

JavaScript 2

6. predavanje

HTTP request, XML, JSON

POWERED BY:



Client - server arhitektura

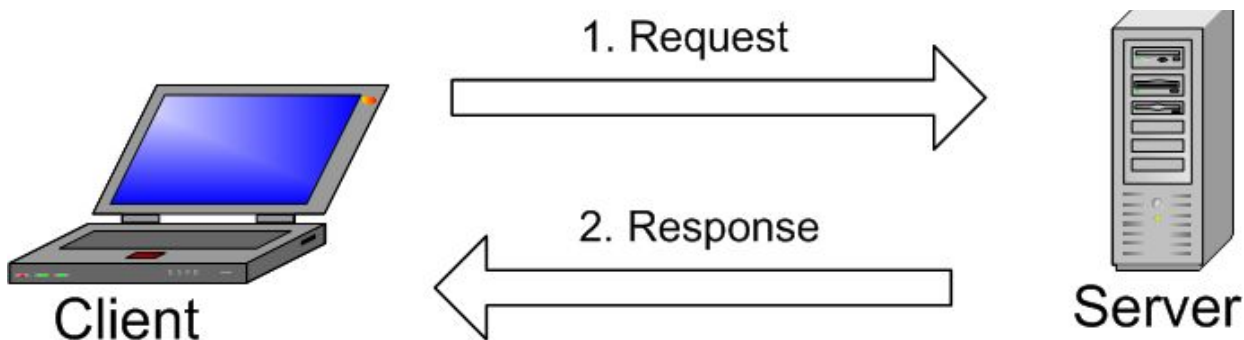
- označava razmjenu podataka između dvaju ili više računala koji su povezani u mrežu gdje klijenti zahtijevaju uslugu, a server im ju omogućuje
- server je namjensko računalo ili softver koje šalje i prima podatke od više klijenata. Postoji više tipova servera.

Client - server arhitektura

- Web server - najčešća primjena je da sadrži web stranice koje klijenti mogu dobiti na zahtjev
- klijent je oznaka za bilo koji računalni program koje se koristi uslugama servera
- Web browser - omogućuje pregled web stranica i multimedijalnih sadržaja vezanih uz njih

Client - server arhitektura

- klijent i server obično komuniciraju preko mreže, ali se mogu nalaziti i na istom računalu



DNS - Domain Name System

- Svako računalo koje je povezano na Internet mora imati jednoznačno dodijeljenu IP adresu
- IP adrese su nužne da bi se podaci upućeni s izvorišnog računala mogli preusmjeriti do odredišnog
- Ljudima je lakše zapamtiti ime od adrese pa je potreban sustav koji će ih povezati - **DNS**
- distribuirani hijerarhijski sustav servera u kojem se nalaze podaci o IP adresama i njihovim imenima
- Referenca na telefonski imenik
- Primjer: <http://spark.ba/school> -> 172.104.142.154

HTTP

- da bi računala komunicirala, moraju imati zajednički jezik i moraju slijediti pravila kako bi i klijent i server znali što očekivati
- jezik i pravila su definirani u komunikacijskom protokolu
- HTTP (engl. *HyperText Transfer Protocol*) je glavna i najčešća metoda prijenosa informacija na Web-u
- Kada upišemo URL u preglednik, šalje se HTTP zahtjev serveru koji dohvaća i vraća Web stranicu
- Konekcija i komunikacija sa serverom se odmah prekida nakon izvršenja zahtjeva klijenta
- GET/POST request

HTTP status

- 2xx - success
200 ok, 201 created...
- 3xx - redirect
301 trajno pomjeren...
- 4xx - greške na klijentu
403 zabranjeno, 404 - nije pronađeno...
- 5xx - greške na serveru
500 interna server greška...

AJAX

- Asynchronous JavaScript And XML
- Skup povezanih tehnika za web razvoj koje se koriste na klijentskoj strani
- Unatoč imenu, zahtjev ne mora biti asinkron i ne mora se koristiti XML
- Za dohvat podataka se koristi **XMLHttpRequest object**

XMLHttpRequest

```
var xmlhttp = new XMLHttpRequest();  
  
xmlhttp.open(Method, URL, Asynchronous, UserName,  
Password);  
xmlhttp.send();  
  
var xmlDoc = xmlhttp.responseXML; // XML response  
var html = xmlhttp.responseText ; // plain text
```

XMLHttpRequest properties

onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number (e.g. "404" for "Not Found" or "200" for "OK")
statusText	Returns the status-text (e.g. "Not Found" or "OK")

XMLHttpRequest methods

abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(method,url,async,uname,pswd)	<p>Specifies the type of request, the URL, if the request should be handled asynchronously or not, and other optional attributes of a request</p> <p>method: the type of request: GET or POST url: the location of the file on the server async: true (asynchronous) or false (synchronous)</p>
send(string)	<p>send(string) Sends the request off to the server.</p> <p>string: Only used for POST requests</p>
setRequestHeader()	Adds a label/value pair to the header to be sent

onreadystatechange

- **listener** koji se poziva na svaku promjenu *readyState* propertya XMLHttpRequest objekta pri asinkronom pozivu

```
var request = new XMLHttpRequest();  
request.onreadystatechange = function () {  
    if (this.readyState === 4){  
        alert(this.readyState);  
    }  
};  
request.open('GET', 'somepage.xml', true);  
request.send(null);
```

readyState

- 0 - UNSENT
- 1 - OPENED
- 2 - HEADERS_RECEIVED
- 3 - LOADING
- 4 - DONE

XML

- jezik koji definira pravila kako formatirati dokument da je jednako čitljiv ljudima i računalima
- sličan je HTML-u
- odvajanje podataka od prezentacije, razmjena podataka, pohrana podataka
- [primjer XML-a](#)

XML

- XML dokument je obična tekstualna datoteka čitljiva na svakoj platformi koja može čitati tekstualne podatke
- format je samodokumentirajući; oznake opisuju sadržaj koji se nalazi unutar njih
- ima stroga sintakсна pravila tako da je jednostavno kontrolirati ispravnost nastalog dokumenta
- računalni programi koji obrađuju dokument zbog toga mogu jednostavno obrađivati XML sadržaj

AJAX i XML

- same-origin policy - link za dohvat XML podataka mora biti istog podrijetla
- same-origin: same protocol, host and port
- [basic AJAX example](#)
- [txt file used in basic AJAX example](#)

Zadatak 1

→ Otvoriti [basic AJAX example](#).

Izdvojiti obradu dobivenog text-a u zasebnu funkciju *parseData()*.

Dodati callback parametar postojećoj *loadDoc()* funkciji te ga pozvati, ukoliko je request uspješno izvršen.

Proslijediti funkciju za obradu XML-a kao callback u *loadDoc()* funkciju.

[rješenje](#)

AJAX i XML

- ['Get CD' example](#)
- [XML file used in CD example](#)
- [formatted example](#)
- help with [DOM Object](#)

Zadatak 2

→ Otvoriti ['Get CD' example](#).

Dodati url i callback parametar postojećoj *loadDoc()* funkciji i pozvati callback, ukoliko je request uspješno izvršen.

Proslijediti url i funkciju za obradu XML-a kao callback u *loadDoc()* funkciju.

Dodati funkciju koja će izračunati i ispisati prosječnu cijenu CD-a.
[rješenje](#)

JSON

- JavaScript Object Notation
- služi za prijenos podataka između servera i aplikacije, neovisno o jeziku
- primjer JSON objekta
- dobiveni JSON je potrebno *parsirati*

```
var jsonObj =  
JSON.parse(request.responseText);
```

```
{"name": "brett", "country": "Australia"}
```

Zadatak 3

→ Ispisati tablicu kao u prošlom zadatku koristeći dobiveni JSON.

<https://output.jsbin.com/hugafagiwe/2.js>

[rješenje](#)

POWERED BY:



Ponavljjanje

POWERED BY:



THANK YOU
for attention