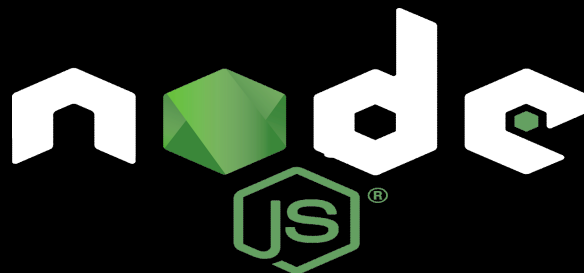


Node.js

JavaScript - advance

ITAcademy

<https://nodejs.org/en/>
Šta je node.js



- Node.js je JavaScript platforma za kreiranje konzolnih desktop aplikacija
- Node.js se najčešće koristi za kreiranje JavaScript serverske logike web aplikacije
- Node.js se naročito često koristi za realizaciju web socket i long pooling aplikacija ili njihovih delova

Upotreba Node.js

- Node.js je napisan većim delom u jeziku C++, ali je programerima endžin dostupan kroz JavaScript
- Najveći broj node.js funkcionalnosti je asinhron
- Dodatni moduli mogu se skidati pomoću **node package manager**-a (isporučuje se najčešće sa samim node.js-om)
- Da bi node program bio startovan, dovoljno je upotrebiti naredbu **node** u kombinaciji sa skriptom (js fajlom):

```
[# echo 'console.log("Hello!!!");' > test.js  
[# node test  
Hello!!!  
#
```

Node.js programiranje i moduli

- Node.js sintaksno koristi čist JavaScript, pa se programiranje u njemu svodi na poznavanje njegovih biblioteka (modula)
- Pored ugrađenih modula, u **Node.js-u**, često se koriste eksterni
- Eksterni moduli instaliraju se pomoću **NPM-a**:

```
# npm ls
/private/var/www/html/proba
└─ (empty)

# npm install socket.io
socket.io@1.4.6 node_modules/socket.io
├─ has-binary@0.1.7 (isarray@0.0.1)
├─ debug@2.2.0 (ms@0.7.1)
├─ socket.io-parser@2.2.6 (isarray@0.0.1, json3@3.3.2, component-emitter@1.1.2, benchmark@1.0.0)
├─ socket.io-adapter@0.4.0 (socket.io-parser@2.2.2)
├─ socket.io-client@1.4.6 (to-array@0.1.4, component-emitter@1.2.0, indexof@0.0.1, object-component@1.0.0)
└─ engine.io@1.6.9 (base64id@0.1.0, ws@1.0.1, engine.io-parser@1.2.4, accepts@1.1.4)

# npm ls
/private/var/www/html/proba
├─ socket.io@1.4.6
│   ├── debug@2.2.0
│   │   └─ ms@0.7.1
│   ├── engine.io@1.6.9
│   └─ accepts@1.1.4
```

Ucitavanje modula

- Sve dostupne module, moguće je dodati skripti metodom **require**.

```
var http = require("http");
```

- Ako modul nije instaliran preko NPM-a (ručno je dodat fajl), može se učitati navođenjem putanje do fajla modula

```
var http = require("http");  
var f = require("./mojmodul.js");
```

- Modul se kasnije koristi pomoću promenljive kojoj je dodeljena povratna vrednost metode require

HTTP modul

- Najkorišćeniji modul u Node.js-u je **http**
- Http se koristi za sve operacije sa soketima
- Pomoću ovog modula moguće je kreirati http server, ili komunicirati sa postojećim http serverom

Kreiranje standardnog http servera

```
var http = require("http");

var server = http.createServer(function(req, res){
    res.write("Hello man!!!");
    res.end();
});

server.listen(1400);
```

```
var http = require("http");
var url = require("url");
var server = http.createServer(function(req, res){
    var qs = url.parse(req.url).query;
    if(qs==null) return;
    var qsarr = qs.split("&");
    var a = parseInt(qsarr[0].split("=")[1]);
    var b = parseInt(qsarr[1].split("=")[1]);
    res.write((a+b).toString());
    res.end();
});

server.listen(1400);
```

Kreiranje i korišćenje websocket servera

- Node.js nema direktnu podršku za WebSocket, već je u tu svrhu potrebno instalirati neku od mnogobrojnih eksternih biblioteka

```
# npm install nodejs-websocket
nodejs-websocket@1.5.0 node_modules/nodejs-websocket
#
```

- Node.js nema direktnu podršku za WebSocket, već je u tu svrhu potrebno instalirati neku od mnogobrojnih eksternih biblioteka

```
var ws = require("nodejs-websocket");
var wsServer = ws.createServer(function(conn){
    conn.on("text",function(str){
        conn.send("Hey there!!!");
        console.log(str);
    });
    conn.on("close",function(code,reason){
        console.log("User left");
    });
}).listen(1300);
```

Server

```
<script>
var ws = new WebSocket("ws://localhost:1300");
ws.onopen = function(){ ws.send("Hello man!!!"); }
ws.onmessage = function(msg){ alert(msg.data); }
</script>
```

Klijent

Express framework

<http://expressjs.com/>

- Express je framework za kreiranje klasičnih http web aplikacija u JavaScript-u i Node.js-u

1

```
# npm install express  
express@4.13.4 node_modules/express
```

```
├── escape-html@1.0.3  
├── array-flatten@1.1.1  
├── utils-merge@1.0.0  
├── merge-descriptors@1.0.1  
├── cookie-signature@1.0.6  
├── vary@1.0.1  
├── content-type@1.0.1  
├── methods@1.1.2  
├── fresh@0.3.0  
├── range-parser@1.0.3  
├── parseurl@1.3.1  
├── content-disposition@0.5.1  
├── cookie@0.1.5  
├── serve-static@1.10.2  
├── path-to-regexp@0.1.7  
├── etag@1.7.0  
├── depd@1.1.0  
├── qs@4.0.0  
├── debug@2.2.0 (ms@0.7.1)  
├── on-finished@2.3.0 (ee-first@1.1.1)  
├── finalhandler@0.4.1 (unpipe@1.0.0)  
├── proxy-addr@1.0.10 (forwarded@0.1.0, ipaddr.js@1.0.5)  
├── send@0.13.1 (destroy@1.0.4, ms@0.7.1, statuses@1.2.1, mime@1.3.4, http-errors@1.3.1)  
├── type-is@1.6.12 (media-typer@0.3.0, mime-types@2.1.11)  
└── accepts@1.2.13 (negotiator@0.5.3, mime-types@2.1.11)
```

2

```
var express = require('express');  
var app = express();  
app.get("/", function(req, res){  
  res.send(new Date());  
});  
app.listen(1300, function(){  
  console.log("Server started...");  
});
```

3

```
# node test  
Server started...
```

Hostovanje statičkih fajlova

- Express može da hostuje statičke fajlove
- Za hostovanje statičkih fajlova u Express-u, koristi se metod aplikacije `use` i globalni metod **static**

```
var express = require('express');
var app = express();

app.use(express.static('resources'));
app.use("/static", express.static('resources'));

app.get("/", function(req, res){
    res.send(new Date());
});
app.listen(1300, function(){
    console.log("Server started...");
});
```



← → ↻ www.mysite.com/static/hello.txt

Hello from static



← → ↻ www.mysite.com/hello.txt

Hello from static

Generator aplikacije

Uz pomoć Node.js modula express generator, moguće je automatski strukturirati aplikaciju

```
# npm install express-generator -g
/usr/local/bin/express -> /usr/local/lib/node_modules/express-generator/bin/express
express-generator@4.13.1 /usr/local/lib/node_modules/express-generator
├── sorted-object@1.0.0
├── commander@2.7.1 (graceful-readlink@1.0.1)
├── mkdirp@0.5.1 (minimist@0.0.8)
└─
# express testapp

create : testapp
create : testapp/package.json
create : testapp/app.js
create : testapp/public
create : testapp/public/stylesheets
create : testapp/public/stylesheets/style.css
create : testapp/routes
create : testapp/routes/index.js
create : testapp/routes/users.js
create : testapp/public/javascripts
create : testapp/public/images
create : testapp/views
create : testapp/views/index.jade
create : testapp/views/layout.jade
create : testapp/views/error.jade
create : testapp/bin
create : testapp/bin/www

install dependencies:
$ cd testapp && npm install

run the app:
$ DEBUG=testapp:* npm start

#
```

`npm install express-generator -g`

`express testapp`