We make IT

JCommerce

Tworzenie niezależnego środowiska do developmentu aplikacji frontendowej

Katowice, 18 grudnia 2015r.

- Typy frontendowych środowisk developerskich
- Elementy niezależnego środowiska do developmentu aplikacji frontendowej
- Praca z fake’ym API RESTowym

Frontendowe środowiska developerskie

Non-SPA Web App

Aplikacja nie jest podzielona na część frontendową i backendową. Z reguły skupia się ona wokół jednego frameworka.

Czynniki uzależniające:

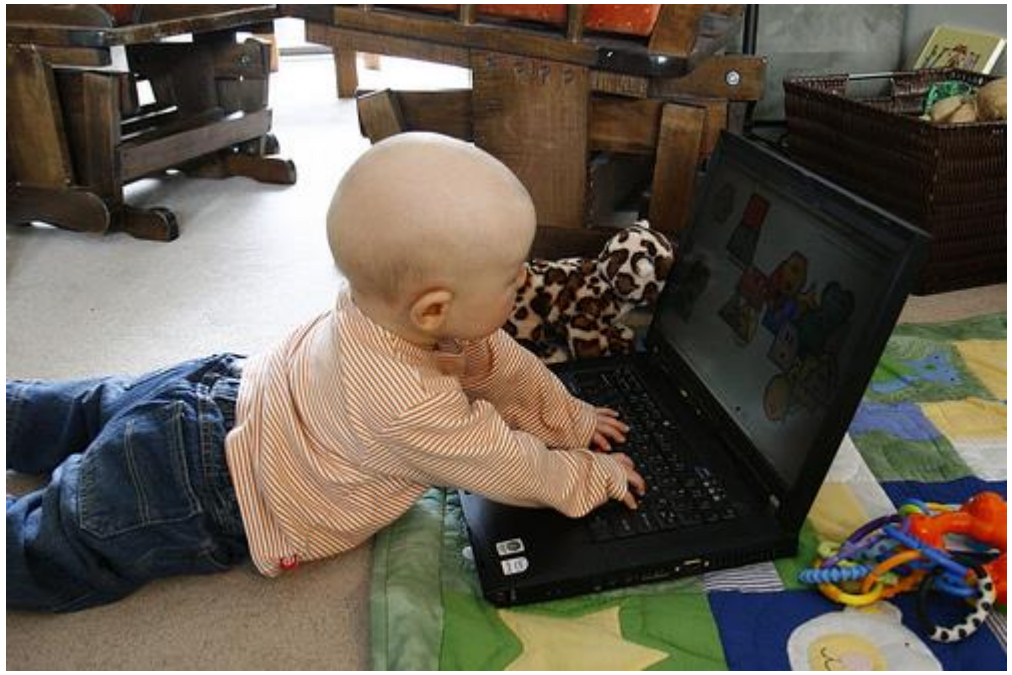
- HTML zawiera zmienne (a czasem nawet logikę) pochodzące z technologii backendowej
- Aplikacja nie serwuje danych, tylko wysyła gotowy markup do wyrenderowania w przeglądarce.

SPA osadzone w Web App

Aplikacja składa się z dwóch częśći: frontend app i backend app, które komunikują się poprzez API

Czynniki uzależniające:

- ładowanie bibliotek js poprzez wrapery backendowe
- przygotowywanie dystrybucji przez biblioteki backendowe
- konieczność serwowania aplikacji poprzez serwer backendowy
- konieczność pracy na prawdziwej bazie danych
- testowanie integracyjne przy użyciu backendowych bibliotek

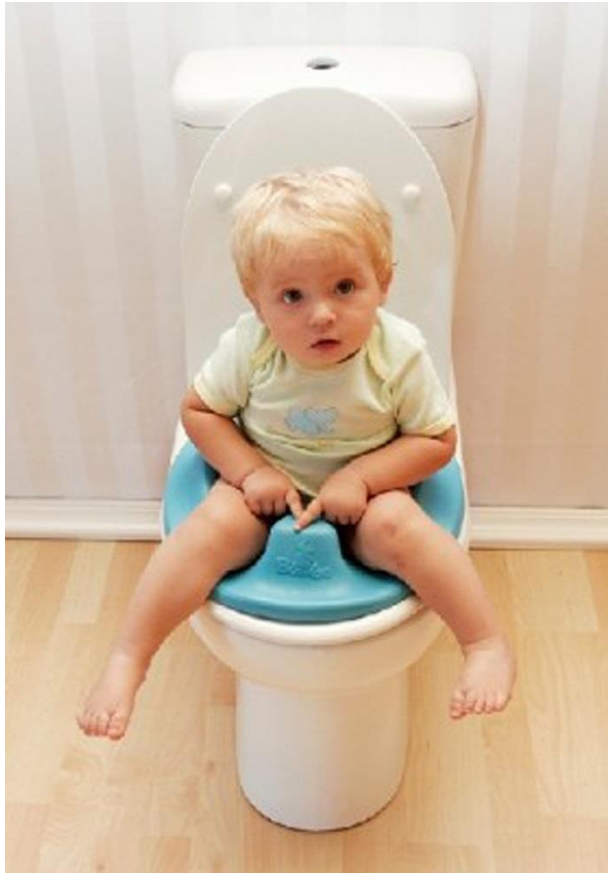


SPA standalone

Aplikacja składa się z dwóch częśći: frontend app i backend app, które komunikują się poprzez API

Czynniki niezależności, czyli do czego backend nam nie jest potrzebny:

- instalacja bibliotek frontendowych poprzez dedykowane JS narzędzia (npm, bower, ...)
- serwowanie aplikacji (grunt / gulp, connect, livereload)
- projektowanie i praca z API (json-server)
- budowanie dystrybucji aplikacji (concatenation, obfruscation, minimifaction, wersjonowanie assetów)
- testowanie integracyjne (jasmine / mocha)



Dlaczego warto być niezależnym?

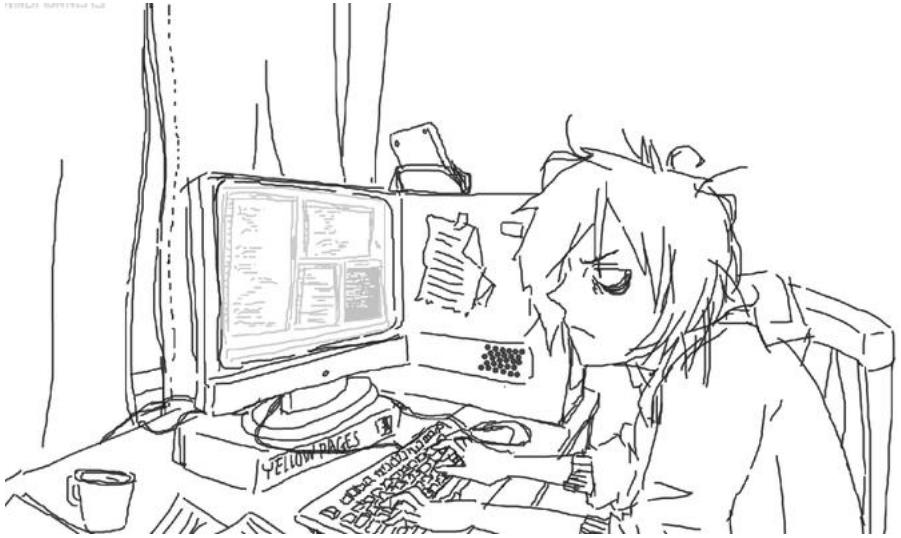
- backend developerzy nie blokują frontend developerów
- pełna kontrola nad implementowaną aplikacją
- to daje szczęście

Frontendowe środowiska developerskie

Co daje niezależność?

- Node.js z managerem npm
- Korzystanie z narzędzi do automatyzacji procesów np. grunt / gulp wraz z licznymi pluginami
- RESTowe fake API, które możemy w każdym momencie dostosować do naszych potrzeb

GRUNT



Setup manualny

czasochłonny, musimy wiedzieć czego chcemy

Setup automatyczny

szybki, skupiony na best practices

We make IT JCommerce

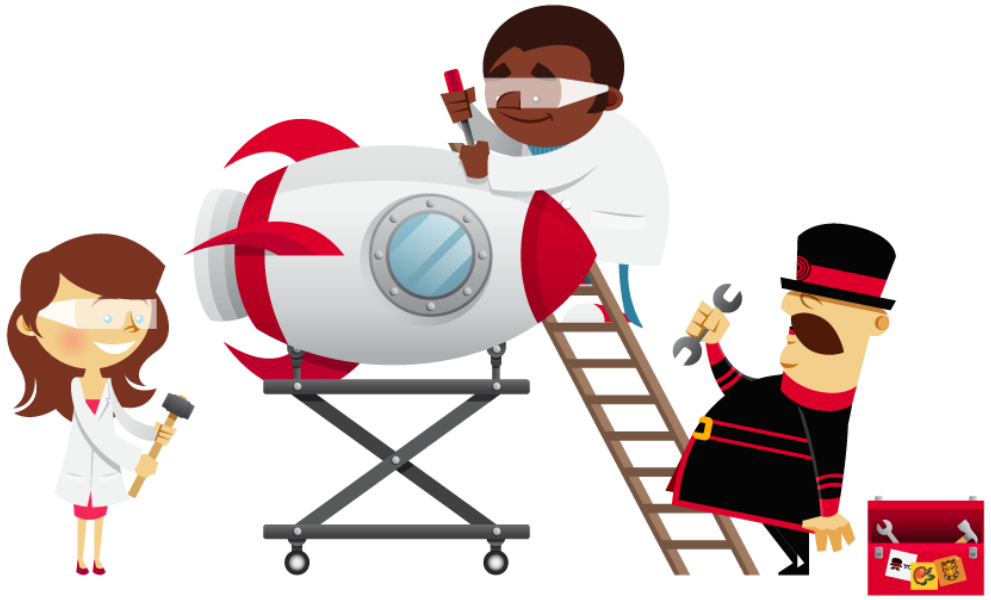
Setup aplikacji z Yeoman: przygotowanie

Instalacja niezbędnych pakietów npm

```
npm install -g grunt-cli bower yo
generator-karma generator-angular
```

Generowanie aplikacji

```
yo angular:app kittens
```



Setup aplikacji z Yeoman: co dostajemy

Serwowania aplikacji

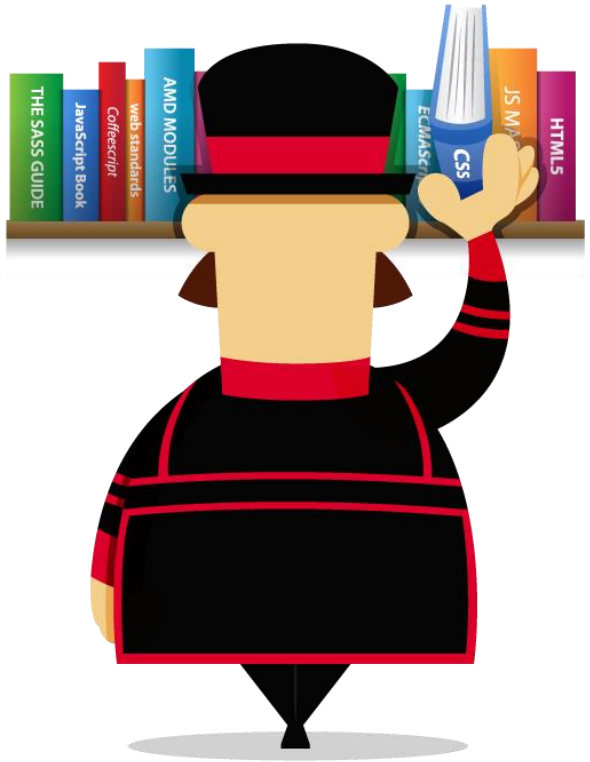
```
grunt serve
```

Testowanie aplikacji

```
grunt test
```

Przygotowanie dystrybucji

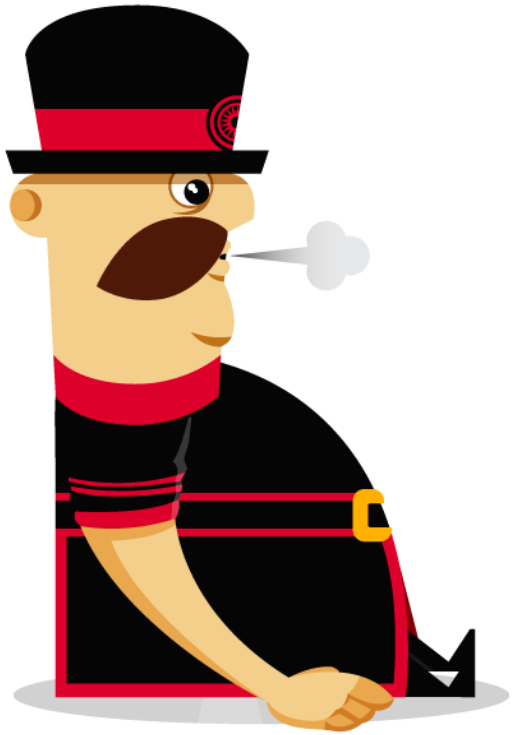
```
grunt build
```



Setup aplikacji z Yeoman: czego brakuje

Wygenerowana aplikacja nie posiada zdefiniowanego źródła danych.

Jeśli planujemy korzystać z RESTowego API, które udostępni nam backend, to czy wpłynie to na niezależność?



We make IT JCommerce

json-server – fake’owe API dla frontendu

json-server: możliwości

- API RESTowe, opierające się o fake'ową bazę danych zapisanąw pliku JSON
- Podstawowe operacje na kolekcjach danych np. Paginacja, sortowanie, filtrowanie
- Definiowanie customowych ścieżek, które mogą być mapowane na istniejące endpointy

```
{
  "posts": [
    { "id": 1, "title": "json-server", "author": "typicode" }
  ],
  "comments": [
    { "id": 1, "body": "some comment", "postId": 1 }
  ],
  "profile": { "name": "typicode" }
}
```

json-server: setup z grunt'em

Warto również skorzystać z pakiet grunt-connect-proxy, który pozwala na:

- ominięcie problemu CORS, gdyż json-server i aplikacja frontendowa będą działać na różnych portach.
- elastyczne przełączanie się pomiędzy prawdziwym a fake'owym API np.

```
grunt serve --api=fake
```

Konfiguracja w Gruntfile.js

We make IT JCommerce

Pytania

We make IT JCommerce

We make IT JCommerce

I ty możesz zostać niezależnym developerem!

Tomasz Borowski

FrontEnd Developer

tomasz.borowski@jcommerce.pl