

# Własne narzędzie CLI

W NodeJS poniżej 30 minut

by Łukasz Rybka

# Agenda

- Cel
- Podstawowa konfiguracja
- Dostępne narzędzia
- Konfiguracja
- Instalacja
- Rozszerzalność

# CEL



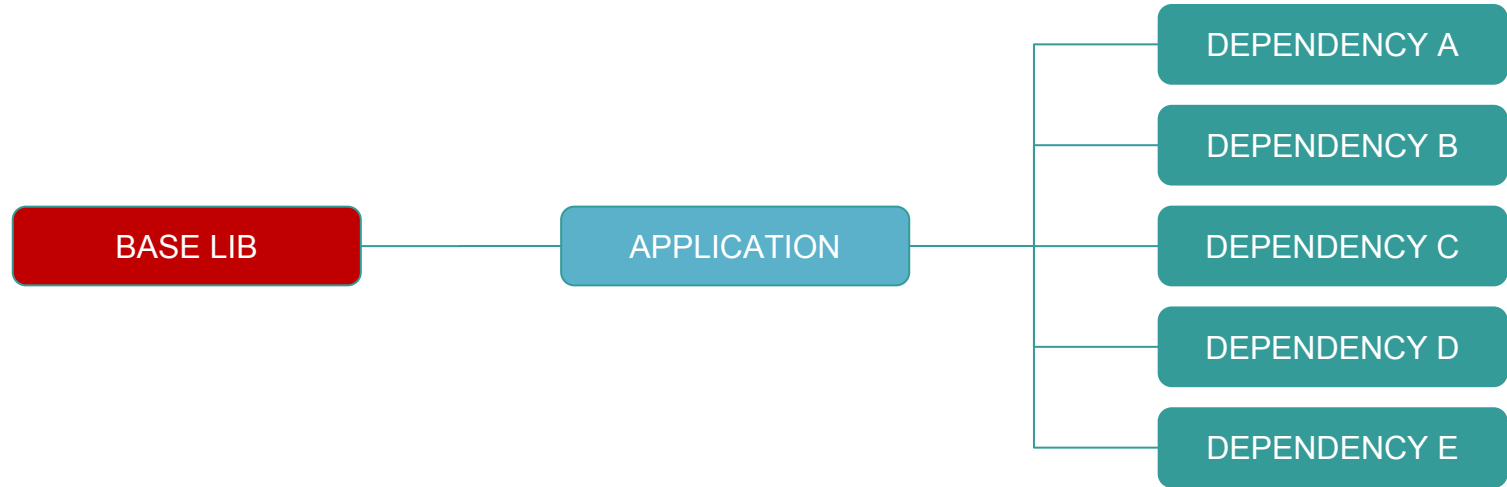
“*A man who dares to waste one hour of time has not discovered the value of life.*

*~ Charles Darwin*

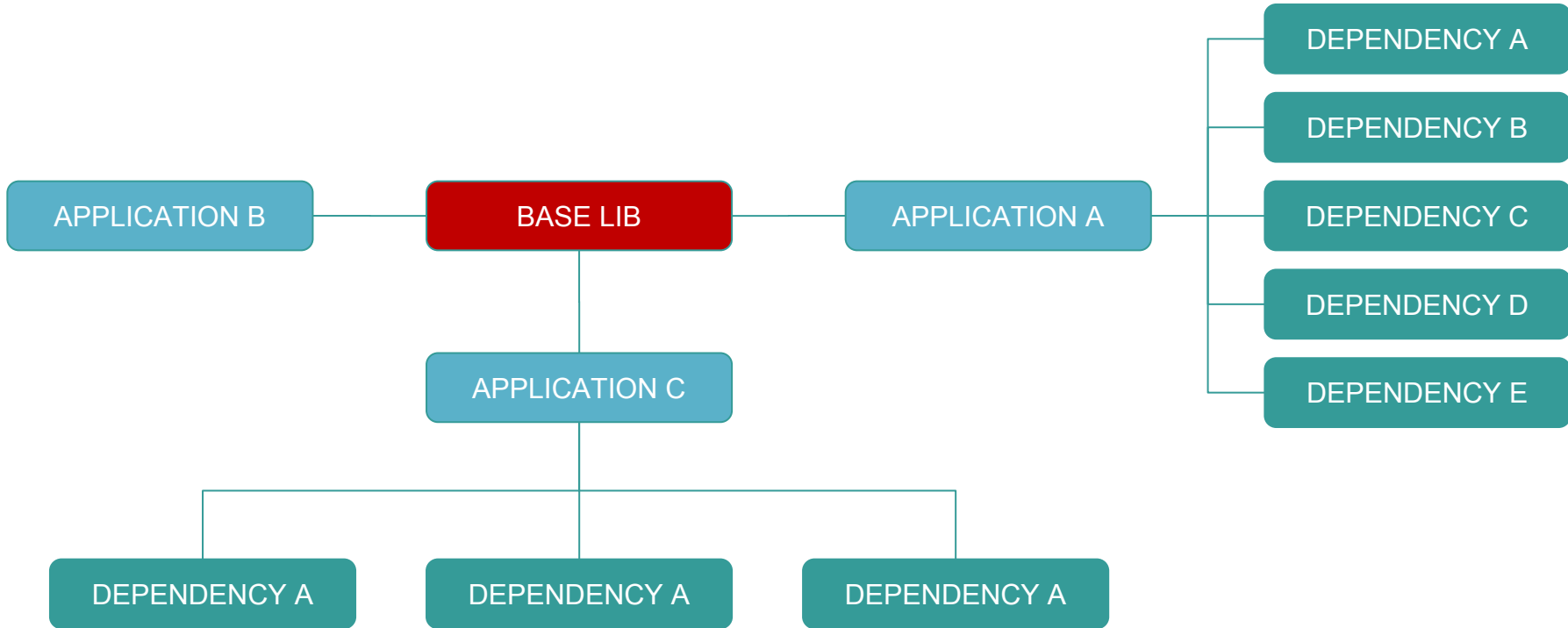
# Typowe problemy

- Za dużo pracy dla zbyt małego zespołu
- Duża ilość aplikacji === duża ilość błędów
- Zmieniająca się architektura
- Zmieniająca się konfiguracja
- Obszerny nowy stack technologiczny

# Architektura



# Architektura



# Automatyzacja i unifikacja

- Skrypty NodeJS w package.json?



# Automatyzacja i unifikacja

- ~~Skrypty NodeJS w package.json?~~
- Task runner (Grunt/Gulp/...)?

# Automatyzacja i unifikacja

- ~~Skrypty NodeJS w package.json?~~
- ~~Task runner (Grunt/Gulp/...)?~~
- Własne narzędzie CLI?

# Automatyzacja i unifikacja

- ~~Skrypty NodeJS w package.json?~~
- ~~Task runner (Grunt/Gulp/...)?~~
- Własne narzędzie CLI!

# KONFIGURACJA

# Inicjalizacja

- Aplikacje CLI w NodeJS inicjalizujemy identycznie jak standardową aplikację - za pomocą polecenia *npm init*

# Biblioteki

- Do potrzeb demonstracyjnych skorzystamy z dwóch bibliotek:
  - Commander - biblioteka upraszczająca pracę z konsolą (parsowanie argumentów)
  - Chalk - biblioteka ułatwiająca stylowanie komunikatów wypisywanych na konsoli

# package.json

```
{
  "name": "careercon",
  "version": "1.0.0",
  "description": "CLI tool for CareerCon",
  "author": "",
  "license": "ISC",
  "dependencies": {
    "chalk": "^2.4.1",
    "commander": "^2.19.0"
  },
  "preferGlobal": true,
  "engines": {
    "node": "8 || 9 || 10"
  },
  "bin": {
    "careercon": "./bin/index.js"
  }
}
```

# bin/index.js

```
#!/usr/bin/env node

const path = require('path');
const pkg = require( path.join(__dirname, '../package.json') );
```



# Pierwsza komenda

```
program
  .command('organizer')
  .alias('org')
  .description('Displays information about organizers')
  .action(() => {
    console.log('BAMT Sp. z o.o. S.K.');
```

```
    console.log('ul. Płocka 5A');
```

```
    console.log('01-231 Warszawa');
```

```
  });
```

# Obsługa braku komendy

```
if (!process.argv.slice(2).length) {  
  console.warn(chalk.red('No commands specified!'));  
  console.warn('');  
  
  program.outputHelp(help => chalk.yellow(help));  
  
  process.exit(1);  
}
```

# Obsługa błędnej komendy

```
program.on('command:*', () => {  
  console.error(chalk.red('Invalid command: %s'));  
  console.error(chalk.red('See --help for a list of available commands.'));  
  
  process.exit(1);  
});
```

# INSTALACJA

# Repozytorium

<https://github.com/Smoczysko/careercon-cli>

# Instalacja - link

## npm-link

Symlink a package folder

### SYNOPSIS

```
npm link (in package dir)
npm link [<@scope>/]<pkg>[@<version>]

alias: npm ln
```

### DESCRIPTION

Package linking is a two-step process.

First, `npm link` in a package folder will create a symlink in the global folder `{prefix}/lib/node_modules/<package>` that links to the package where the `npm link` command was executed. (see [npm-config](/misc/config) for the value of `prefix`). It will also link any bins in the package to `{prefix}/bin/{name}`.

# Instalacja - npm install

```
npm install --global
```

```
git+ssh://
```

```
git@github.com:Smoczysko/careercon-cli.git
```

# EXTENSIONS



# Extensions - akcja

```
module.exports = () => {  
  console.log('Hello from Gdańsk!');  
};
```

# Extensions - json

```
{  
  "commands": [{  
    "name": "gdansk",  
    "alias": "gd",  
    "description": "Shows information about Gdansk",  
    "action": "cli/gdansk.js"  
  }]  
}
```

# Extensions - obsługa

```
const loadAppConfiguration = () => {  
  try {  
    return require(path.resolve(process.cwd(), CONFIG_FILE_NAME));  
  } catch (ex) {  
    return null;  
  }  
};
```

# Repozytorium

[https://github.com/  
Smoczysko/careercon-cli-gdansk](https://github.com/Smoczysko/careercon-cli-gdansk)

# Pytania?

# Dzięki!

<https://infoshareacademy.com>

<http://www.dragonia.org.pl>