

NodeJS intro

NodeJS je javascript "runtime" nastao od strane Ryan-a Dahl-a, izgradjen u Google-ovom V8 endzinu.

Da objasnimo ove pojmove:

- runtime: okruzenje u kojem se nesto izvrsava. To znaci da NodeJS nije programski jezik, nego mehanizam koji omogucava izvrsavanje javascript-a koda (na masini, ne samo u browser-u),
- V8 endzin: je srz google chrome browser-a, razivjen sa ciljem da se postignu visoke performanse kod asinhronog izvrsavanja koda (ajax, inicijalno namijenjen prvenstveno za google maps).

NodeJS u sustini moze izvorsiti sve sto i ostali serverski/sistemske programski jezici:

- citanje/pisanje/brisanje fajlova,
- generisanje dinamickog sadrzaja,
- rad sa bazama podataka,
- rad sa formama,
- generisanje desktop aplikacija (ElectronJS)
- itd.

NodeJS je pisan da radi asinhrono, pri cemu je "single-threaded" i non-blocking.

Asinhrono znaci da se kod ne izvrsava sekvencijalno linija po linija, nego NodeJS server prihvata request i prosledjuje da u web api (koje je dalje zaduzen da izvrsavanje request-a i vracanje podataka kada je isti završen. Single-threaded znaci da postoji samo jedna instanca NodeJS-a koja prihvata request-ove (mada ovo uglavnom ne vazi za produkciju, gdje se najcesce kreira nekoliko NodeJS child instanci kako bi se pojacale performanse). Non-blocking znaci da se program ne blokira cekajuci na izvrsavanje nekog posla.

NodeJS ima ugradjeni http server. Primjer najjednostavnije aplikacije:

```
const http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end('<h1>Hello World!</h1>');
}).listen(8080);
```

U ovom primjeru koristimo takozvane module - konkretno http modul. Ovaj modul služi da kreiranje http servera - u NodeJS okruzenju nije neophodan eksterni alat (kao sto je apache kod php).

http je u našem slučaju objekat nad kojim možemo pozvati `createServer` funkciju. Ova funkcija prima funkciju kao parametar koja ima dva argumenta - `req` za request objekat i `res` za response objekat. Kao što možemo vidjeti, u ovom slučaju `req` nismo koristili u izvršenju, ali svakako bismo mogli, ukoliko bi nam npr. trebali query parameter ili neki drugi podatak iz request-a. `res` objekat se koristi za generisanje odgovora a servera. Konacno, na `createServer` metodu imamo "chain" metodu `listen`, koja zapravo podize server na portu koji je prosledjen toj funkciji. U ovu svrhu se vrlo često koristi 8080 ili 3000 ili 5000.

Da bismo mogli efikasno koristiti NodeJS, moramo dobro poznavati Javascript (ili još bolje - typescript, što je u sustini javascript sa tipovima).

Bitne NodeJS globalne varijable:

- `__dirname` - trenutni direktorijum
- `__filename` - trenutni fajl
- `process` - info o okruženju gdje se node app izvršava
- `require` - importovanje fajlova u commonjs stilu
- `import` - importovanje fajlova u module stilu

Bitni ugrađeni NodeJS moduli:

- `os` - module za rad sa OS-om
- `path` - module za rad sa putanjama do foldera/fajlova
- `fs` - modul za manipulaciju fajlovima
- `http` - modul za rad za http serverom

Konkretni primjer upotrebe node http servera za elementarnim routing-om:

```
import http from 'http';
```

```
http.createServer(function (req, res) {  
  if(req.url === '/json') {  
    res.writeHead(200, {'Content-Type': 'application/json'});  
    res.end('{"a": 1}');  
  } else if( req.url === '/') {  
    res.writeHead(200, {'Content-Type': 'text/html'});  
    res.end('Hello World!');  
  } else {  
    res.writeHead(404, {'Content-Type': 'text/html'});  
    res.end('<h1>Nothing found here!</h1>');  
  }  
})
```

```
}  
}).listen(3000);
```

Auto-restart NodeJS aplikacija

Za ovu namjenu mozemo koristiti paket koji se zove nodemon. Isti mozemo instalirati globalno ili kao dev dependancy. Primjer upotrebe kao dev dependency:

- kucamo npm i -D nodemon u konzoli (tamo gdje nam je package.json)
- u package.json treba dodati:
"scripts": {
 "start": "nodemon index.js" // ne mora nuzno biti index.js
}
- pokrenemo aplikaciju pomocu komande npm run start

Upotreba mysql2 biblioteke

Prvi korak je instalacija biblioteke:

npm i mysql2

Zatim import:

```
import * as mysql from 'mysql2';
```

Nakon toga potrebno je napraviti konekciju ka bazi:

```
const conn = mysql.createConnection({  
  host: 'localhost',  
  user: 'root',  
  database: 'fotodesk'  
});
```

Posle ovoga, mozemo pisati query:

```
http.createServer(function (req, res) {  
  if (req.url === '/users') {
```

```
    res.writeHead(200, {'Content-Type': 'application/json'});
    conn.query("SELECT * FROM user", function(err, results, fields) {
        res.end(JSON.stringify(results));
    })
} else {
    res.writeHead(404, {'Content-Type': 'text/html'});
    res.end('<h1>Nothing found here!</h1>');
}
}).listen(5000);
```