
NPM Fundamentals

Introduction to the Node Package Manager

What is NPM?

- Platform for creating reusable code *packages* (aka *modules*)
 - Metadata (`package.json`)
 - Tooling (`npm`) installed with Node.js
 - NPM Website
 - Version management
 - It is huge!
 - Over 500k packages
 - Over 10 billion downloads per month
 - Recently acquired by GitHub
-

Installing a Package

- Open a shell in an *empty folder*
- Search for package *Chalk*: `npm search chalk`
 - Search online on npmjs.com
- Run: `npm install chalk`
- Create `app.js` using the *Chalk* package

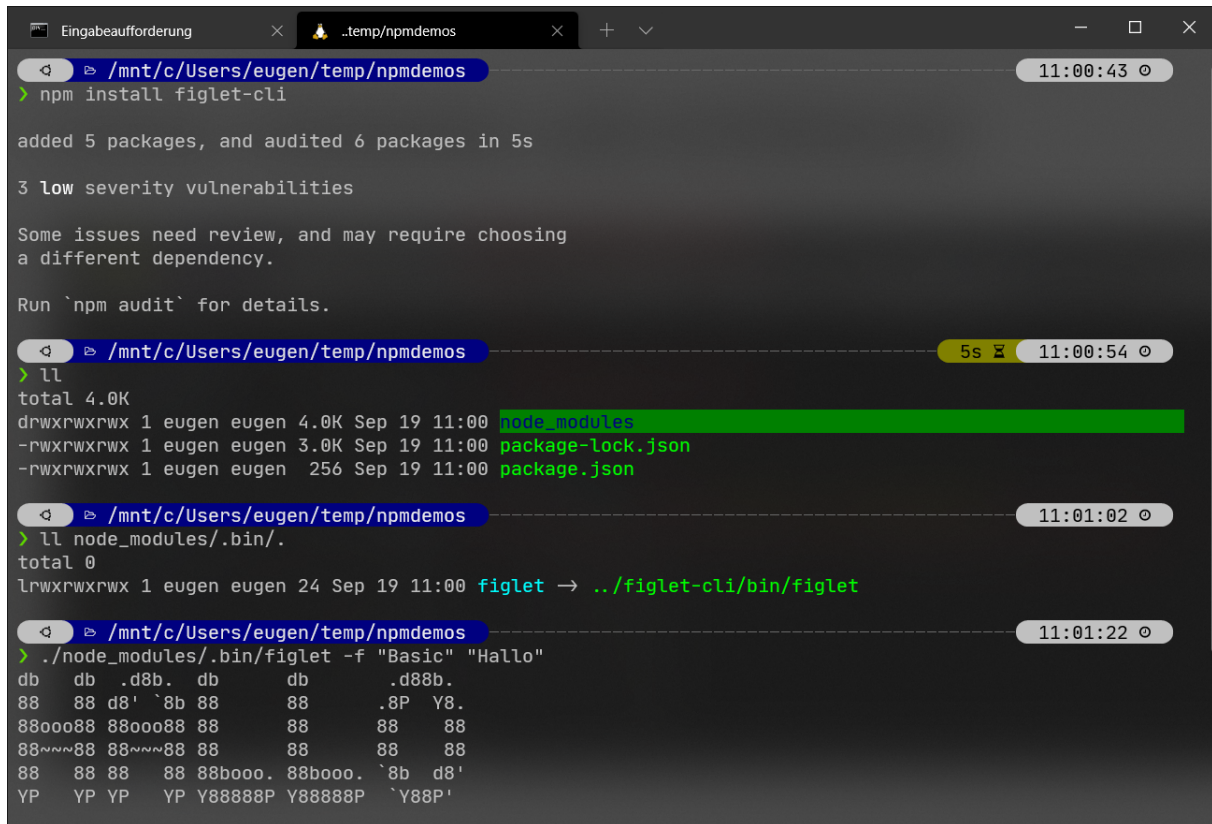
```
const chalk = require('chalk');  
console.log(chalk.red.bgBlue('Hello world!'));
```

- Run program: `node app.js`
-

***node_modules* Folder**

- Stores all locally installed packages
 - ...and their dependencies, and the dependencies of the dependencies...
 - Can contain large number of files
 - Don't copy it from machine to machine
 - Never check it in into Source Code Control
 - Executables
 - In *node_modules/.bin*
 - Use `npx` or scripts in *package.json* to run
 - Problems
 - How to find out which packages a program needs?
 - How to find out if new versions of packages are available?
 - How to reinstall necessary packages on a target machine?
 - Solution: *package.json*
-

Local Command Line Interfaces



```
Eingabeaufforderung  x  ..temp/npmdemos  x  +  v  -  □  x
> npm install figlet-cli
11:00:43

added 5 packages, and audited 6 packages in 5s

3 low severity vulnerabilities

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.

> ll
11:00:54
total 4.0K
drwxrwxrwx 1 eugen eugen 4.0K Sep 19 11:00 node_modules
-rwxrwxrwx 1 eugen eugen 3.0K Sep 19 11:00 package-lock.json
-rwxrwxrwx 1 eugen eugen 256 Sep 19 11:00 package.json

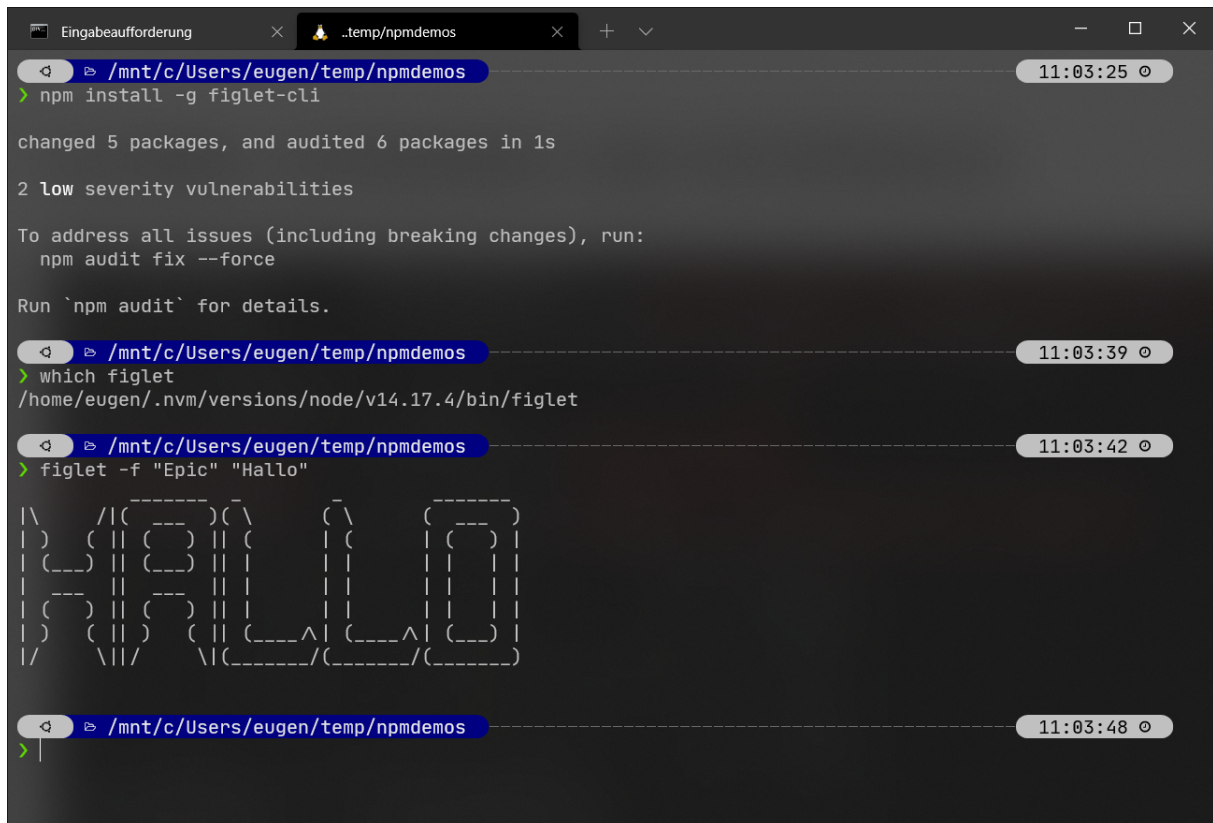
> ll node_modules/.bin/.
11:01:02
total 0
lrwxrwxrwx 1 eugen eugen 24 Sep 19 11:00 figlet -> ../figlet-cli/bin/figlet

> ./node_modules/.bin/figlet -f "Basic" "Hallo"
11:01:22
db db .d8b. db db .d88b.
88 88 d8' `8b 88 88 .8P Y8.
88ooo88 88ooo88 88 88 88 88
88~~~88 88~~~88 88 88 88 88
88 88 88 88 88booo. 88booo. `8b d8'
YP YP YP YP Y88888P Y88888P `Y88P'
```

Figure 1: NPM local install

More about FIGlet...

Global Command Line Interfaces



The image shows a terminal window with the following commands and output:

```
Eingabeaufforderung x ..temp/npmdemos x + v
> npm install -g figlet-cli
11:03:25

changed 5 packages, and audited 6 packages in 1s

2 low severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.

> which figlet
11:03:39
/home/eugen/.nvm/versions/node/v14.17.4/bin/figlet

> figlet -f "Epic" "Hallo"
11:03:42

  \  / |  (  _  )  (  \  (  \  (  _  )
 | )  (  |  (  )  |  (  |  (  |  (  )
 | (  )  |  (  )  |  (  |  (  |  (  )
 | )  (  |  (  )  |  (  |  (  |  (  )
 | )  (  |  (  )  |  (  |  (  |  (  )
 | /  \  /  \  /  \  /  \  /  \  /  \
  \ /  \ /  \ /  \ /  \ /  \ /  \ /

>
11:03:48
```

Figure 2: NPM global install

Creating *package.json*

- Open a shell in an *empty folder*
- Run: `npm init`
- Answer questions (hit *enter* to accept default)
- Run: `npm install chalk`
- Take a look at the generated `package.json` file, especially note `dependencies` object

```
{
  "name": "npmdemo",
  "version": "1.0.0",
  "description": "...",
  ...
}
```

```
"dependencies": {  
  "chalk": "^2.0.1"  
}
```

Version Numbers in *package.json*

- 3.5.0 means: Take currently latest version of the package
 - It does *not* mean: Take version 3.5.0
 - ^3.5.0 means: Take most recent major version (i.e. 3.x.x)
 - ~3.5.0 means: Take most recent minor version (i.e. 3.5.x)
 - Related: Semantic Versioning standard
 - More about semantic versioning in NPM
 - npm semver calculator
 - Install a specific version of a package with e.g. `npm install chalk@1.0.0`
-

Updating

- Run `npm update` to update dependencies (respecting semver)
 - Run `npm outdated` to check for outdated dependencies
 - Run `npm install...`
 - ...with a specific version number (e.g. `npm install sax@0.1.1`)
 - ...with a specific tag (e.g. `npm install sax@latest`)
 - Use `npm-check-updates`: `npx npm-check-updates -u`
 - For Angular:
 - Do not update Angular-related packages just for the fun of it
 - Follow guidelines on <https://update.angular.io/>
 - Use Angular CLI's `ng update`
-

***package.json* Content**

- Package metadata (e.g. author, description, etc.)
 - Version number
 - Tip: Follow Semantic Versioning standard
 - Dependencies
 - Necessary at runtime (`dependencies`)
 - Necessary at compile time (`devDependencies`, `--save-dev` or `-D` option, will later be important for *TypeScript*)
 - Collection of scripts (*npm scripts*)
 - Run at certain events (e.g. after install)
 - Run with `npm run script-name`
 - Note that `node_modules/.bin` is automatically added to the path
 - Exercise: Inspect *package.json* from *jQuery*
 - Read more details about *package.json*
-

Further Readings and Exercises

- Want to know more? Read/watch...
 - NPM docs