

Uvod u npm - menadžer paketa za Node

Napisao [Žarko Stanković](#), dana 25. jul 2012 00:57

Node.js se generalno sastoji od dve ključne grupe biblioteka:

1. Built-in (node-core) - one koje dolaze uz samu instalaciju node-a (razvija ih node.js tim)
2. Userland - one koje razvija node.js zajednica (bilo koji pojedinac ili grupa)

Razvojni tim node-a je odlučio da u sam `core` uključi samo one biblioteke koje su neophodne za veoma nizak (low-level) pristup hardveru i izradu osnovnih aplikacija/servera, a da sve ostalo prepusti zajednici (userland).

Šta sad ovo znači???

Kada instalirate `node` primetićete da i nema baš mnogo biblioteka (modula/fajlova). Zapravo sam `core` je veoma mali i kompletna [dokumentacija](#) se može pročitati za jedno popodne (ne morate to još uvek da radite, jer vam verovatno mnogo toga neće biti jasno :D).

Kao što sam gore naveo, osnovni node `core` će vam omogućiti npr pristup hardveru i kreiranje servera ali dalje od toga neće ići:

```
var http = require('http');
http.createServer(function (request, response) {
  response.end('Ovaj web server je kreirao Node.js');
}).listen(8888);
```

server.js

Postavlja se pitanje:

Šta sada da radim? Instalirao sam node ali nisam dobio ni biblioteke za obradu slike, ni drajvere za bazu podataka, ni gomilu drugih stvari koje sam očekivao da ću dobiti...

Zajednica (userland) kao spas!!!

Već ste shvatili da se uz instalaciju node-a ne dobija "od iple do lokomotive" i da će vam trebati još nešto sa strane... Rešenje je i više nego jednostavno:

Node.js zajednica svakodnevno objavljuje nove pakete (module) za node programe. U ovom trenutku ukupno ih ima: **39 538!**

Sve što treba da znate jeste da *node* ima svoj zvanični menadžer paketa zvani **npm** ili "*Node-ov menadžer paketa*".

npm je zvanični centralni repozitorijum node modula kao i softver za upravljanje bibilotekama (modula) od kojih vaša aplikacija zavisi (*dependencies*). Adresa web prezentacije je: <https://npmjs.org/>

Nekada davno (pre node verzije 0.6.3) *npm* se morao posebno instalirati mimo node instalacija. Međutim, postalo je evidentno da je npm jako bitan deo node-a te je core tim odlučio da ga uključi u podrazumevanu instalaciju node-a, drugim rečima npm je automatski instaliran sa instalacijom node-a. To možete i sami da proverite (naravno ako imate node >= 0.6.3 instaliran):

```
$ npm -v
1.3.8
```

Samo korišćenje npm-a može biti vrlo jednostavno ali i napredno ukoliko imate potrebe za time. U ovom članku biće objašnjena osnovna upotreba npm-a, a to je upravljanje zavisnih bibiloteka (dependencies) i njihova instalacija. Postupak je krajnje jednostavan i sve što je potrebno jeste da vaša aplikacija sadrži jedan manifest fajl (sa meta podacima) koji se zove **package.json**. Taj fajl sadrži sve relevantne informacije vezane za vašu aplikaciju uključujući i koje to biblioteke (module) vaša aplikacija zahteva. U sledećem primeru dat je najjednostavniji oblik ovog *package.json* fajla:

```
{
  "name": "moja-cool-aplikacija"
,  "version": "0.0.1"
,  "author": "Zarko Stankovic <stankovic.zarko@gmail.com>"
,  "description": "Ovo je opis moje aplikacije..."
,  "dependencies": {
    "express": "3.3.x"
  ,  "jade": "0.35.x"
  ,  "mongoose": "3.x.x"
```

package.json

```
}  
, "keywords": ["web", "blog", "sajt", "app"]  
}
```

Ovaj fajl, **package.json**, je potrebno da imate ukoliko planirate da javno objavite vašu biblioteku (modul) ili je potrebno za neki deployment sistem koji će automatski instalirati vaše dependencies. Preporučljivo je da vaša aplikacija ima ovaj fajl iako nije neophodan za rad. Da biste instalirali sve module neophodne za rad vaše aplikacije dovoljno je samo da se pozicionirate na root (osnovni) direktorijum vaše aplikacije i izvršite jednu komandu u terminalu:

```
$ npm install
```

ili čak skraćeno:

```
$ npm i
```

To je sve! Ova komanda će pročitati *package.json* i odraditi ceo posao za vas. Ukoliko nemate package.json onda morate sami ručno da instalirate jedan po jedan modul...

```
$ npm install express
```

```
$ npm install jade
```

```
$ npm install mongoose
```

Manifest fajl **package.json** može imati mnogo podataka i za više informacije možete pogledati [package.json cheatsheet](#) ili pročitati man stranicu samog **npm**-a:

1) <http://package.json.nodejitsu.com/>

2) `$ npm help json`

Gde se instaliraju moduli???

Podrazumevana lokacija za instaliranje modula je u root direktorijumu vašeg projekta u direktorijumu `./node_modules`. Ova lokacija je lokalna (vidljiva samo iz vašeg projekta) i preporučena u većini slučajeva. Korisnici upoznati sa sistemom **composer** će videti sličnosti odmah. Za ovaj direktorijum nema potrebe da vi kao korisnik bilo šta preuzimate - ceo posao završava `npm`.

Druga opcija je da instalirate modul globalno. To znači da će on biti vidljiv u celom sistemu (ili preciznije rečeno čak i izvan dosega vaše aplikacije). Za globalnu instalaciju modula dodajte `-g` parametar:

```
$ npm install -g nodemon
```

Sa ovom direktivom `npm` će instalirati `nodemon` modul globalno i biće vidljiv u celom sistemu.*
Ovo tačnije znači da će `npm` instalirati binarnu datoteku u putanju binarnih datoteka vidljivih u vašem sistemu. Na Ubuntu sistemu ovo je uglavnom `/usr/local/bin/` direktorijum. Ovo možete proveriti ako u terminalu izvršite: `$ echo $PATH`

Nakon uspešne globalne instalacije ovog node modula možete direktno izvršiti komandu `nodemon` sa komandne linije (pod uslovom da je u PATH-u - vidi gore):

```
$ nodemon  
8 Sep 23:13:08 - [nodemon] v0.7.2  
8 Sep 23:13:08 - [nodemon] watching: /home/zarko/dev/node/nodejs.rs  
8 Sep 23:13:08 - [nodemon] starting `node ./app.js`
```

Sa konzolne linije izvršite `npm` komandu bez ikakvih argumenata kako biste videli njen kratak opis:

```
$ npm
```

```
Usage: npm <command>
```

```
where <command> is one of:
```

```
add-user, adduser, apihelp, author, bin, bugs, c, cache,
completion, config, ddp, dedupe, deprecate, docs, edit,
explore, faq, find, find-dupes, get, help, help-search,
home, i, info, init, install, isntall, la, link, list, ll,
ln, login, ls, outdated, owner, pack, prefix, prune,
publish, r, rb, rebuild, remove, restart, rm, root,
run-script, s, se, search, set, show, shrinkwrap, star,
start, stop, submodule, tag, test, tst, un, uninstall,
unlink, unpublish, unstar, up, update, version, view,
whoami
```

```
npm <cmd> -h      quick help on <cmd>
npm -l            display full usage info
npm faq           commonly asked questions
npm help <term>   search for help on <term>
npm help npm      involved overview
```

```
Specify configs in the ini-formatted file:
```

```
/home/zarko/.npmrc
```

```
or on the command line via: npm <command> --key value
```

```
Config info can be viewed via: npm help config
```

```
npm@1.1.65 /usr/local/lib/node_modules/npm
```

Jedan kratak primer korišćenja ostalih komandi.

Ukoliko već imate instalirane *dependencies* module u vašoj aplikaciji ali želite da ih ažurirate, izvršite:

```
$ npm update # ažurira sve module vaše aplikacije iz fajla package.json
```

```
$ npm update mongoose # samo jedan specifični modul
```

Ukoliko imate problema sa korišćenjem *npm*-a u svakom trenutku možete dobiti brzu pomoć ako mu prosledite argument `-h`:

```
$ npm update -h
npm update [pkg]
```

ili čak detaljniju pomoć ako ispred komande navedete `help` ključnu reč:

```
$ npm help update
```

To bi bilo sve za brzo upoznavanje sa `npm`-om, menadžerom paketa za node. Za detaljnije informacije posetite zvaničnu prezentaciju <https://npmjs.org/> ili koristite pomoć sa konzolne linije.

Srećno i očekujemo uskoro vaše lične module objavljene u `npm` registru!

Preuzeto sa Node Srbija - <http://nodejs.rs>