



FRONT-END **Level UP**

Welcome to the jungle!

Oleg Ruzhytskyi

Node.js

Wieloplatformowe środowisko uruchomieniowe o otwartym kodzie do tworzenia aplikacji typu server-side napisanych w języku JavaScript. Przyczynił się do stworzenia paradygmatu „JavaScript everywhere” umożliwiając programistom tworzenie aplikacji w obrębie jednego języka programowania zamiast polegania na odrębnych po stronie serwerowej.

Node.js

Node.js składa się z [silnika V8](#), biblioteki libUV oraz kilku innych bibliotek.

V8 – otwarty silnik JavaScript, stworzony i rozwijany przez Google, używany w przeglądarce Google Chrome.

NPM

Domyślny manager pakietów dla środowiska Node.js

Alternatywą jest Yarn. Był stworzony przez Facebook. Bierze z NPM wszystko to co dobre i dokłada do tego swoje rozwiązania.

package.json

Plik który zawiera informacje na temat naszego projektu.

```
$ npm init
```

```
{
  "name": "frontend-level-up",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Oleg Ruzhytskyi",
  "license": "ISC"
}
```

package.json

Warto dodać pole - "private": true

Jest to sposób na zapobieganie przypadkowej publikacji prywatnych repozytoriów.

```
{  
  "name": "frontend-level-up",  
  "version": "1.0.0",  
  "description": "",  
  "private": true,  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "Oleg Ruzhytskyi",  
  "license": "ISC"  
}
```

package.json

```
$ npm install package
```

```
{
  "name": "frontend-level-up",
  "version": "1.0.0",
  "description": "",
  "private": true,
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Oleg Ruzhytskyi",
  "license": "ISC",
  "dependencies": {
    "package": "^1.0.1"
  }
}
```

package.json

Zainstalować wszystkie paczki z
package.json

```
$ npm install
```

Zainstalować paczki tylko z listy
“dependencies”

```
$ npm install --production
```

Zainstalować paczkę do listy
“dependencies”

```
$ npm install package
```

Zainstalować paczkę do listy
“devDependencies”

```
$ npm install package  
--save-dev
```


package.json

```
{
  "name": "frontend-level-up",
  "version": "1.0.0",
  "description": "",
  "private": true,
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Oleg Ruzhytskyi",
  "license": "ISC",
  "dependencies": {
    "package": "^1.0.1"
  },
  "devDependencies": {
    "package2": "^0.28.3"
  }
}
```

NPM - przydatne info

Instalacja paczki globalnie

```
$ npm install package  
--global
```

Instalacja konkretnej wersji paczki

```
$ npm install package@1.1.1
```

Skróty

```
$ npm i package -g
```

Info o tym jak używać

```
$ npm help
```

Instalacja pary paczek na raz

```
$ npm i package package1
```

Task runners

- Grunt - <https://gruntjs.com/>
- Gulp - <https://gulpjs.com/>
- Npm scripts - <https://docs.npmjs.com/misc/scripts>

package.json

\$ npm run script

```
{
  "name": "frontend-level-up",
  "version": "1.0.0",
  "description": "",
  "private": true,
  "scripts": {
    "start": "node start.js",
    "build": "node build.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Oleg Ruzhytskyi",
  "license": "ISC",
  "dependencies": {
    "package": "^1.0.1"
  },
  "devDependencies": {
    "package2": "^0.28.3"
  }
}
```

Web application builder

- Webpack - <https://webpack.js.org/>
- Parcel - <https://parceljs.org/>
- Rollup - <https://rollupjs.org/>

Webpack

```
$ npm i webpack webpack-cli
```

```
{  
  "name": "frontend-level-up",  
  "version": "1.0.0",  
  "description": "",  
  "private": true,  
  "scripts": {  
    "build": "webpack --mode production",  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "Oleg Ruzhytskyi",  
  "license": "ISC",  
  "dependencies": {  
    "webpack": "^4.42.1",  
    "webpack-cli": "^3.3.11"  
  }  
}
```

```
> node_modules  
  ✓ src  
    JS index.js  
    {} package-lock.json  
    {} package.json  
    📦 webpack.config.js
```

Webpack - config

<https://webpack.js.org/configuration/>

webpack.config.js

```
const path = require('path')

module.exports = {
  entry: './src/index.js',
  output: {
    filename: 'index.js',
    path: path.resolve(__dirname, 'dist')
  },
}
```

Webpack - config

HTML

```
$ npm i html-webpack-plugin --save-dev
```

```
$ npm run build
```

```
▼ dist
  <> index.html
  JS index.js
  > node_modules
  ▼ src
    <> index.html
    JS index.js
  {} package-lock.json
  {} package.json
  📦 webpack.config.js
```

```
const path = require('path')
const HtmlWebpackPlugin = require('html-webpack-plugin')

module.exports = {
  entry: './src/index.js',
  output: {
    filename: 'index.js',
    path: path.resolve(__dirname, 'dist')
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/index.html',
      filename: './index.html'
    })
  ]
}
```


Babel

Babel – darmowy i otwarty transpilator JavaScript, którego głównym celem jest konwersja ECMAScript 2015+ (ES6+) na kod zgodny z ES5, czyli ze starszymi przeglądarkami internetowymi. Tym samym Babel jest narzędziem umożliwiającym korzystanie z najnowszych funkcji języka JavaScript zanim zostaną one wszędzie zaimplementowane)

Babel

```
$ npm i babel-loader @babel/core  
@babel/preset-env --save-dev
```

```
{  
  "name": "frontend-level-up",  
  "version": "1.0.0",  
  "description": "",  
  "private": true,  
  "scripts": {  
    "build": "webpack --mode production",  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "Oleg Ruzhytskyi",  
  "license": "ISC",  
  "dependencies": {  
    "webpack": "^4.42.1",  
    "webpack-cli": "^3.3.11"  
  },  
  "devDependencies": {  
    "@babel/core": "^7.9.0",  
    "@babel/preset-env": "^7.9.0",  
    "babel-loader": "^8.1.0",  
    "html-webpack-plugin": "^4.0.4"  
  }  
}
```

```
const path = require('path')  
const HtmlWebPackPlugin = require('html-webpack-plugin')  
  
module.exports = {  
  entry: './src/index.js',  
  output: {  
    filename: 'index.js',  
    path: path.resolve(__dirname, 'dist')  
  },  
  module: {  
    rules: [  
      {  
        test: /\.js$/,  
        exclude: /node_modules/,  
        use: {  
          loader: 'babel-loader',  
          options: {  
            presets: ['@babel/preset-env']  
          }  
        }  
      }  
    ]  
  },  
  plugins: [  
    new HtmlWebPackPlugin({  
      template: './src/index.html',  
      filename: './index.html'  
    })  
  ]  
}
```

@babel/preset-env

Preset to przygotowane dla nas gotowe ustawienia. Daje możliwość babelu zrozumieć jak transformować kod.

“browserslist” - Konfiguracja do udostępniania docelowych przeglądarek i wersji Node.js między różnymi narzędziami front-endu.

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

```
{
  "name": "frontend-level-up",
  "version": "1.0.0",
  "description": "",
  "private": true,
  "scripts": {
    "build": "webpack --mode production",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Oleg Ruzhytskyi",
  "license": "ISC",
  "dependencies": {
    "webpack": "^4.42.1",
    "webpack-cli": "^3.3.11"
  },
  "devDependencies": {
    "@babel/core": "^7.9.0",
    "@babel/preset-env": "^7.9.0",
    "babel-loader": "^8.1.0",
    "html-webpack-plugin": "^4.0.4"
  },
  "browserslist": [
    "defaults"
  ]
}
```

Webpack - config

\$ npm i css-loader style-loader --save-dev

```

├─ dist
│  ├─ index.html
│  └─ JS index.js
├─ node_modules
├─ src
│  └─ styles
│     └─ # style.css
│        ├─ index.html
│        └─ JS index.js
├─ {} package-lock.json
├─ {} package.json
└─ 📦 webpack.config.js
```

```
const path = require('path')
const HtmlWebPackPlugin = require('html-webpack-plugin')

module.exports = {
  entry: './src/index.js',
  output: {
    filename: 'index.js',
    path: path.resolve(__dirname, 'dist')
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        exclude: /node_modules/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }
      },
      {
        test: /\.css$/,
        exclude: /node_modules/,
        use: ['style-loader', 'css-loader']
      }
    ]
  },
  plugins: [
    new HtmlWebPackPlugin({
      template: './src/index.html',
      filename: './index.html'
    })
  ]
}
```

Webpack - config

Loaders

<https://webpack.js.org/loaders/>

Plugins

<https://webpack.js.org/plugins/>

Webpack - modes

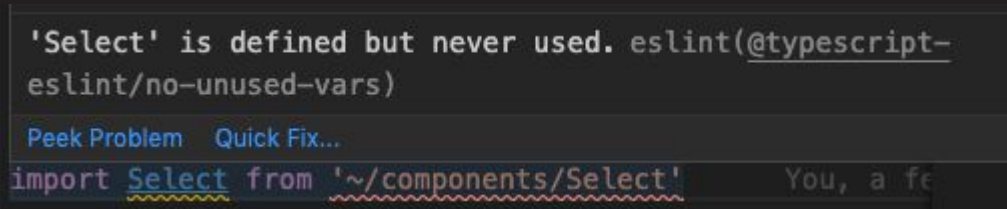
- Development
- Production
- None

<https://webpack.js.org/configuration/mode/>

Linters

Narzędzie do analizy kodu które odznacza błędy, bugi, stylistykę i konstrukcje kodu.

- JSLint - <https://jshint.com/>
- JSHint - <https://jshint.com/>
- ESLint - <https://eslint.org/>



The screenshot shows an IDE interface. At the top, a light blue error message states: "'Select' is defined but never used. eslint(@typescript-eslint/no-unused-vars)". Below this message are two buttons: "Peek Problem" and "Quick Fix...". The code editor below shows the line: `import Select from '~/components/Select'`. The word `Select` is underlined with a red wavy line, indicating the error. To the right of the code, the text "You, a fe" is partially visible.

```
'Select' is defined but never used. eslint(@typescript-  
eslint/no-unused-vars)  
Peek Problem Quick Fix...  
import Select from '~/components/Select' You, a fe
```

Gdzie szukać wiedzy?

<https://github.com/kamranahmedse/developer-roadmap>

<https://github.com/getify/You-Dont-Know-JS>

Mainlingi:

- CSS weekly
- Frontend focus
- JavaScript weekly
- Node weekly
- Web tools weekly
- Web design weekly



Dziękuję!

Pytania ?