



O MNIE

PIOTR TARASIUK

mail: piotr.tarasiuk@softwarehut.com

github: <https://github.com/piotar>

linkedin: <https://linkedin.com/in/piotr-tarasiuk>

Zaliczenie (?)

Czym jest NodeJS?

NodeJS

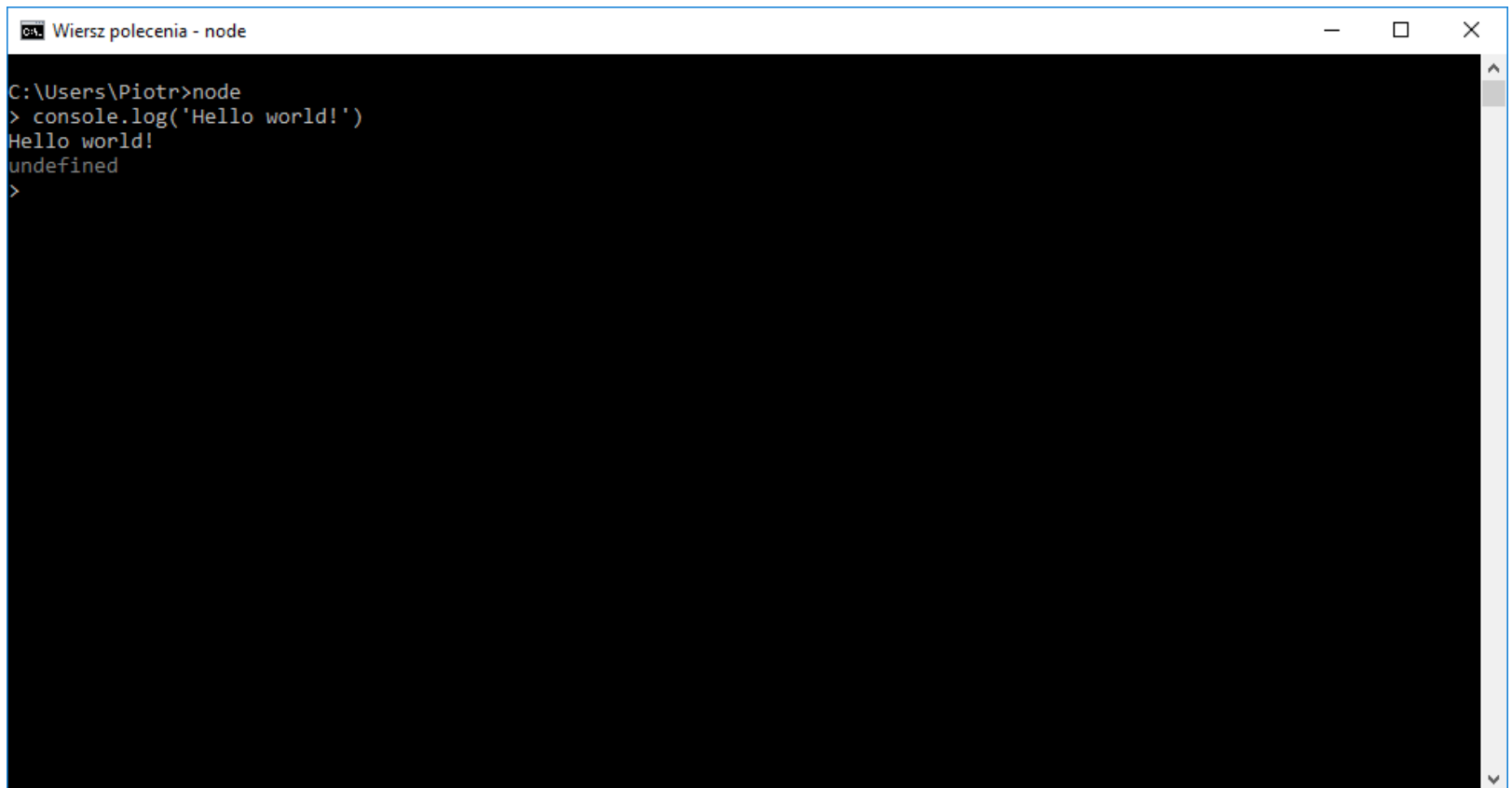
- Jest środowiskiem uruchomieniowym zaprojektowanym do tworzenia wysoce skalowalnych aplikacji, napisanych w języku **JavaScript** poza kontekstem przeglądarki.
- **NodeJS** jak również interpreter **JavaScript** w naszej przeglądarce bazują na tym samym silniku zwanym **V8 JavaScript Runtime Engine**.

V8 JAVASCRIPT RUNTIME ENGINE

- otwarty kod źródłowy
- napisany w języku c++
- możliwość rozszerzenia silnika przez napisane moduły w c++
- kompiluje kod JavaScript do **kodu maszynowego**

CZYM JEST KOD MASZYNOWY?

- Jest to zestaw rozkazów procesora w którym zapis programu wyrażony jest w postaci liczb binarnych stanowiących rozkazy oraz ich argumenty
- Kod maszynowy to postać programu komputerowego (zwana postacią wykonywalną lub binarną) przeznaczona do bezpośredniego lub prawie bezpośredniego wykonania przez procesor.



```
C:\Users\Piotr>node
> console.log('Hello world!')
Hello world!
undefined
>
```

The image shows a Windows command prompt window with the title "Wiersz polecenia - node". The window has a standard Windows title bar with minimize, maximize, and close buttons. The command prompt shows the following sequence of commands and output:

- The user enters the command `node` at the `C:\Users\Piotr>` prompt.
- The user enters the command `> console.log('Hello world!')`.
- The output `Hello world!` is displayed.
- The output `undefined` is displayed.
- The prompt `>` is shown again, indicating the command has finished execution.

NA CO POZWALA NAM NODEJS?

- NodeJS pozwala nam na manipulowanie systemem plików, stworzenie własnego serwera, łączenie się z bazami danych, obsługą wejścia/wyjścia, itp.
- JavaScript po stronie przeglądarki daje jedynie możliwość manipulowania danymi w naszym oknie(window).

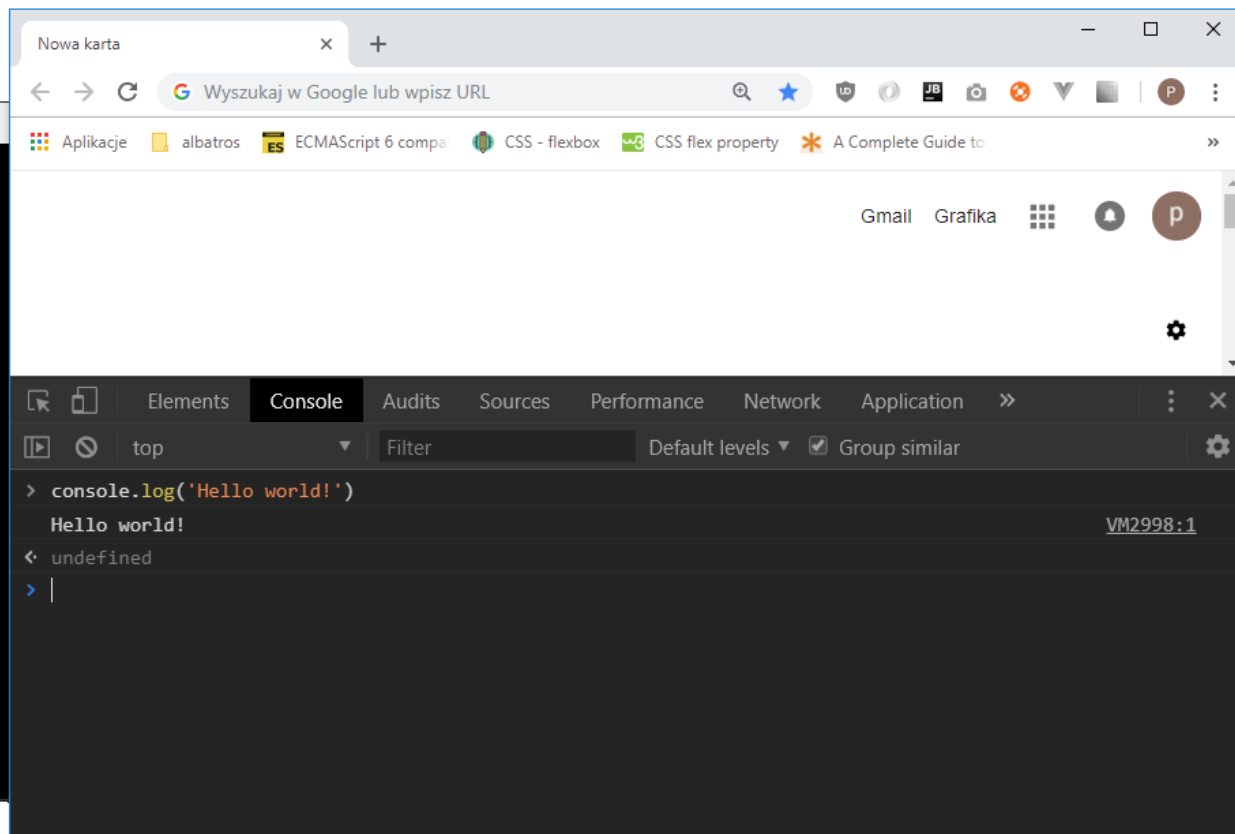
NODEJS VS JS W PRZEGLĄDARCE

global < = > window

process < = > document

```
Wiersz polecenia - node

C:\Users\Piotr>node
> console.log('Hello world!')
Hello world!
undefined
>
```



DLACZEGO NODEJS JEST TAK POPULARNY?

- Asynchroniczne wejście/wyjście ([non-blocking I/O model](#))
- NodeJS wykorzystuje programowanie zdarzeniowe ([event driven](#))
- NodeJS package ecosystem ([npm](#))

CZYM JEST WEJŚCIE/WYJŚCIE(I/O)?

- Jest czymś co każda aplikacja wykonuje cały czas, np. operacje na bazie danych, wysyłanie i odbieranie żądań HTTP, operację na systemie plików.
- Trzeba pamiętać, operacje I/O wymagają czasu!

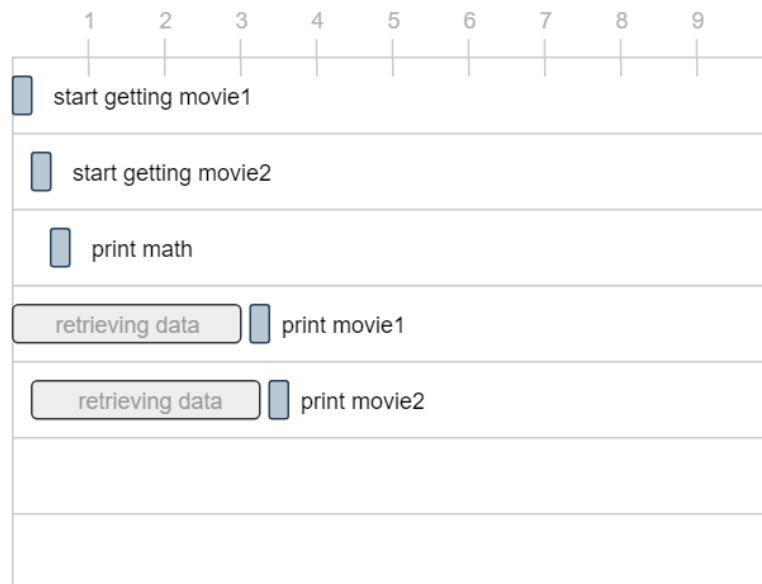
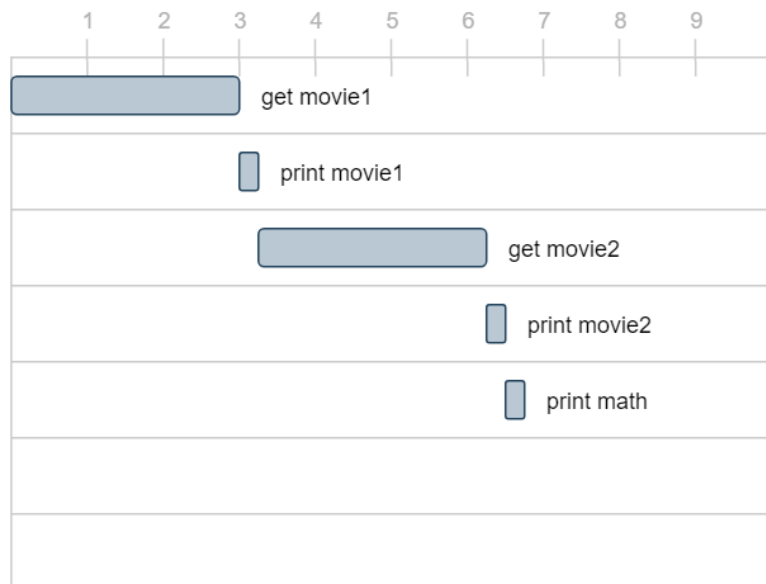
SYNCHRONICZNE
VS
ASYNCHRONICZNE



```
1 const getMovie = require('./sync.service');
2
3 const movie1 = getMovie('Matrix');
4 console.log(movie1);
5
6 const movie2 = getMovie('007');
7 console.log(movie2);
8
9 console.log('2 + 5 = ', 2 + 5);
```



```
1 const getMovie = require('./async.service')
2
3 getMovie('Matrix', movie1 => {
4   console.log(movie1);
5 });
6
7 getMovie('007', movie2 => {
8   console.log(movie2);
9 });
10
11 console.log('2 + 5 = ', 2 + 5);
```



CZY JEST NPMJS?

- Node package manager, domyślny manager pakietów dla środowiska NodeJS.
- Npm jest aplikacją wiersza poleceń, za pomocą której można instalować aplikacje dostępne w repozytorium Npm.
- Ale nie tylko...

PODSTAWY NODEJS

REQUIRE

Funkcja wbudowana w NodeJS, pozwala nam na wczytanie zewnętrznego pliku/modułu.

```
1 // własny moduł
2 const user = require('./user'); // ścieżka relatywna
3
4 // wbudowany moduł
5 const fs = require('fs'); // ścieżka absolutna
6
7 // zewnętrzny moduł
8 const _ = require('lodash'); // ścieżka absolutna
```

module.exports

Pozwala na wyeksportowanie zmiennych/funkcji.

```
1 module.exports = 'Ala ma kota';
2 // lub
3 module.exports.add = (a, b) => a + b;
4 module.exports.sub = (a, b) => a - b;
5 // lub
6 module.exports = {
7   add: (a, b) => a + b,
8   sub: (a, b) => a - b,
9 };
10 // lub
11 exports.add = (a, b) => a + b;
12 exports.sub = (a, b) => a - b;
13 // ale nie tak!
14 exports = {
15   add: (a, b) => a + b,
16   sub: (a, b) => a - b,
17 };
18 // ale nie tak!
19 exports = 'Ala ma kota';
```

MODUŁ (module)

- wbudowany (fs, os, itd...)
- własny
- zewnętrzny (lokalny/globalny)

MODUŁY WBUDOWANE



```
1 console.log('start app');  
2  
3 const os = require('os');  
4 const username = os.userInfo().username;  
5 console.log(username);  
6  
7 const fs = require('fs');  
8 fs.writeFileSync('abc.txt', username);
```

<https://nodejs.org/dist/latest-v12.x/docs/api/>

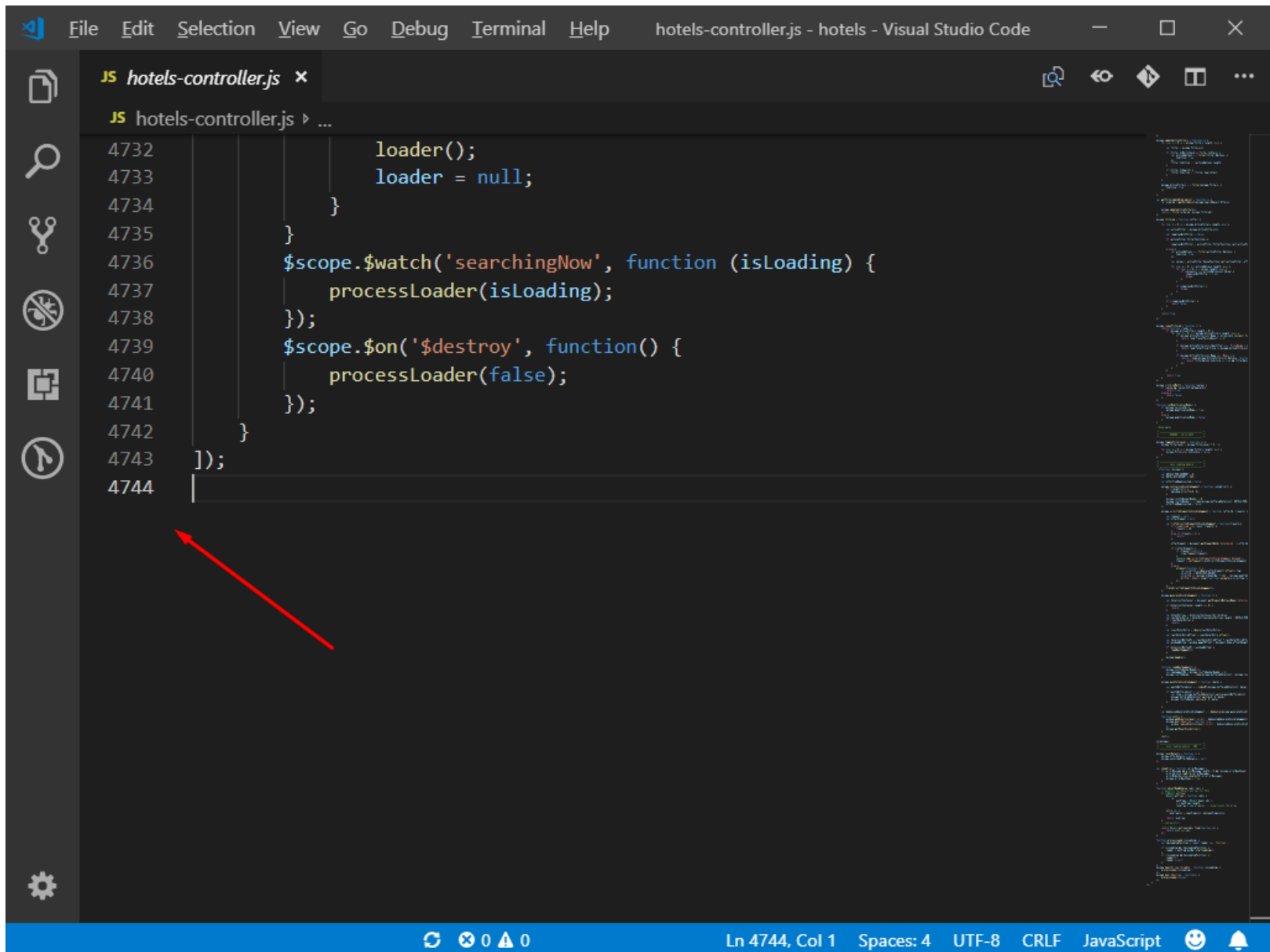
WŁASNE MODUŁY



```
1 // app.js
2 console.log('start app');
3
4 const user = require('./user');
5
6 console.log(user.firstName);
7 console.log(user.lastName);
```



```
1 // user.js
2
3 module.exports = {
4   firstName: 'Jan',
5   lastName: 'Nowak',
6 };
7
8 module.exports.city = 'Bialystok';
```

ZEWNĘTRZNE MODUŁY





```
1 // npm init
2 // npm install lodash
3
4 console.log('start app');
5
6 const _ = require('lodash');
7
8 console.log(_.isString('Ala ma kota'));
9 console.log(_.isString(123));
```

Wiersz polecenia - node

```
C:\Users\Piotr>node
> module
Module {
  id: '<repl>',
  exports: {},
  parent: undefined,
  filename: null,
  loaded: false,
  children: [],
  paths:
    [ 'C:\\Users\\Piotr\\repl\\node_modules',
      'C:\\Users\\Piotr\\node_modules',
      'C:\\Users\\node_modules',
      'C:\\node_modules',
      'C:\\Users\\Piotr\\.node_modules',
      'C:\\Users\\Piotr\\.node_modules',
      'C:\\Program Files\\nodejs\\lib\\node' ] }
>
```

NPM

```
Wiersz polecenia
D:\podyplomowe\nodejs\scripts>npm

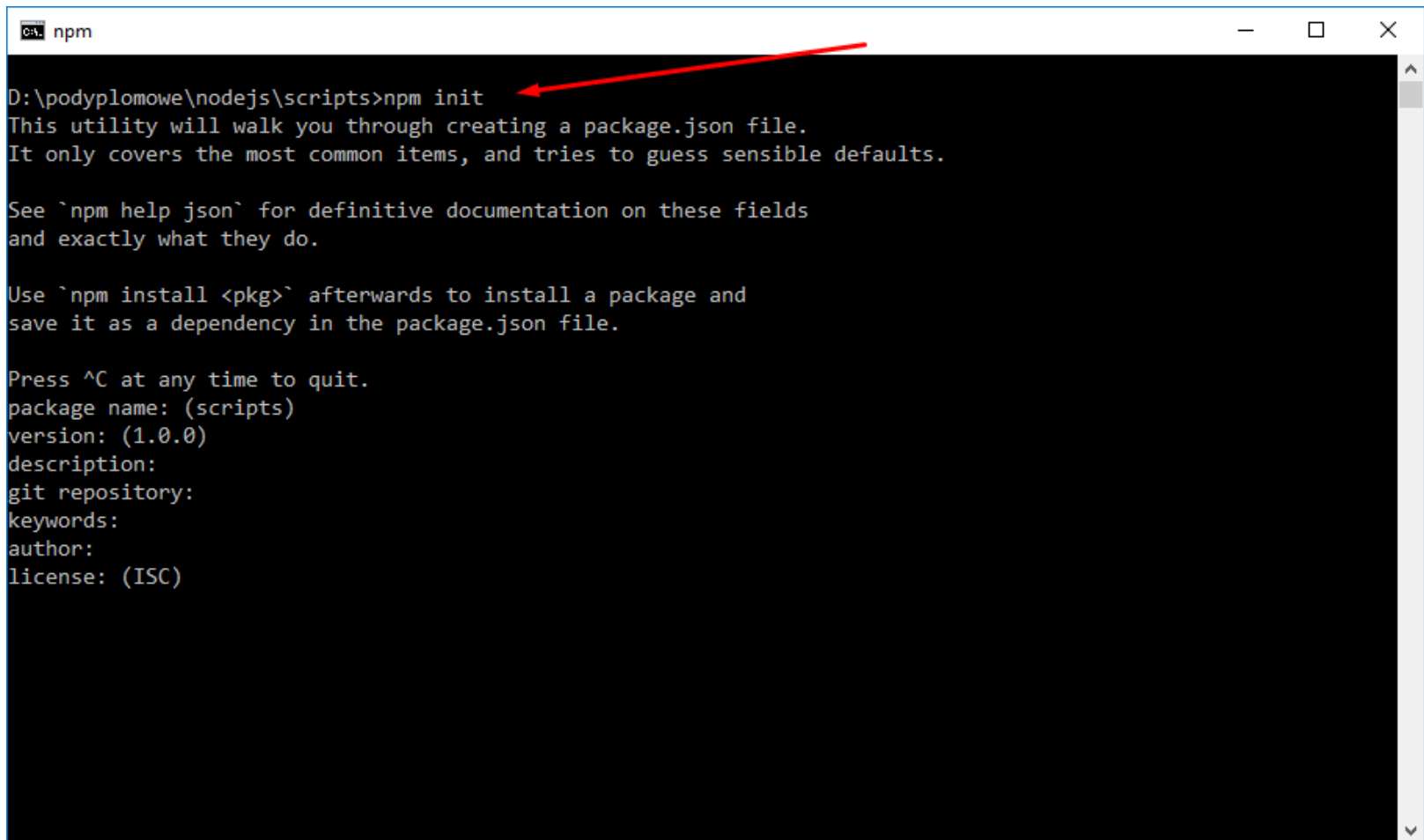
Usage: npm <command>

where <command> is one of:
  access, adduser, audit, bin, bugs, c, cache, ci, cit,
  completion, config, create, ddp, dedupe, deprecate,
  dist-tag, docs, doctor, edit, explore, get, help,
  help-search, hook, i, init, install, install-test, it, link,
  list, ln, login, logout, ls, outdated, owner, pack, ping,
  prefix, profile, prune, publish, rb, rebuild, repo, restart,
  root, run, run-script, s, se, search, set, shrinkwrap, star,
  stars, start, stop, t, team, test, token, tst, un,
  uninstall, unpublish, unstar, up, update, v, version, view,
  whoami

npm <command> -h  quick help on <command>
npm -l           display full usage info
npm help <term>  search for help on <term>
npm help npm     involved overview

Specify configs in the ini-formatted file:
  C:\Users\Piotr\.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@6.4.1 C:\Users\Piotr\AppData\Roaming\npm\node_modules\npm
```



```
cmd npm
D:\podyplomowe\nodejs\scripts>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (scripts)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
```




```
Wiersz polecenia
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (scripts)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to D:\podyplomowe\nodejs\scripts\package.json:

{
  "name": "scripts",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}

Is this OK? (yes)

D:\podyplomowe\nodejs\scripts>
```



```
1 {  
2   "name": "abc",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1"  
8   },  
9   "author": "",  
10  "license": "ISC"  
11 }
```


lodash

4.17.11 • Public • Published a month ago

Readme

0 Dependencies

77 010 Dependents

104 Versions

lodash v4.17.11

The **Lodash** library exported as **Node.js** modules.

Installation

Using npm:

```
$ npm i -g npm
$ npm i --save lodash
```

install

> npm i lodash

↓ weekly downloads

18 499 683

version	license
4.17.11	MIT

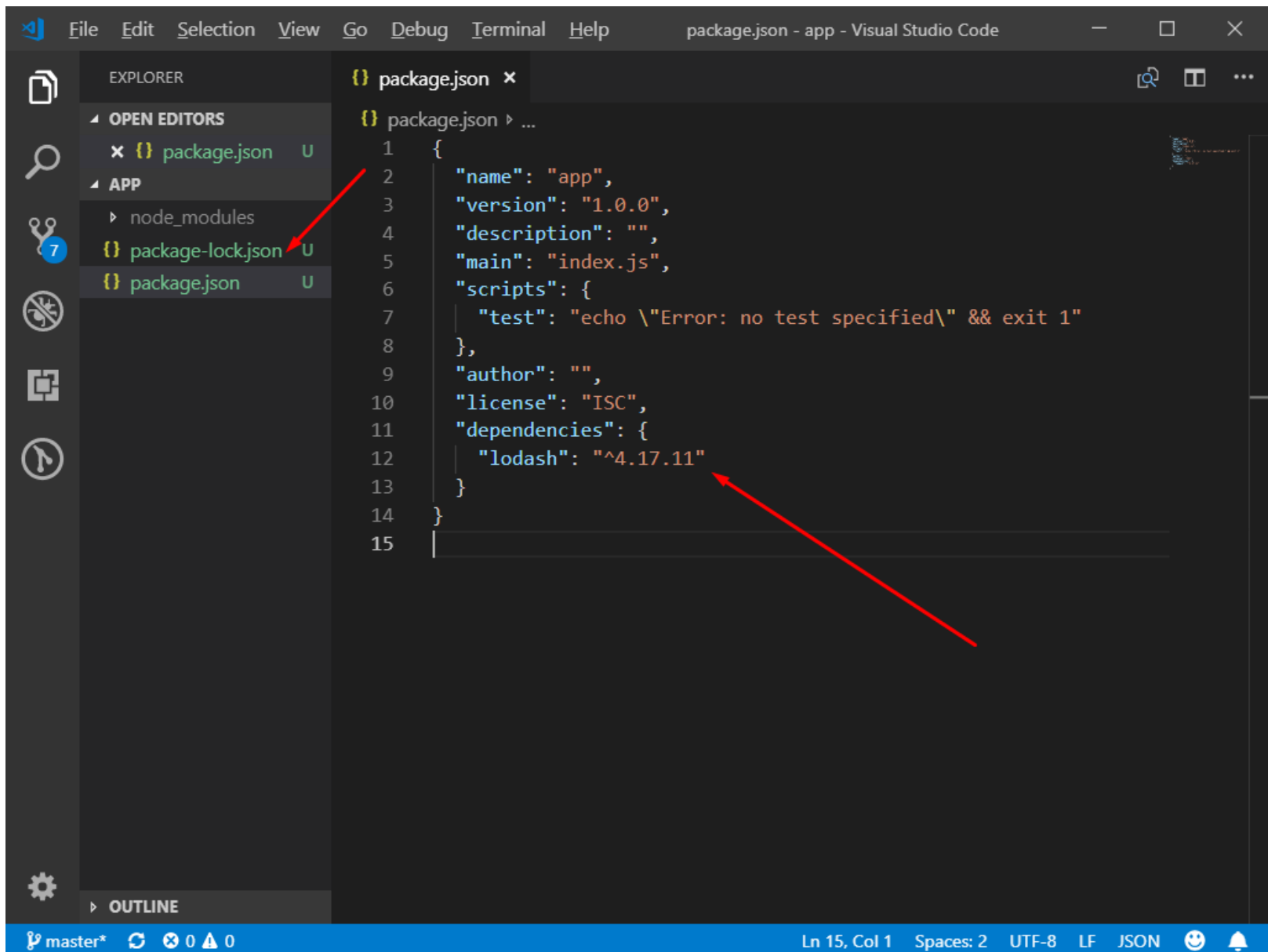
open issues	pull requests
1	3

Wiersz polecenia

```
D:\podyplomowe\nodejs\scripts\app>npm install lodash
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN app@1.0.0 No description
npm WARN app@1.0.0 No repository field.

+ lodash@4.17.11
added 1 package from 2 contributors and audited 1 package in 1.575s
found 0 vulnerabilities

D:\podyplomowe\nodejs\scripts\app>_
```



Wiersz polecenia

```
D:\podyplomowe\nodejs\scripts\app> npm install --global nodemon
C:\Users\Piotr\AppData\Roaming\npm\nodemon -> C:\Users\Piotr\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js

> nodemon@1.18.5 postinstall C:\Users\Piotr\AppData\Roaming\npm\node_modules\nodemon
> node bin/postinstall || exit 0

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\nodemon\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ nodemon@1.18.5
added 233 packages from 136 contributors in 8.516s

D:\podyplomowe\nodejs\scripts\app>
```