



Angular 2

Moja pierwsza aplikacja

Autorzy:

Michał Michalczuk, Mateusz Turzyński

3 kwietnia 2017

goyello



Plan na dzisiaj

Czas warsztatów: 2h 30 min

- Web Client- Web Server: gdzie leży jaki kod, gdzie działa Angular
- Krótko o Angular – do czego warto go użyć
- TypeScript: opcjonalnie typowany JavaScript, co on ma wspólnego z Angular-em?
- Angular-cli: szybkie i wygodne narzędzie do budowania i uruchamiania projektów Angular
- Notes list – klasycznie prosta przykładowa aplikacja

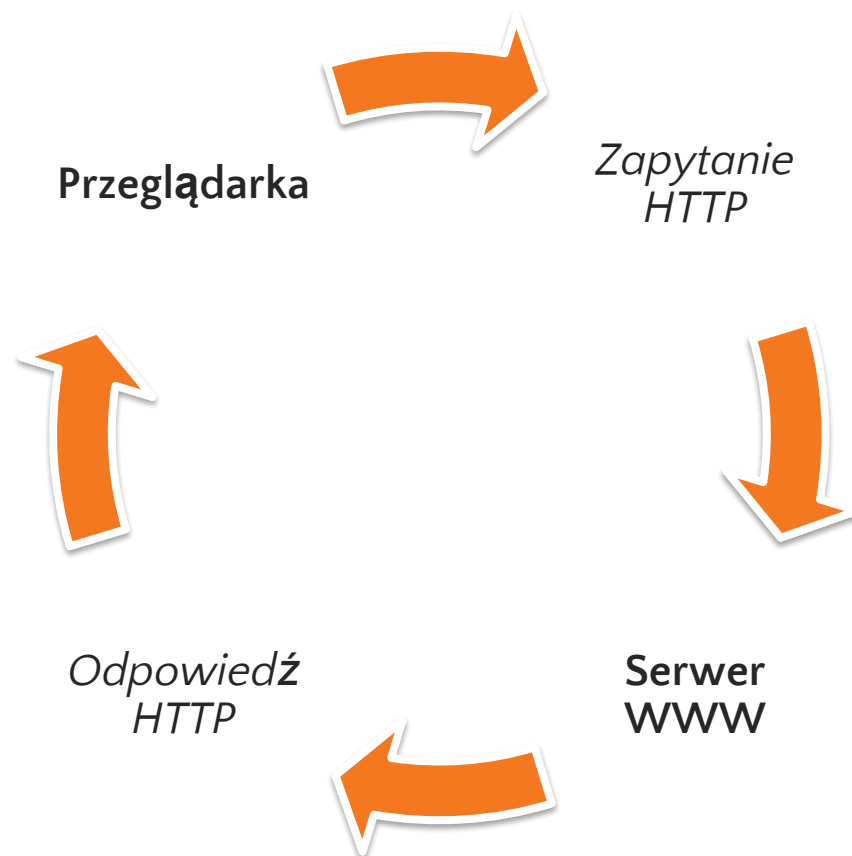


Request – Response

Czyli jak działa Internet

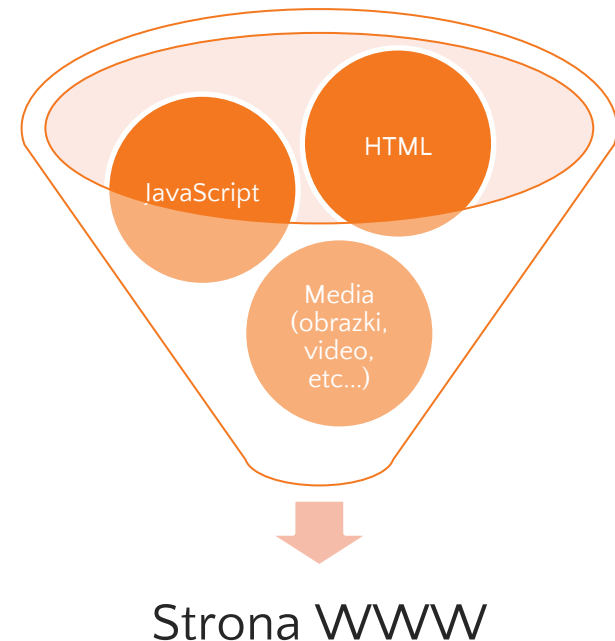
Protokół HTTP – co musimy o nim wiedzieć?

- Protokół HTTP pozwala nam przeglądać strony WWW
- Opisuje sposób komunikowania się **przeglądarki** (*client side*) z **serwerem www** (server side)
- Działa w trybie **Request-Response**
- Przeglądarka z reguły musi wysłać *kilkadziesiąt* żądań HTTP, żeby wyświetlić jedną stronę



Co potrafi przeglądarka WWW?

- Wysłać żądanie do serwera *WWW*
- Odebrać i przetworzyć odpowiedź
- Wyświetlić (wyrenderować) zwrócony kod *HTML*
- **Wykonać** zwrócony kod *JavaScript*
- Odebrać dane od użytkownika (formularze)



Strona WWW



Single Page Application

Czyli nadrabiamy niedoskonałości HTTP

Jak działają aplikacje typu Facebook, Gmail, etc...

- Serwer zwraca osobno *dane (model)* (np posty na ścianie) i *szablony HTML*
- *Szablon HTML (view)* opisuje jak dane powinny być prezentowane
- Kod *JavaScript* interpretuje szablon i wyświetla pełną treść użytkownikowi
- Przeglądarka prosi o kolejne dane w tle (AJAX)

- Przykładowy widok (*view*)

```
<div class="panel panel-default">
  <div class="panel-heading">
    <h3>
      {{ selectedNote.title }}
    </h3>
    <h4>
      {{ selectedNote.createdOn | date: "dd/MM/yyyy HH:mm" }}
    </h4>
  </div>
  <div class="panel-body">
    <p class="text-justify">
      {{ selectedNote.content }}
    </p>
  </div>
</div>
```

- Przykładowy model

```
{
  title: "Ala ma kota",
  createdOnDate: "2017-04-04T12:00:00.000",
  author: "Goyello",
  description: "just another JSON document"
}
```



- Framework SPA
- Działa pod kontrolą przeglądarki dzięki *JavaScript*
- Znany też pod nazwą *Angular 2*
- Rozwijany przez *Google*
- W dużym skrócie pozwala tworzyć dynamiczne i interaktywne strony WWW

<https://angular.io>





Język wyboru Angular 2

TypeScript ... w 15 min

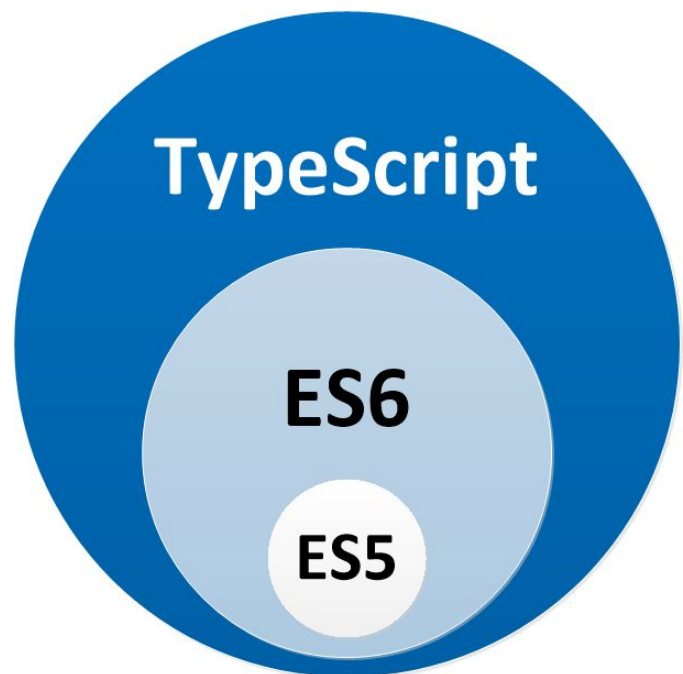


- Nadzbiór JavaScript
- Kod JavaScript jest legalnym TypeScript
- Od 2012 rozwijany przez Microsoft
- Otwarty kod na Github-ie
- v **2.2.2**

<https://github.com/Microsoft/TypeScript>



TypeScript a JavaScript



- ES = EcmaScript
- ES5 vs ES6
- TypeScript implementuje ES6 + ES2017 + więcej



TypeScript a JavaScript



- TypeScript musi zostać skompilowany do JavaScript
- Przeglądarka musi zrozumieć nasz kod

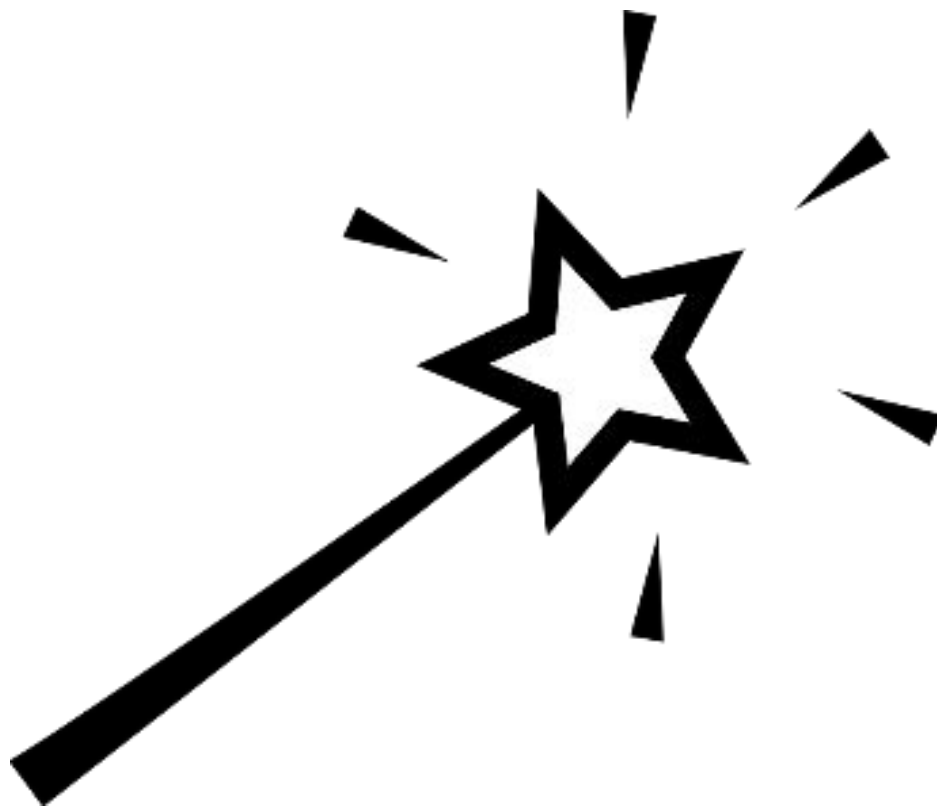
TypeScript: przykład

```
// class - like in ES6, or C# or Java or C++  
// export - to use it in other modules  
export class NotesService {  
    // access modifiers - wow  
    private notes: string[] = [];  
  
    add(text: string) {  
        this.notes.push(text);  
    }  
  
    // optional strong typing  
    get(): string[] {  
        return this.notes;  
    }  
}
```

The logo consists of a blue square with the white letters 'TS' inside it.

TS

Znacie wzorzec „moduł” w JS ?



TypeScript: przykład po skompilowaniu do ES5

```
"use strict";
var NotesService = (function () {
    function NotesService() {
        this.notes = [];
    }
    NotesService.prototype.add = function (text) {
        this.notes.push(text);
    };
    NotesService.prototype.get = function () {
        return this.notes;
    };
    return NotesService;
})();
exports.NotesService = NotesService;
//# sourceMappingURL=notes.service.js.map
|
```

TS

tsconfig.json – czyli mówimy kompilatorowi co ma robić


```
{
  "compilerOptions": {
    "module": "commonjs",
    "noImplicitAny": true,
    "removeComments": true,
    "outDir": "dist",
    "sourceMap": true
  },
  "include": [
    "**/*.ts"
  ],
  "exclude": [
    "node_modules"
  ]
}
```

- Jakie pliki
- Do czego ma kompilować
- Gdzie będzie wynik
- Więcej opcji
(<https://www.typescriptlang.org/docs/handbook/tsconfig-json.html>)



Użyjmy naszego NoteService

```
// import module ( ES6 way <3 )  
import { NotesService } from './notes.service';  
  
var service = new NotesService();  
  
service.add('first note');  
service.add('goyello note ... the second one');  
  
console.log(service.get());
```



Dodajmy jeszcze parę elementów
języka



Interfejsy

```
export interface NotesServiceInterface {
    add(text: string): void;
    get(): string[];
}

// class - like in ES6, or C# or Java or C++
// export - to use it in other modules.
// Now it has to implement interface
export class NotesService implements NotesServiceInterface {
    // access modifiers - wow
    private notes: string[] = [];

    add(text: string) {
        this.notes.push(text);
    }

    // optional strong typing
    get(): string[] {
        return this.notes;
    }
}
```

The logo consists of a blue square with the white letters 'TS' inside. Below the square is a small orange circle.

TS

Typy a trzymanie danych

```
export type Note = {  
  text: string,  
  createdOnDate: Date  
}
```

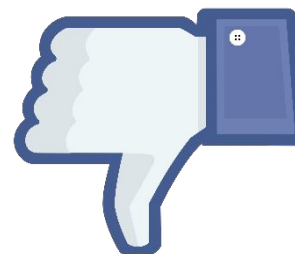
```
export interface Note {  
  text: string,  
  createdOnDate: Date  
}
```

```
export class Note {  
  text: string;  
  createdOnDate: Date;  
}
```



Proste modele.

Opisuj albo
jako `type` albo
`interface`.



Klas używaj
gdy mają
jakieś metody.

Dodajmy typ Note – do zamodelowania notatki

```
export class Note {  
    text: string;  
    createdOnDate: Date;  
}
```

Nasz serwis powinien używać `Note`. Zmieniamy interface który implementuje.

```
export interface NotesServiceInterface {  
    add(text: string): void;  
    get(): Note[];  
}
```



NotesService po małej zmianie

```
export class NotesService implements NotesServiceInterface {  
    // access modifiers - wow  
    private notes: Note[] = [];  
  
    add(text: string) {  
        const newNote = {  
            text: text,  
            createdOnDate: new Date()  
        } as Note;  
  
        this.notes.push(newNote);  
    }  
  
    // optional strong typing  
    get(): Note[] {  
        return this.notes;  
    }  
}
```

The TypeScript logo, consisting of a blue square with the white letters "TS" inside.

Małym podsumowaniem

- Opcjonalnie typowany język
- Nadzbiór JavaScript
- Trzeba go skompilować go JavaScript
- Używa modułów i importów ES6

Co więcej ?

- Rozbudowane zarządzanie typami
- Typy generyczne
- Kierunek – programowanie funkcyjne
- Można używać kodu JavaScript w TypeScript -> Typings
- itd 😊





Angular, TypeScript, NPM, angular-cli

Czyli krótki przegląd narzędzi

Co potrzebujemy, żeby zacząć pisać

- **Angular** – framework *client-side* typu *SPA*
- **TypeScript** – nasz „nadjęzyk” programowania
- **NPM** – manager paczek dla JavaScript
- **Angular CLI** – czarodziejska „różdżka”
 - Ściąga wszystkie potrzebne paczki za pomocą *NPMa*
 - Zakłada projekt i generuje kod startowy
 - Wspomaga naszą pracę przez generowanie kodu
 - Buduje nasz projekt i kompresuje go
 - Uruchamia prosty serwer WWW, żebyśmy mogli przetestować naszą aplikację



Piszemy!

Aplikacja do robienia notatek

github.com/michalczukm/gy-angular-workshops

Konkursowe zadanie domowe

Zasady

- Czas na wykonanie: **tydzień (wtorek 11.04 włącznie)**
- **Punkt wyjściowy pracy** – branch “workshops-4” na repozytorium <https://github.com/michalczukm/gy-angular-workshops/>
- Oceniamy **ilość** wykonanych zadań i **jakość** wykonania
- Nagroda: **300zł** bon do Saturna
- Wyślij kod na mateusz.turzynski@goyello.com jako link do repozytorium lub .zip

Funkcjonalności do zaimplementowania

- Przycisk **usuwania** notek
- **Due date** i jego wyświetlanie na liście i szczegółach
- Możliwość ustawiania **priorytetów** i ich wyświetlenie na liście i szczegółach
- **Filtrowanie**
 - po tytule
 - po *due date*
 - po priorytetach
- **wyróżnij** notatki
 - z *due date* na jutro
 - z najwyższym priorytetem
- przycisk **cancel** przy dodawaniu notatki – powinien wrócić do wyświetlania szczegółów
- **edycja** notatki

Dziękujemy z uwagę

- Kod aplikacji:
<https://github.com/michalczukm/gy-angular-workshops>
- Tutorial TypeScript:
<https://www.typescriptlang.org/docs/tutorial.html>
- Tutorial Angular: <https://angular.io/docs/ts/latest/tutorial/>
- Angular-CLI: <https://github.com/angular/angular-cli#usage>

Kontakt do nas:

mateusz.turzyński@goyello.com

michal.michalczuk@goyello.com