

Node.js

NPM, hasznos modulok, telepítés és futtatás



Automatizálási és
Alkalmazott
Informatikai Tanszék

Node Package Manager (NPM)

- Node.js package kezelő rendszer, nem csak Node.js-hez
- Tartalmaz
 - > Alkalmazásokat
 - > Modulokat
- Ingyenes és open source, bárki feltöltheti a saját programját
- Node.js-el együtt települ
- Segítségével egyszerűen definiálhatók egy alkalmazás működéséhez szükséges függőségek
 - > Package.json fájl



Package.json

- Valid JSON-t tartalmazó állomány, amely meghatározza ...
- Tartalmazza:
 - > Alkalmazás nevét
 - > Verzióját
 - > Leírását
 - > Kulcsszavakat
 - > Szerzőt
 - > Alkalmazás belépési pontját (melyik .js fájl induljon el)
 - > Npm parancsok
 - > Függőségek, környezet függően

Semantic versioning (semver)

- Verziózási szabvány, 3 szegmensű formátum: x.y.z, pl: 1.0.0
 - > Patch release, 1.0.1: Csak hibajavításokat tartalmaz, teljes kompatibilitást az előző verzióval
 - > Minor release, 1.1.0 : Új szolgáltatások kerültek be, teljes visszafelé kompatibilitás az előző verzióval
 - > Major release, 2.0.0: Visszafelé kompatibilitást megtörhető változások történtek
- Függőségek verzióinak definiálása semver-el
 - > Patch releases: 1.0 vagy 1.0.x vagy ~1.0.4
 - > Minor releases: 1 vagy 1.x vagy ^1.0.4
 - > Major releases: * vagy x

Package.json

```
{
  "name": "char-server_demo",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git@...../ExpressJsChatServerDemo.git"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.13.3",
    "express": "^4.13.3",
  }
}
```

Package-ek telepítése és frissítése

- Globális telepítése
 - > Parancs: `npm <package-name> install -g`
 - > Az így telepített alkalmazások akár parancssorból is elérhetők, például:
 - gulp
 - bower
- Alkalmazás szintű telepítés
 - > Package.json-t tartalmazó mappában a következő paranccsal: `npm install`
- Frissítés
 - > A telepített modulok mindig csak a megengedett verzió szabályok szerint frissülnek
 - > Parancs: `npm update`

Népszerű és hasznos package-ek

- Express
 - > Web keretrendszer
- gulp, grunt
 - > Feladatok futtatása, webalkalmazások fejlesztésénél automatikus CSS preprocessor futtatás, állományok összefűzése, uglify és minify, stb.
- async
 - > Aszinkron műveletek szinkronizációja, Promise kezelés, stb.
- lodash, underscore.js
 - > Hasznos JavaScript függvények, kollekciók, objektumok hatékony manipulálására, stb.
- bower
 - > Package manager böngészők számára
- moment
 - > Dátum és idő kezelés

Népszerű és hasznos package-ek

- Socket.io
 - > Egyszerű socket létrehozás szerver és kliens oldalon
- Adatbázis kapcsolatok
 - > mongoskin
 - > mysql
 - > oracledb
 - > mssql
- ORM-ek
 - > mongoose
 - > orm (relációs adatbázisokhoz)

Telepítés éles környezetbe

- Telepítő készlet = forráskód – npm package-ek
- Alkalmazás elindítása
 - > npm start
 - > node .
- Probléma:
 - > Nem kezelt hiba esetén az alkalmazás le fog állni!
 - > Monitoring hiánya
- Szükség van egy „kiszolgáló” alkalmazásra

Forever

- Parancssoros eszköz script-ek folyamatos futtatásához
 - > Nem csak JavaScript!
 - > Amint nem fut az alkalmazás újraindítja azt
- Legegyszerűbb megoldás
- Több alkalmazás támogatása
- Egyedi konfigurációs állománnyal rendelkezik

```
forever start ./forever/development.json
```



```
npm install forever -g
15 dependencies   version 0.15.1
146 dependents    updated 2 months ago
342,232 downloads in the last month
download rank: top 1% of 183,000 packages
```

P(rocess) M(anager) 2

- Forever-nél fejlettebb folyamat kezelő Node.js/io.js alkalmazásokhoz
- Beépített load balancer
- Azonnali alkalmazás frissítés és újraindítás támogatása
- Naplózás és metrikák támogatása



```
pm2 start app.js
```

```
npm install pm2 -g
27 dependencies version 0.14.7
69 dependents updated a month ago
144,840 downloads in the last month
download rank: top 1% of 183,000 packages
```

```
[joni] ~/keymetrics/PM2 $ pm2 list
```

App name	id	mode	pid	status	restart	uptime	memory	watching
API	0	cluster	26076	online	0	2m	22.582 MB	disabled
API	1	cluster	26085	online	0	2m	22.527 MB	disabled
API	2	cluster	26274	online	1	2m	22.566 MB	disabled
API	3	cluster	26133	online	0	2m	22.563 MB	disabled
Worker	4	fork	0	stopped	0	0	0 B	enabled
Mailer	5	fork	26165	online	0	2m	15.125 MB	disabled
Front	7	fork	26865	online	0	12s	14.465 MB	enabled

PM2 monitoring :

● API

[0] [cluster_mode]

● API

[1] [cluster_mode]

● Worker

[2] [fork_mode]

● Mailer

[3] [fork_mode]

Strongloop Process Manager

- A StrongPM hasonlóan a PM2-höz egy folyamat kezelő rendszer Node.js alkalmazásokhoz
- Egyik legfejlettebb, legtöbb szolgáltatással bíró alkalmazás
 - > Néhány fejlett funkciója licence köteles!
- Legfontosabb funkciók:
 - > Build & deploy helyi vagy távoli rendszerre
 - > CPU és memória profiler
 - > Monitoring
 - > Multi host és cluster környezet támogatás
 - > Grafikus kezelőfelület
- Windows környezet nem támogatott
 - > De build & távoli deploy elérhető



Process Manager

StrongPM vs PM2 vs Forever

Feature	StrongPM	pm2	Forever
Run app locally	slc start	pm2 start app.js	forever start app.js
Restart on failure	Yes	Yes	Yes
Graceful/rolling restarts	Available	Available	No
OS startup script support	Yes	Yes	No
Security	HTTP auth and HTTP+SSH	SSH only for deploy	No
Log aggregation/rotation	Yes; log file and syslog	Yes; multihost, with rotation. Log file only; no syslog	No
Multiple app support	Manages multiple local or remote apps	Manages multiple local or remote apps	Manage multiple local apps
Language Support	Node only	Can run any script file	Can run any script file

Forrás: <http://strong-pm.io/compare/>

Feature	StrongPM	pm2	Forever
Build & Deploy			
Build and package repositories	Supports Git and npm-pack based builds	No	No
NEW! Deploy apps to Docker container	Yes	No	No
Remote deploy	Yes	Yes	No
Multiple deploys/revert	Deploys over SSH, HTTP. Revert any previous deployment.	Deploys over SSH, revert to previous deployment	No
Clustering & Management			
Clustering	Available	Available	No
Resize clusters	At deploy-time, start-time, and runtime	At deploy-time and start-time only	No
Manage remotely	Deploy over SSH+HTTP Manage over REST (+SSH)	Deploy over SSH	No
Load balancer auto-configuration	Yes	No	No
Profiling			
Profiling	Heap and CPU profiles	No	No
Profile triggering	Trigger profiling based on slow event loop	No	No
Metrics			
Metrics	CPU, memory, database, NoSQL connectors, many others	CPU, memory, stack traces reported on error output	No
Integrate with external metrics	Yes: StatsD, graphite, Splunk, etc.	keymetrics.io only	No

Socket.io

- Real time szerver és kliens kétirányú kommunikáció JavaScript alapokon
 - > Websocket és pulling támogatás
- Támogatott adat típusok
 - > JSON objektum
 - > Buffer objektum
 - Kép
 - Audio
 - Video
- Példa alkalmazás
 - > Chat
 - > Közös szerkeszthető dokumentumok
 - > Játékok, ...



socket.io

Swagger



- Nyelv független API leíró REST szolgáltatásokhoz
- <http://editor.swagger.io/>

The screenshot displays the Swagger UI interface. At the top, there is a green header bar with the Swagger logo, a red gear icon, and a text input field containing the URL `http://petstore.swagger.wordnik.com/api/api-docs.json`. To the right of the input field is a button labeled "special-key" and an "Explore" button. Below the header, the interface shows two sections of API endpoints. The first section is titled `/user` and includes a sub-header with links for "Show/Hide", "List Operations", "Expand Operations", and "Raw". It lists eight endpoints with their respective HTTP methods and descriptions:

- `POST /user.json/createWithArray`: Creates list of users with given input array
- `POST /user.json`: Create user
- `POST /user.json/createWithList`: Creates list of users with given list input
- `PUT /user.json/{username}`: Updated user
- `DELETE /user.json/{username}`: Delete user
- `GET /user.json/{username}`: Get user by user name
- `GET /user.json/login`: Logs user into the system
- `GET /user.json/logout`: Logs out current logged in user session

 The second section is titled `/pet` and also includes a sub-header with links for "Show/Hide", "List Operations", "Expand Operations", and "Raw". It lists four endpoints:

- `GET /pet.json/{petId}`: Find pet by ID
- `POST /pet.json`: Add a new pet to the store
- `PUT /pet.json`: Update an existing pet
- `GET /pet.json/findByStatus`: Finds Pets by status