



ftninformatika

Java Web Development

Modul 2

Termin 3

Sadržaj

1. WWW
2. HTTP uvod
3. HTTP zahtev
4. HTTP odgovor

Dodatno:

1. URL
2. Port

Sadržaj www sajta

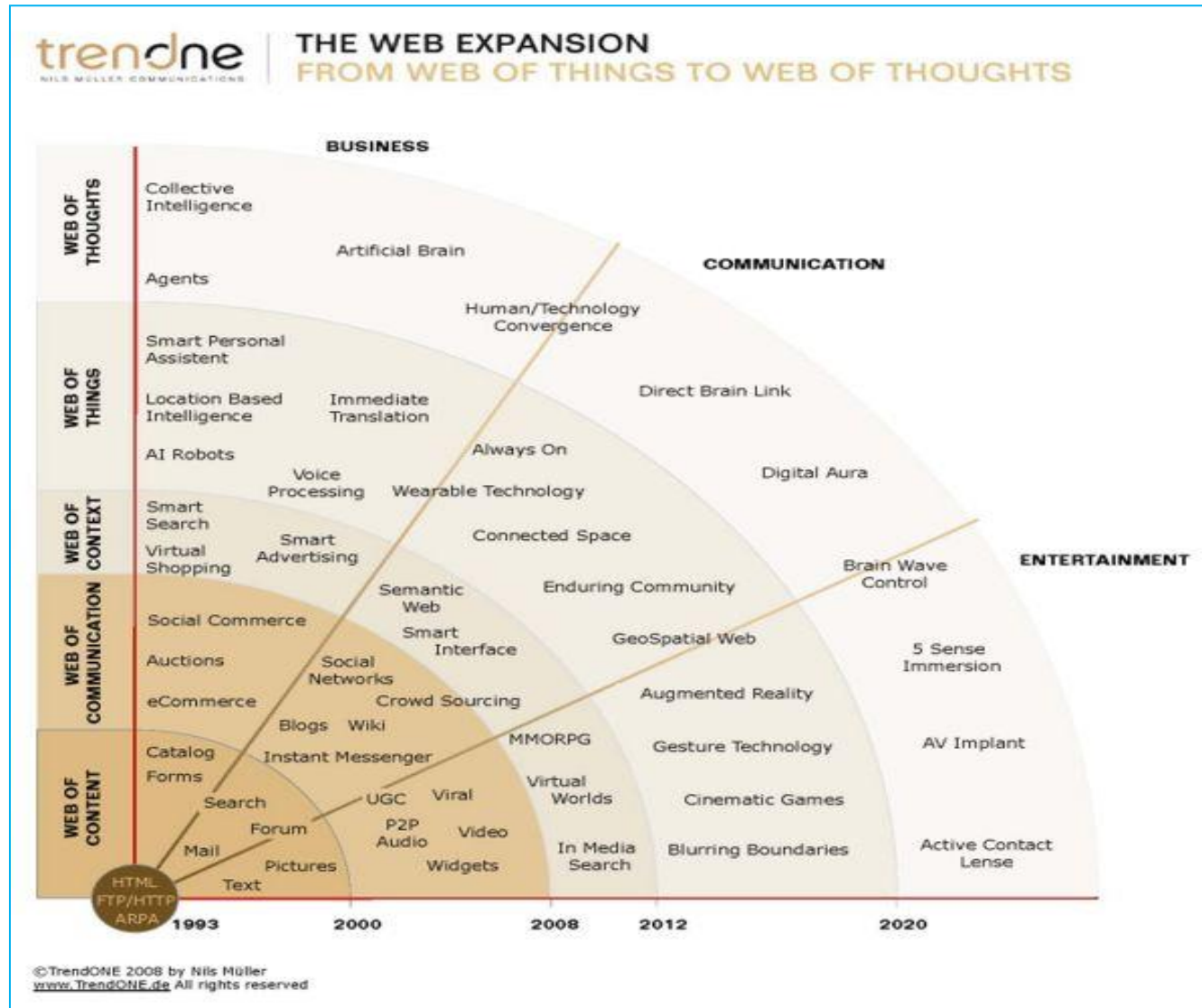
- HTML stranice
- multimedijalni elementi (slike, animacije, itd)
- drugi tipovi datoteka
- Prostor u kome se nalaze dokumenti kategorisani po *Uniform Resource Locator* (URL)
- www server i klijent komuniciraju preko HTTP protokola

World Wide Web

- Web 1.0 – Korisnici su puki “čitaoci” informacija (wikipedia). Statički sadržaj uglavnom. Server traži resurs u svom fajl sistemu
- Web 2.0 (1999) – Dinamički sadržaj. Koncept Web as a Platform. Nema više desktop aplikacija. Korisnici kreiraju sadržaj. Personalizovani korisnici (user account) WebShops, Social Networks. Rich User Experience, Cloud Computing, Software as a Service (SaaS).
- Web 3.0 (2006) - web of content where the meaning can be processed by machines
- Web 4.0, 5.0 – MobileOpen Linked, Intelligent, VR, AR

WWW

World Wide Web



HTTP uvod

Uvod

- HTTP je sraćenica od *HyperText Transfer Protocol*.
- HTTP je fundamentalni protokol na koji se bazira rad WWW

Protokol definiše :

- kako se formatiraju i prenose poruke WWW
- koje akcije bi trebalo da preduzmu veb serveri i pregledači kao odgovor na različite komande
- Verzije:
 - HTTP/1.0
 - HTTP/1.1 (permanent/persistent/**keep-alive** connection) -1997
 - HTTP/2.0 (performance improvements, header compression, usage of encryption, and prioritization of requests)- 2014

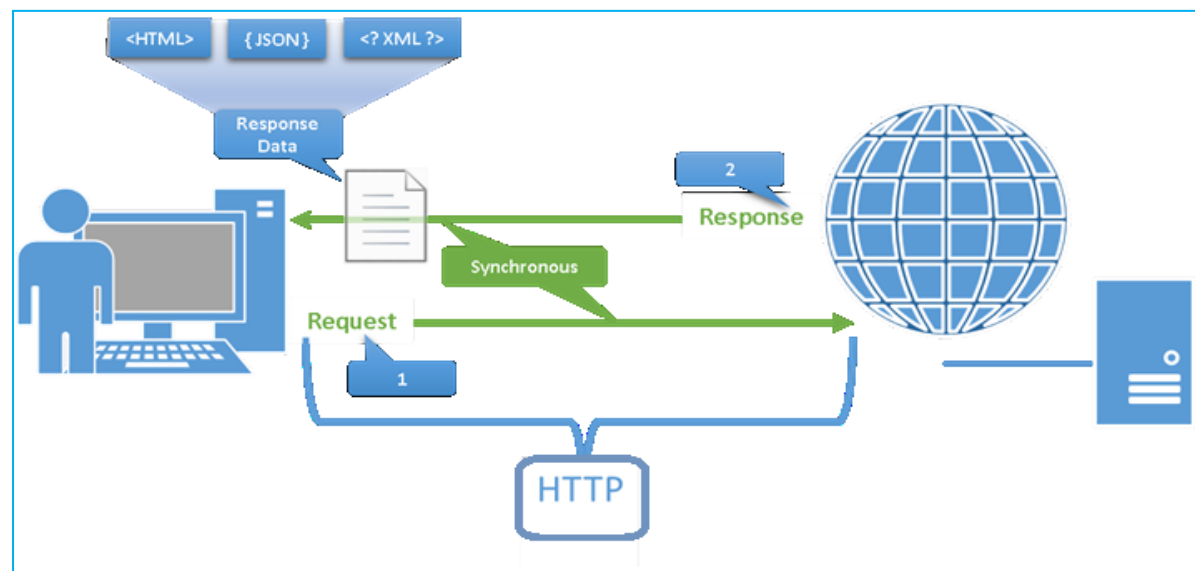
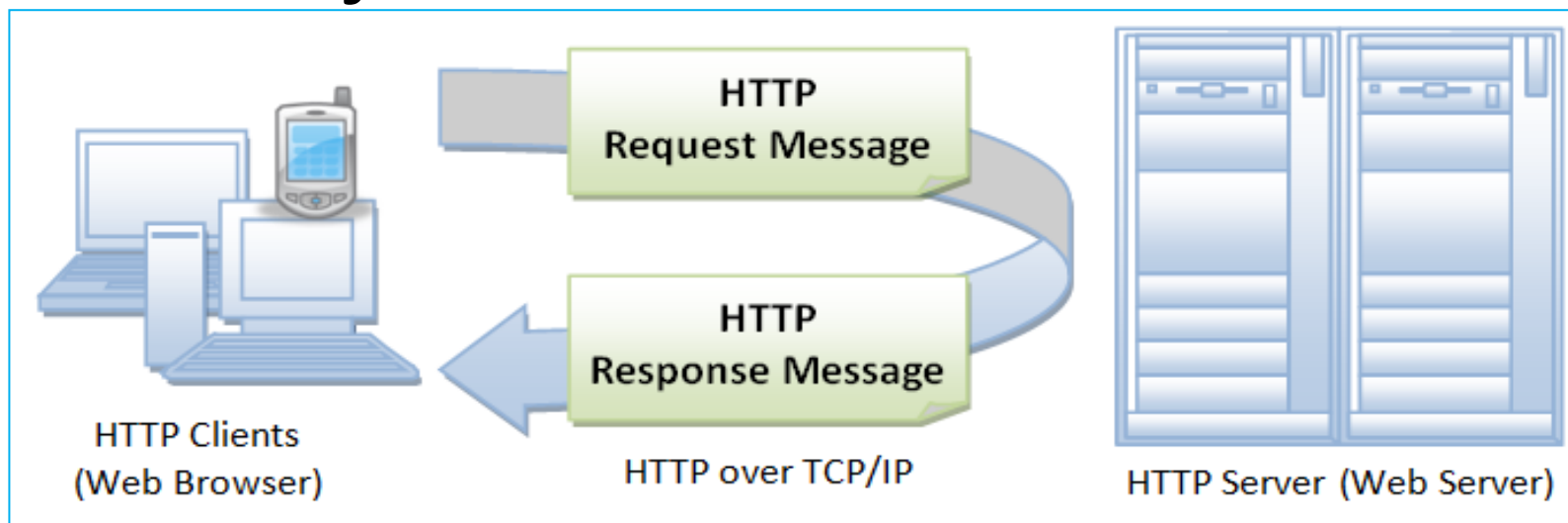
HTTP uvod

HTTP komunikacija

- zasnovana na zahtev/odgovor principu
- svaki par zahtev/odgovor se smatra nezavisnim od ostalih
- ne omogućava praćenje korisničke sesije, tj. niza zahteva upućenih od strane istog klijenta

HTTP uvod

HTTP komunikacija



HTTP uvod

Korisnička sesija

HTTP je stateless protokol koji ne zateva od servera čuvanje statusa klijenta ili korisničke sesije klijenta tj. rezultata obrade zahteva upućenih od strane istog klijenta.

HTTP serveri prevazilaze prethodno tako što implementiraju različite metode za održavanje i upravljanje sesijom, tipično se oslanjajući na jedinstveni identifikator *cookie* ili neki drugi parametar koji omogućava praćenje zahteva koji originiraju od istog klijenta (npr. URL Rewriting mehanizam), kreirajući stateful protokol iznad HTTP protokola.

HTTP uvod

HTTP verzije

- U verziji 1.0 po završetku isporuke odgovora klijentu konekcija se zatvara (za novu komunikaciju klijenta sa serverom opet treba da se uspostavi konekcija).
- U verziji 1.1 konekcija se ne zatvara tj. konekcija ostaje otvorena (keep-alive). Klijent će istu konekciju da koristi pri slanju novog zahteva ka serveru. Konekcija ostaje otvorena sve dok neko od stana u komunikaciji (klijent ili server) ne odluči da je neophodno da završi komunikaciju sa drugom stranom, što će uraditi tako što će zatvoriti konekciju.

HTTP uvod

HTTP verzije 1.1 prednosti i mane

Prednosti ver 1.1 :

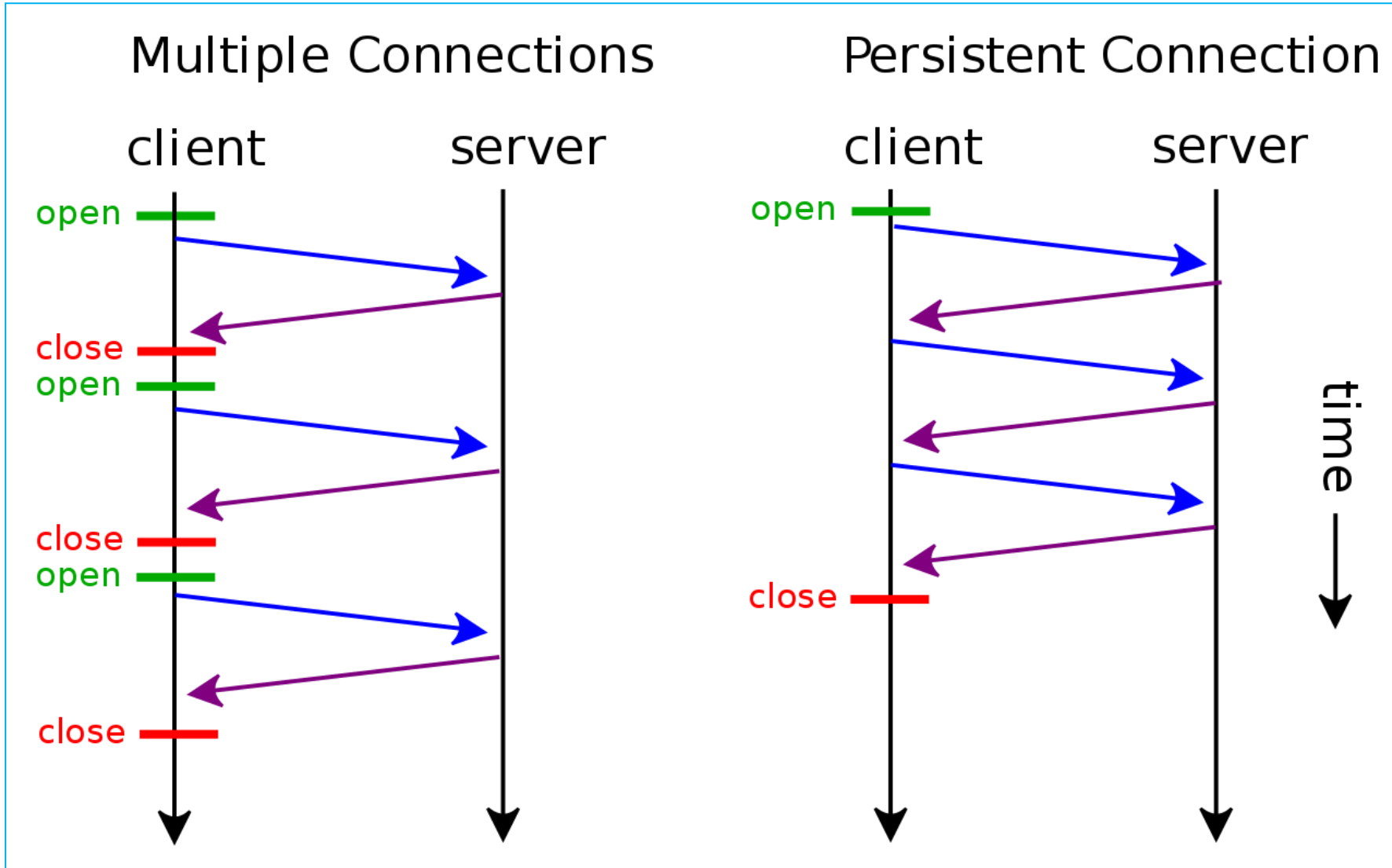
- Smanjeno zauzeće CPU jer je smanjen broj poruka koje se kreiraju, obrađuju i šalju mrežom.
- Smanjeno zagušene mreže (manje poruka za kreiranja TCP konekcija).

Mana ver 1.1 :

- Situacija u kojoj je klijent preuzeo sve podatke od servera ali nije zatvorio konekciju je problem. U takvoj situaciji server nepotrebno troši resurse za otvorenu vezu, umesto da te resurse mogu da koriste drugi klijenti.
- Prethodno može da utiče na dostupnost servera da prima nove zahteve klijenta, ako je na serveru ograničen broj klijenta koje istovremeno server opslužuje.
- Server će izvršiti zatvaranje konekcije koja je idle u zavisnosti od konfiguracije.

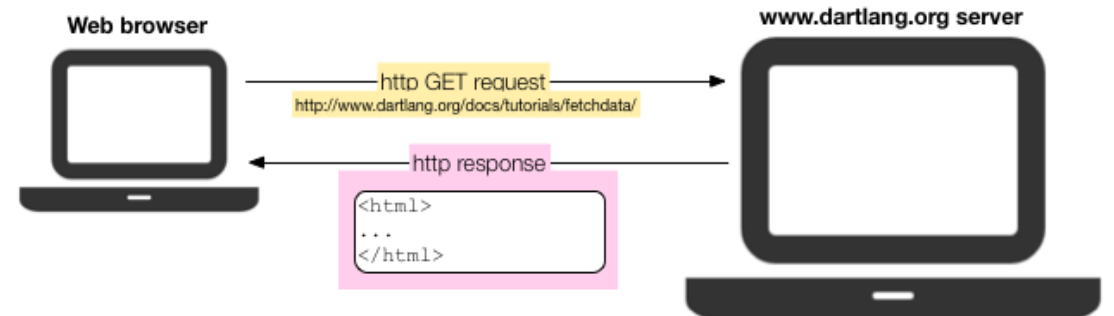
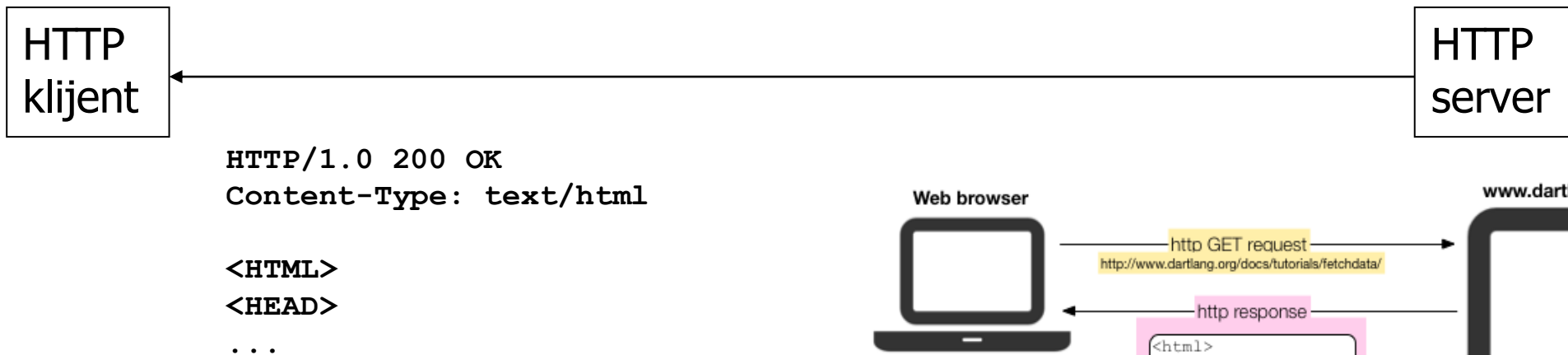
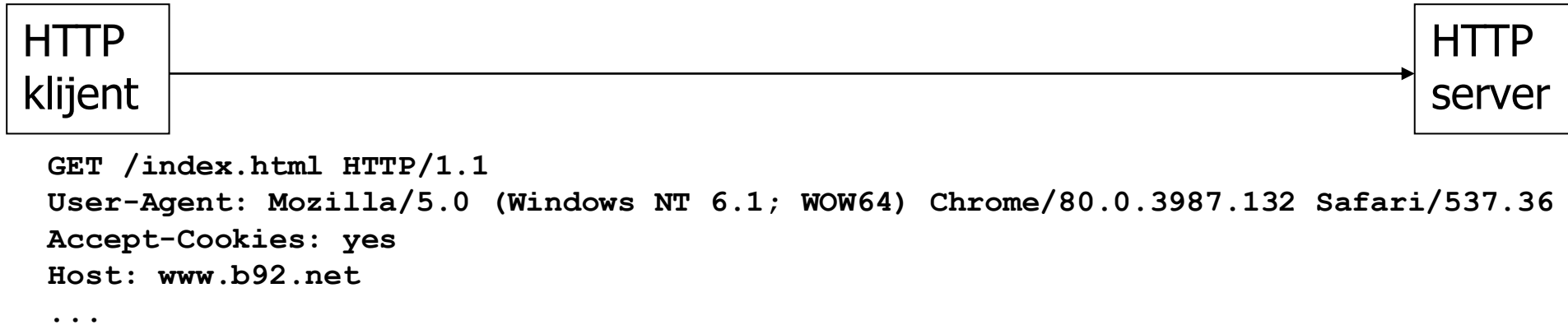
HTTP uvod

HTTP 1.0 i HTTP 1.1 komunikacija



HTTP uvod

HTTP razmena poruka



Chrome browser i Postman

<https://www.b92.net/index.html>

<https://www.youtube.com/watch?v=QV2iYFI5eSk>

HTTP zahtev

Struktura

- Počinje redom:
METHOD /putanja HTTP/verzija
- METHOD je:
 - GET,
 - POST, i dr.
- dodatni redovi sadrže attribute oblika:
Ime: vrednost
- prazan red na kraju
 - Ako je POST zahtev posle praznog reda idu parametri forme

HTTP zahtev

Metode

- GET – zahteva resurs od web servera
- POST – šalje parametre forme i traži odgovor
- HEAD – zahteva samo HTTP odgovor (response), bez slanja samog resursa
- PUT – omogućava klijentu da pošalje datoteku na web server
- OPTIONS – od web servera se traži spisak metoda koje podržava
- DELETE – omogućava klijentu da obriše resurs sa web servera

HTTP zahtev

Atributi u HTTP zahtevu

- User-Agent – identifikuje web browser

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.2) Gecko/20070219 Firefox/2.0.0.2

- Accept – definiše koje tipove resursa navigator prihvata kao odgovor na ovaj zahtev

Accept:

text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5

- Accept-Language – definiše koji jezike očekuje kao odgovor

Accept-Language: en-us,en;q=0.5

- Accept-Encoding – definiše koje kodiranje očekuje kao odgovor

Accept-Encoding: gzip,deflate

HTTP zahtev

Atributi u HTTP zahtevu

- Accept-Charset – definiše koju kodnu stranu očekuje

Accept-Charset: ISO-8859-1,utf-8;q=0.8,ASCII;q=0.7,*;q=0.6

- Cookie – definiše mehanizam praćenja sesije

Cookie: id1172566682241_1=1172566682241_1

- Referer – definiše URL sa kojeg se došlo na ovu stranicu
 - koristi se za statistiku
 - hotlinking

Referer: http://localhost/

- Connection – HTTP1.1 "kaže" serveru da ne zatvara konekciju po isporuci resursa

Connection: Keep-Alive

- q= broj definiše *qvalue*, a predstavlja **relative quality factor** odn. floating point vrednost "težine" parametra
- favorizovani Charset je ISO-8859-1 ili utf-8, ali ukoliko oni nisu podržani može i ASCII, a ako ništa od prethodnog nije podržano prihvaću i * (bilo koji drugi)

Primer HTTP zahteva

GET /index.html **HTTP/1.1**

User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)

Host: www.b92.net

Accept-Language: en-us

Accept-Encoding: gzip, deflate

Connection: Keep-Alive



Ako je vrednost **Close**,
konekcija se zatvara

HTTP zahtev

Keep Alive

HTTP persistent connection, also called HTTP keep-alive, or HTTP connection reuse, is the idea of using a single TCP connection to send and receive multiple HTTP requests/responses, as opposed to opening a new connection for every single request/response pair.

The newer HTTP/2 protocol uses the same idea and takes it further to allow multiple concurrent requests/responses to be multiplexed over a single connection.

HTTP odgovor

Struktura

- Počinje redom:
HTTP/verzija kod tekstualni_opis
- dodatni redovi sadrže attribute:
Ime: vrednost
- prazan red
- sledi sadržaj datoteke

HTTP odgovor

Kodovi odgovora

1xx Informational

100 Continue

101 Switching Protocols

102 Processing (WebDAV)

2xx Success

★ 200 OK ←

203 Non-Authoritative Information

206 Partial Content

226 IM Used

★ 201 Created ←

★ 204 No Content ←

207 Multi-Status (WebDAV)

202 Accepted ←

205 Reset Content

208 Already Reported (WebDAV)

3xx Redirection

300 Multiple Choices

303 See Other

306 (Unused)

301 Moved Permanently ←

★ 304 Not Modified ←

307 Temporary Redirect

302 Found ←

305 Use Proxy

308 Permanent Redirect (experimental)

4xx Client Error

★ 400 Bad Request ←

★ 403 Forbidden ←

406 Not Acceptable

★ 409 Conflict

412 Precondition Failed

415 Unsupported Media Type

418 I'm a teapot (RFC 2324)

423 Locked (WebDAV)

426 Upgrade Required

431 Request Header Fields Too Large

450 Blocked by Windows Parental Controls (Microsoft)

★ 401 Unauthorized ←

★ 404 Not Found ←

407 Proxy Authentication Required

410 Gone

413 Request Entity Too Large

416 Requested Range Not Satisfiable

420 Enhance Your Calm (Twitter)

424 Failed Dependency (WebDAV)

428 Precondition Required

444 No Response (Nginx)

451 Unavailable For Legal Reasons

402 Payment Required

405 Method Not Allowed

408 Request Timeout

411 Length Required

414 Request-URI Too Long

417 Expectation Failed

422 Unprocessable Entity (WebDAV)

425 Reserved for WebDAV

429 Too Many Requests

449 Retry With (Microsoft)

499 Client Closed Request (Nginx)

5xx Server Error

★ 500 Internal Server Error ←

503 Service Unavailable ←

506 Variant Also Negotiates (Experimental)

509 Bandwidth Limit Exceeded (Apache)

598 Network read timeout error

501 Not Implemented ←

504 Gateway Timeout

507 Insufficient Storage (WebDAV)

510 Not Extended

599 Network connect timeout error

502 Bad Gateway ←

505 HTTP Version Not Supported

508 Loop Detected (WebDAV)

511 Network Authentication Required

HTTP odgovor

Atributi u HTTP odgovoru

- Content-type – definiše tip odgovora

Content-Type: text/html

- Cache-Control – definiše kako se keš na klijentu ažurira
 - koristi se i Pragma: no-cache

Cache-Control: no-cache

- Location – definiše novu adresu kod redirekcije

Location: new.html

- Connection – potvrda klijentu da li da zatvori konekciju ili da je ostavi otvorenu

Connection: Keep-Alive

Primer HTTP odgovora

HTTP/1.0 200 OK

Date: Tue, 04 May 02004 08:55:09 GMT

Status: 200

Servlet-Engine: Tomcat Web Server/3.1 (JSP 1.1; Servlet 2.2; Java 1.4.2_02; Windows XP 5.1 x86; java.vendor=Sun Microsystems Inc.)

Content-Type: text/html

Last-Modified: Fri, 24 Oct 02003 16:07:24 GMT

Content-Length: 2524

Content-Language: en

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
```

```
<html>
```

```
<head>
```

```
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
  <meta name="GENERATOR" content="Mozilla/4.72 [en] (WinNT; U) [Netscape]">
```

```
  <meta name="Author" content="Anil K. Vijendran">
```

```
  <title>Tomcat v3.1</title>
```

```
</head>
```

```
<body></body>
```

```
</html>
```

Dodatno

URL

Uniform Resource Locator (URL)

- standard definisan u IETF RFC 1738 predstavlja podatke koji se mogu upotrebiti za dobavljanje resursa
- URL format - protokol:putanja-do-resursa

Protokol://Adresa_računara:port/Dinamički ili statički resurs

Primeri

<http://mojSajt.com:80/pictures> - Simbolička adresa i dinamički adresa
NAPOMENA 80 je podrazumevani port za http protokol

<http://192.168.0.1:4203/help.html> – Numerička adresa i statički resurs

Primer

The diagram illustrates the components of the URL `http://www.domain.com:1234/path/to/resource?a=b&x=y`. Red horizontal bars are placed under each component, with a vertical line connecting the bar to its label below or above the URL.

- protocol**: `http`
- host**: `www.domain.com`
- port**: `1234`
- resource path**: `/path/to/resource`
- query**: `?a=b&x=y`

Port

Port

- U računarskoj mreži *port* je softverski zadat kanal kojim komuniciraju aplikacije putem računarskih mreža.
- Predstavlja broj u opsegu 0-65535. Neki od ovih brojeva su predifinisani (0-1023) dok ostale portove mogu da koriste korisničke aplikacije.
- Portom razlikujemo aplikacije na računarima u domenu mrežne komunikacije.