

Testy E2E - czemu nie uzywac angularowych

Mittwoch, 31. Juli 2019 09:44

E2e - smoke testy, nie pisze sie w podst. Fazie projektu

Piramida testowania

Ng e2e sie nie uzywa za bardzo, robi sie to inaczej, tak zeby np testowac tez backend

Pisze sie to raczej z zewnatrz, np w Javie - w capie sa do teamy ktore sie tym specjalizuja (selenium java)

Wersjonowanie node modules

Mittwoch, 31. Juli 2019 10:31

Wersjonowanie

X.Y.Z

Zmiana na X - nie musi być kompatybilna wstecznie

Y - kompatybilna powinna być

Z - bugfixy tylko

Dalej

Mittwoch, 31. Juli 2019 10:32

DOM - pamięciowa reprezentacja dokumentu HTML (drzewo obiektów w JS)

DOMContentLoaded - Event po przetworzeniu HTML'a przez przeglądarki

Polyfile - pliki jsowe które wyrównują platforme w przeglądarkach typu IE 11 xD, żeby dało się używać konstrukcji językowych

Wzorce projektowe

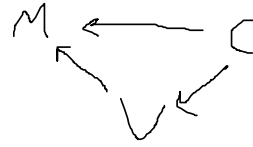
Mittwoch, 31. Juli 2019 11:16

MVVM - Model - View - ViewModel

V
<html>

VM

M



"Clean Architecture" książka
Autora "Clean Code"

MVC nie występuje "w przyrodzie"

Model jest modelem anemicznym - jest pozbawiony zachowań. Przychodzi z serwera. Sam stan. (DTO) - bez zachowania

ViewModel - model pod nasz widok. Stan + zachowanie

Serwer -> Model -> ViewModel -> Operacje na froncie -> ViewModel -> Model -> Serwer

Flux / Redux

Decoupling - ceną jest utracenie trackingu (np w event sourcingu w mikroserwisach)

Smart Componenty (Containers) i Dumb Componenty (Presentational components)

Model - definicja kontraktu między frontem a backendem. Klasy są niepotrzebne. Bo model jest bez zachowań - sam kontrakt.

Książka: JavaScript - The good parts

Bindingi

Mittwoch, 31. Juli 2019 12:11

Interpolacja - zamiana tego co mamy w modelu na stringa {{ }}

Do eventów (click)

Do klasy [class.table-active]

Do atrybutów [attr.aria-label]

Do propertisów [book]

HTML ma wyrażać kontent!

DOM:

atrybuty, propertisy i eventy

Random

Mittwoch, 31. Juli 2019 12:38

```
@import "~bootstrap/scss/bootstrap" // Tylda oznacza katalog node_modules
```

W bootstrapie _reboot.scss resetuje CSSy. Tak sie robi zawsze jak sie robi swoje stylowanie zeby nadpisac rozne rzeczy z przeglądarki.

Zone.js biblioteka z darta. Otoczenie kodu zoną, można interceptować wszystkie wywołania asynchroniczne itp.

Po zaznaczeniu elementu w Chromie jest on dostpeny w konsoli pod zmienna \$0

Na inlinowych elementach nie da sie ustawiac marginesow wielkosci itp. To sa jak literki

O marginesach powinien decydowac kontener (parent) obiektu a nie sam obiekt. (HTML)

JS:
Truthy: not falsy
Falsy: false, 0, "", undefined, null, NaN

Dostęp do view z Viewmodelu (komponentu):

```
@ViewChild()
```

#bookForm to jest template variable - dostep w danym template albo z viewmodel za pomoca view child

Lepiej unikać magii w template - łatwiej testować komponenty. Np template variable

```
const formElement = $event.target as HTMLFormElement;  
const author = (authorElement && authorElement.value) || '';
```

```
console.log({  
  ...this.book,  
  author,  
  title  
});
```

Spread operator - shallow copy

Nazwy eventow click, submit itp więc bookChange a nie bookChanged! Naturalnie

Requeesty HTTP mozna cancelowac (Fetch API)

```
return foundBook ? of(foundBook) : throwError(`No book with id: ${id} found`);
```

Nie renerduje sie formularzy bez danych jak sie na nie czeka

Typizowanie z Pick i Exclude - jak mozna wyslac cos bez jakiegos parametru

```
export interface Book {  
  id: number;  
  author: string;  
  title: string;  
};  
export type BookProperties = Pick<Book, Exclude<keyof Book, 'id'>>;
```

Elementy pozycjonuje sie w parent komponencie

```
ngOnInit(): void {  
  this.book$ =  
    this.bookId != null ?  
    this.bookService.findOne(this.bookId)  
    .pipe(  
      catchError((err) => {  
        return new  
        Observable(subscriber  
=> {  
          this.router.navigate(  
            ['/book']).then(() =>  
              subscriber.error(err)  
            );  
        })  
      })  
    )  
    : throwError(err);  
  }  
}
```

.forRoot() - dodatkowo importuje serwis, zwraca
ModuleWithProviders)

=== zawsze !==

Wyjątek: Chyba ze sprawdz sie czy wartosc nie jest nullem
ani undefined to wtedy !=

"Dyskusyjny design"

Testy

Mittwoch, 31. Juli 2019 14:04

Komponentow:

(class) i (DOM)

Class to zwykle jednostkowe klasy komponentu

DOM potrzebuja infra angulara. (TestBed udostepnia infrastrukture)

ng test --code-coverage

```
it('creates a component', () => {  
  return TestBed.configureTestingModule({  
    declarations: [BookDetailsComponent]  
  }).compileComponents().then(() => {  
    const fixture =  
      TestBed.createComponent(BookDetailsComponent);  
    expect(fixture.componentInstance).toBeDefined();  
  })  
});
```


Service Providers

Donnerstag, 1. August 2019 09:26

DI działa na zasadzie drzewa, można provide'ować serwisy na poziomie modułów i komponentów. Angular szuka providera dla komponentu idąc w górę drzewa - bierze pierwszy.

Jesli interfejs i 2 implementacje serwisu - InjectionToken

Konwencja forRoot - Jak sie robi bibliotekę reużywalną, którą ktoś będzie importować

Export module pattern: (forRoot()). Aby można było exportować też providerów. Domyslnie importowane są tylko komponenty pipe i dyrektywy

```
Export class BookModule {
  static forRoot(): ModuleWithProviders {
    return {
      ngModule: BookModule,
      providers: [BookService]
    }
  }
}
```

RxJS

Donnerstag, 1. August 2019 10:26

```
findAll(): Observable<Book[]> {  
  return new Observable<Book[]>(subscriber => {  
    // Ten callback jest wywoływany od razu w momencie subskrypcji - synchronicznie  
    setTimeout(() => {  
      subscriber.next(this.books)  
      subscriber.complete();  
    }, 1000);  
    // return {  
    // unsubscribe(): void {  
    // console.log('unsubscribing..');  
    // }  
    // }  
  });  
}
```

```
<form #bookForm (submit)="notifyOnBookUpdate($event)"  
*ngIf="book$ | async as book">
```

ngIf ten wyżej wystawia jako templateVariable book jak będzie już.

Routing

Donnerstag, 1. August 2019 11:18

Zawsze wejście na stronę domową robi się:

```
{  
  path: '',  
  pathMatch: 'full',  
  redirectTo: 'books'  
},
```

Router może reużywać komponenty jak ma możliwość - np jak zmieni się ID w rucie a komponent zostaje ten sam. Wtedy powinno się subskrybować na observable `ActivatedRoute` i być na to gotowym.

Jak jest przejście z dialogu na ten sam dialog trzeba o to zadbać. I nie używać snapshota.

Struktura aplikacji

Donnerstag, 1. August 2019 11:35

SharedModule

I

FeatureModules

SharedModule np exportuje
CommonModule

SharedModule

SharedModule is a conventional name for an [NgModule](#) with the components, directives, and pipes that you use everywhere in your app. This module should consist entirely of declarations, most of them exported.

The SharedModule may re-export other widget modules, such as [CommonModule](#), [FormsModule](#), and NgModules with the UI controls that you use most widely.

The SharedModule should not have providers for reasons [explained previously](#). Nor should any of its imported or re-exported modules have providers.

Import the SharedModule in your *feature* modules, both those loaded when the app starts and those you lazy load later.

From <https://angular.io/guide/ngmodule-faq#what-kinds-of-modules-should-i-have-and-how-should-i-use-them>

Zagnieżdzenie komponentów w innych komponentach w strukturze plików jeśli mają sens tylko w danym kontekście.

Podział na katalogi:

Shared > topic > technical
(componenty,serwisy itp)

Feature > technical (components,
services)

Resolvery

Donnerstag, 1. August 2019 14:32

Gdy komponent (dialog) potrzebuje danych zanim zostanie wyrenderowany.

Specyficzny dla komponentu może być umieszczony w komponencie - np `BookDetailsResolver` `book-details.resolver.ts`

Resolver jest wykonywany zawsze, nawet jak przechodzi się z tego samego komponentu na ten sam (router reużywa)

Router puszcza eventy jak jest `ResolveStart` i `ResolveEnd` - można się zapisać i pokazać spinnera

Forms

Donnerstag, 1. August 2019 16:13

```
// Pattern! Przekształcenie na BookFormModel a następnie patchValue - ustawienie formularza.  
interface BookFormModel {  
}
```

```
.ng-invalid.ng-touched {  
@extend .is-invalid;  
}
```

Validator też można umieścić na poziomie grupy

```
This.bookForm.setValidators()
```

Są też asynchroniczne validatory, podawane jako 3 argument w FormControl

Komponent <add-validation>, na zewnątrz inputów, który załatwia sprawę.

```
#bookFormDirective="ngForm" [class.ng-submitted]="bookFormDirective.submitted">
```

To wyżej aby button submit był aktywny, ale w przypadku wciśnięcia żeby pokazywały się błędy walidacji - dobra praktyka.

Testy

Freitag, 2. August 2019 09:04

```
it('async', fakeAsync(() => {  
  setTimeout(() => {  
    expect(true).toBe(true);  
  }, 6000);  
  tick(6000); // symuluje czekanie 6 sekund w fakeAsync zonie  
}));
```

Przesyłanie dat - String w formacie ISO

Na backendzie konfiguracja Jacksona aby zwracał ten format

Moment.js -> Luxon

```
return this.http.get<BookTo[]>('api/books').pipe(transformToDates<Book>);
```

IE

Freitag, 2. August 2019 15:55

Browserslist - usuniecie "not" przed IE

I tyle :O