTTP-Version ::= "HTTP-Version" ":" "HTTP" "/" digit "." digit
```

În prezent este operațional protocolul 1.1 deci toate mesajele de cerere și de raspuns vor începe cu linia HTTP/1.1.

Mesajele pot fi codificate conform autorității IANA (Internet Assigned Numbers Authority) fiind permise codificările:

\* gzip (GNU zip) este un cod Lempel-Ziv (LZ77) cu suma de control pe 32 de biți

\* compress este un cod produs de programul compress din toate mediile UNIX, după codificarea Lempel-Ziv-Welch (LZW)

Aceste codificări sunt specificate de câmpul Content-Transfer-Encoding.

Pentru MIME, se specifică tipul și subtipul mediului de informații (de exemplu: text/html, text/plain, image/jpeg, video/mpeg etc.) în câmpul Content-Type. Un mesaj poate fi transmis în format multipart, constând din mai multe entități, toate având o sintaxă comună. Dacă o aplicație recepționează un subtip nerecunoscut, în mod automat îl va trata ca multipart/mixed.

Simbolul "\*" specifică toate tipurile/subtipurile de medii dintr-o anumită categorie. De exemplu, pentru a accepta doar imagini, indiferent de format, se va transmite Accept: image/\*. Pot fi specificați unul sau mai mulți factori de calitate relativă. De pildă, cererea Accept: audio/\*; q=0.2, audio/basic este interpretată astfel: "se preferă tipul audio/basic dar serverul va trebui să trimită toate tipurile audio având calitatea de cel puțin 80%".

Metode:

HTTP permite 8 metode care indică operația dorită să fie aplicată sursei identificate.

GET este cea mai folosită metodă, fiind utilizată atunci când serverului i se cere o resursă. Reprezintă o cerere de accesare a unor informații (entități) identificate de Request-URI. Semantica metodei GET se schimbă în cerere condiționată dacă mesajul de cerere include câmpuri antet If-Modified-Since, If-Match, If-Range etc. Dacă se specifică un câmp Range, atunci GET va specifica o cerere parțială.

HEAD se comportă exact ca metoda GET, dar serverul returnează doar antetul resursei, ceea ce permite clientului să inspecteze antetul resursei, fără a fi nevoie să obțină și corpul resursei.

POST a fost proiectată pentru a trimite date de intrare către server. În cazul acestei metode datele se include în corpul cererii.

PUT este folosită pentru a depune documente pe server, fiind inversul metodei GET.

DELETE șterge sursa specificată.

TRACE: este o metodă folosită de obicei pentru diagnosticare, putând da mai multe informații despre traseul urmat de legătura HTTP, fiecare server proxy adăugându-și semnătura în antetul Via.

OPTIONS: este folosită pentru identificarea capacităților serverului Web, înainte de a face o cerere. Reprezintă o cerere de informații despre opțiunile de comunicare disponibile într-un dialog cerere/răspuns.

CONNECT: este o metodă folosită în general de serverele intermediare.

Coduri de stare: Pentru fiecare cerere a unui client, serverul HTTP răspunde cu o serie de coduri de stare a operației solicitate, dintre care menționez:

1. Coduri de informare (1xx) care dau informații despre o anumită acțiune:

\*100 Continue - clientul poate continua cererea, trebuind să trimită următoarea parte a unui mesaj parțial;

\*101 Switching Protocols - serverul înțelege cererea, dar necesită recepționarea unui câmp Upgrade pentru a ști ce tip de protocol va fi folosit la nivelul aplicației (e.g. pentru transmiterea de informații multimedia, când poate fi utilizat un protocol sincron, în timp-real);

## 2.Coduri de succes (2xx) raportează efectuarea cu succes a unui operațiuni:

\* 200 Ok - cererea a fost rezolvata cu succes;

Documentul complet de 6 pagini il poti citi daca il descarci din [Biblioteca.RegieLive.ro](http://Biblioteca.RegieLive.ro)

## Imagini din documentul complet:

	Cuprins
Seriele a două	
Extras teoretic	
Protocolul HTTP - Descriere	
Modul de funcționare	
Metode de cerere	
Metode	
Coduri de stare	
Securitate	
Concluzie	
Anexa	

**Securitate:** de mai jos sunt prezentate unele exemple de cereri și răspunsuri.

**Protocolul HTTP (transfer de date):**

HTTP (Hypertext Transfer Protocol) este un protocol de comunicație pentru transferul informației în internet care sunt preluate pe servere World Wide Web, care pot fi accesate prin intermediul browserului sau altor aplicații de client. Este unul din cele trei componente principale ale WWW, celelalte două fiind HTML și cascada de stiluri CSS.

Protocolul este definit în documentul RFC 2616 (Hypertext Transfer Protocol). Scopul este să definească un set de reguli care să permită comunicarea dintre un client și un server în cadrul unei sesiuni de lucru. Protocolul este definit în RFC 2616 (Hypertext Transfer Protocol).

Protocolul este definit în documentul RFC 2616 (Hypertext Transfer Protocol).

Protocolul este definit în documentul RFC 2616 (Hypertext Transfer Protocol).

HTTP este compoziția exactă a metodei GET, dar servirea este mai simplă decât în cazul metodei POST, care necesită verificarea și inspectarea parametrilor, fiind a fi accesat și obținut și conținutul.

HTTP este compoziția exactă a metodei GET, dar servirea este mai simplă decât în cazul metodei POST, care necesită verificarea și inspectarea parametrilor, fiind a fi accesat și obținut și conținutul.

HTTP este compoziția exactă a metodei GET, dar servirea este mai simplă decât în cazul metodei POST, care necesită verificarea și inspectarea parametrilor, fiind a fi accesat și obținut și conținutul.

Mai multe detalii se gasesc in [pagina documentului din Biblioteca.RegieLive.ro](http://pagina.documentului.din.Biblioteca.RegieLive.ro)