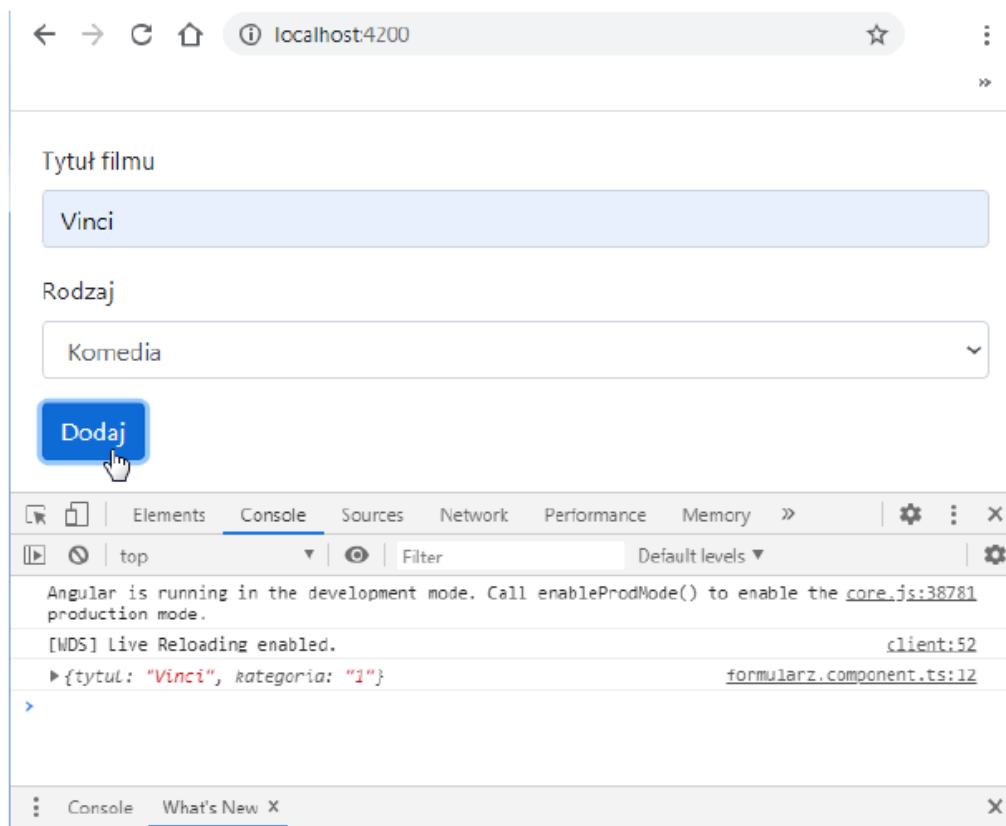
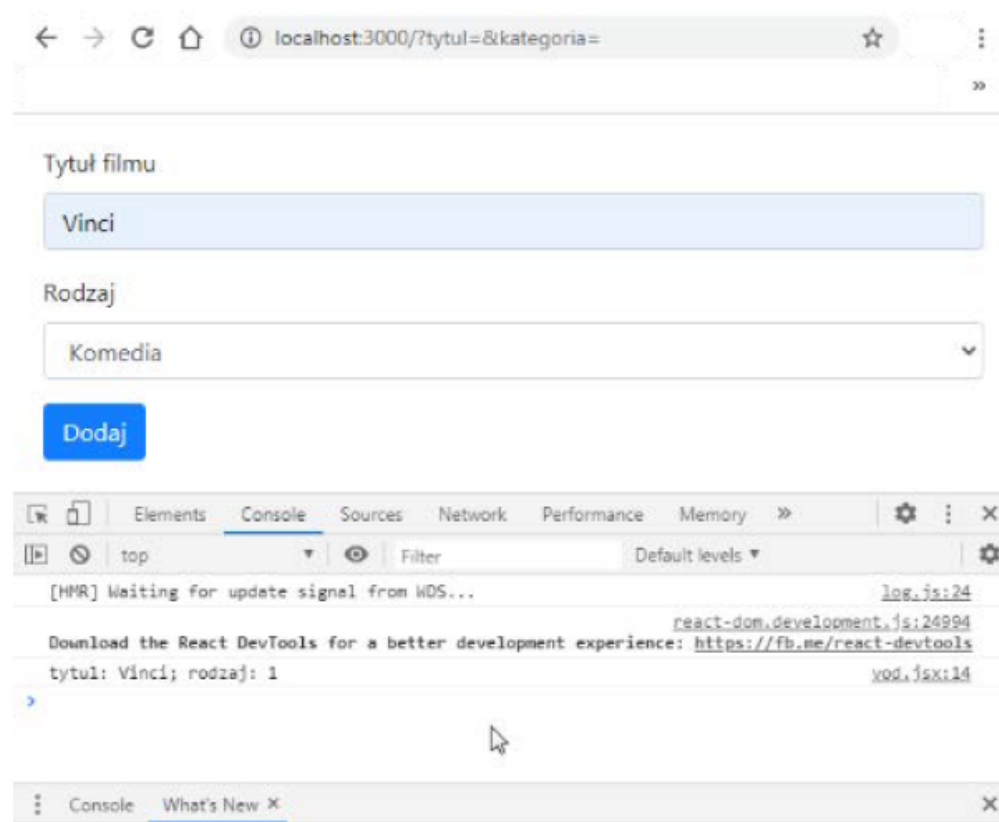


film



Obraz 1a. Formularz wykonany we frameworku Angular. Wybrano przycisk Dodaj



Obraz 1b. Formularz wykonany z biblioteką React.js. Wybrano przycisk Dodaj

film

Tytuł filmu

Vinci

Rodzaj

Komedia

Obyczajowy

Sensacyjny

Horror

lista rozwijana



film

należy zapoznać się z informacjami
na końcu arkusza !!!

Angular

To use ngModel in ngForm add: `import { FormsModule } from '@angular/forms';` in `app.module.ts` file. Add `FormsModule` to imports table

To use Bootstrap add to `styles.css`: `@import '~bootstrap/dist/css/bootstrap.css';`

React.js

To use Bootstrap add: `import 'bootstrap/dist/css/bootstrap.css';`

Bootstrap Forms (Źródło <https://getbootstrap.com/docs/4.0/components/forms/>)

Be sure to use an appropriate type attribute on all inputs (e.g., `email` for email address or `number` for numerical information) to take advantage of newer input controls like email verification, number selection, and more.

Textual form controls—like `<input>`s, `<select>`s, and `<textarea>`s—are styled with the `.form-control` class. Included are styles for general appearance, focus state, sizing, and more.

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" />
  </div>
  <div class="form-group">
    ...
  </div>
</form>
```

Important! In React render method use `className` instead of `class`; `htmlFor` instead of `for`.

Bootstrap buttons (Źródło <https://getbootstrap.com/docs/4.0/components/buttons/>)

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control. The `btn` classes are designed to be used with the `<button>` element. Add modifier classes such as `btn-primary`, `btn-secondary`, `btn-success` and more to `btn` class in order to add background colors

film

```
npm install bootstrap --save
```

instalacja biblioteki bootstrap (przed egzaminem)

```
ng g c formularz
```

tworzenie nowego komponentu formularz

```

  app
  - formularz
    # formularz.component.css
    <> formularz.component.html
    TS formularz.component.spec.ts
    TS formularz.component.ts
  # app.component.css
  <> app.component.html
  TS app.component.spec.ts
  TS app.component.ts
  TS app.config.server.ts
  TS app.config.ts
```

film > src > app > formularz > <> formularz.component.html > ...

Go to component

```
1 <form>
2   <div class="form-group">
3     <label for="nazwaFilmu">Tytuł filmu</label>
4     <!-- musi być atrybut name!!!!-->
5     <input type="text" class="form-control" id="nazwaFilmu" name="tytulFilmu" [(ngModel)]="tytul">
6   </div>
7   <div class="form-group">
8     <label for="rodzajFilmu">Rodzaj</label>
9     <select id="rodzajFilmu" class="form-control" name="kategoriaFilmu" [(ngModel)]="kategoria">
10      <option></option>
11      <option value="1">Komedia</option>
12      <option value="2">Obyczajowy</option>
13      <option value="3">Sensacyjny</option>
14      <option value="4">Horror</option>
15    </select>
16  </div>
17  <div class="form-group">
18    <button class="btn btn-primary" type="button" (click)="dodaj()">Dodaj</button>
19  </div>
20 </form>
```

film > src > app > formularz > TS formularz.component.ts > ...

```
1  ∨ import { Component } from '@angular/core';
2  import { FormsModule } from '@angular/forms';
3
4  ∨ @Component({
5    selector: 'app-formularz',
6    standalone: true,
7    imports: [FormsModule],
8    templateUrl: './formularz.component.html',
9    styleUrls: ['./formularz.component.css']
10 })
11 ∨ export class FormularzComponent {
12   tytul = "";
13   kategoria = "";
14
15   ∨ dodaj(){
16   ∨     let info = {
17     tytul: this.tytul,
18     kategoria: this.kategoria
19   }
20   //console.log(`${tytul: "${this.tytul}", kategoria: "${this.kategoria}"}`)
21   console.log(info);
22   }
23 }
```

TS app.component.ts M X

src > app > TS app.component.ts > ...

```
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3  import { FormularzComponent } from '../formularz/formularz.component';
4
5  @Component({
6    selector: 'app-root',
7    standalone: true,
8    imports: [RouterOutlet, FormularzComponent],
9    templateUrl: './app.component.html',
10   styleUrls: ['./app.component.css']
11 })
12 export class AppComponent {
13   title = 'film3';
14 }
```



film

```
<> app.component.html M X  
film > src > app > <> app.component.html > app-formularz  
Go to component  
1 <app-formularz></app-formularz>
```

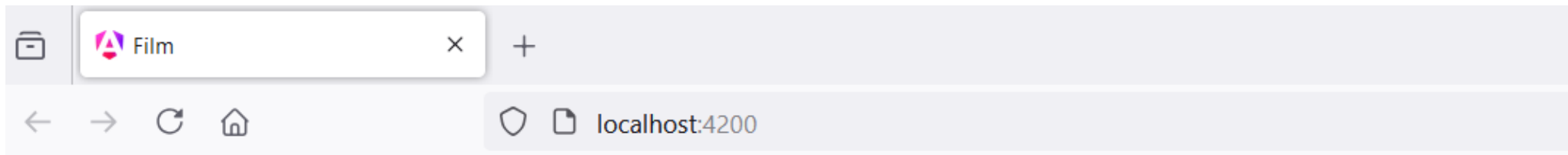
```
# styles.css M X  
film > src > # styles.css > ...  
1 /* You can add global styles to this file, and also import other style files */  
2 @import "../node_modules/bootstrap/dist/css/bootstrap.css";  
3 body { padding: 20px; }  
4
```

dokumentacja

Listing 1. Wzór dokumentacji klasy

```
*****  
nazwa klasy: <tu wstaw nazwę klasy>  
pola:        <nazwa pola> - <co przechowuje>  
             <nazwa pola> - <co przechowuje>  
metody:      <nazwa metody>, <co zwraca> - <opis>  
informacje: <opis>  
autor:       <numer zdającego>  
*****
```


film



Tytuł filmu

Vinci

Rodzaj

Komedia

Dodaj

Firefox

Inspektor **Konsola** Debugger Sieć Edytor stylów Wydajność Pamięć Dane Dostępność Wątki

Filtruj zawartość

[vite] connecting...

❗ Obiekt Components jest przestarzały. Zostanie wkrótce usunięty.

[vite] connected.

Angular is running in development mode.

Angular hydrated 2 component(s) and 24 node(s), 0 component(s) were skipped. Learn more at <https://angular.dev/guide/hydration>.

▶ Object { tytuł: "Vinci", kategoria: "1" }

>>

film

Chrome

The screenshot shows a web browser window with a single tab titled "Film". The address bar displays "localhost:4200". The browser's developer tools are open, showing the "Console" tab. The main content area displays a form with two input fields: "Tytuł filmu" containing "Vinci" and "Rodzaj" containing "Komedia". A blue button labeled "Dodaj" is located below the form. The console shows the following messages:

```
Angular is running in development mode.  
Angular hydrated 2 component(s) and 24 node(s),  
more at https://angular.dev/guide/hydration.  
▶ {tytul: 'Vinci', kategoria: '1'}
```

Red circles highlight the "Dodaj" button in the form and the log entry in the console.

film - testy

ng test

— polecenie uruchamiające testy

Przeglądarką Chrome steruje zautomatyzowane oprogramowanie testowe.

Karma v 6.4.4 - connected; test: complete; DEBUG

Chrome 129.0.0.0 (Windows 10) is idle

Jasmine 4.6.1 Options

4 specs, 1 failure, randomized with seed 42756 finished in 0.162s

[Spec List](#) | [Failures](#)

AppComponent > should render title

Expected undefined to contain 'Hello, film'.

```
at <Jasmine>
at useContext.apply (http://localhost:9876/_karma_webpack_/webpack:/src/app/app.component.spec.ts:27:55)
at _ZoneDelegate.invoke (http://localhost:9876/_karma_webpack_/webpack:/node_modules/zone.js/fesm2015/zone.js:369:28)
at ProxyZoneSpec.onInvoke (http://localhost:9876/_karma_webpack_/webpack:/node_modules/zone.js/fesm2015/zone-testing.js:2081:39)
at _ZoneDelegate.invoke (http://localhost:9876/_karma_webpack_/webpack:/node_modules/zone.js/fesm2015/zone.js:368:34)
```

film - testy

TS app.component.spec.ts X

film > src > app > TS app.component.spec.ts > ...

```
4   describe('AppComponent', () => {
15     });
16
17     it(`should have the 'film' title`, () => {
18       const fixture = TestBed.createComponent(AppComponent);
19       const app = fixture.componentInstance;
20       expect(app.title).toEqual('film');
21     });
22
23     it('should render title', () => {
24       const fixture = TestBed.createComponent(AppComponent);
25       fixture.detectChanges();
26       const compiled = fixture.nativeElement as HTMLElement;
27       expect(compiled.querySelector('h1')?.textContent).toContain('Hello, film');
28     });
29   });
30
```

film - testy

```
<> app.component.html M X
film > src > app > <> app.component.html > app-formularz
  Go to component
1 <h1>Hello, film</h1>
2 <app-formularz></app-formularz>
```

The screenshot shows a web browser window with the Karma test runner interface. The browser's address bar shows the URL `localhost:9876/?id=75951214#`. A notification at the top states "Przełglądarką Chrome steruje zautomatyzowane oprogramowanie testowe." Below this, a green banner displays the message "Karma v 6.4.4 - connected; test: complete;" with a "DEBUG" button on the right. The main content area shows "Chrome 129.0.0.0 (Windows 10) is idle". The Jasmine test runner logo and version "4.6.1" are visible, along with an "Options" button. A dark green status bar reports "4 specs, 0 failures, randomized with seed 41206" and "finished in 0.076s". The test results list includes:

- AppComponent
 - should create the app
 - should render title
 - should have the 'film' title
- FormularzComponent
 - should create

film - testy

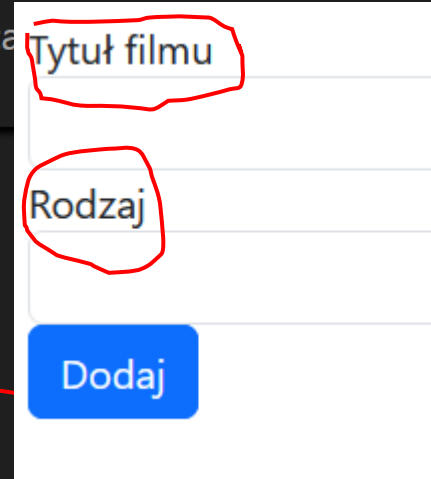
```
✓ Browser application bundle generation complete.  
Chrome 129.0.0.0 (Windows 10): Executed 4 of 4 SUCCESS (0.078 secs / 0.056 secs)  
TOTAL: 4 SUCCESS  
█
```

FormularzComponent test

TS formularz.component.spec.ts U X

src > app > formularz > TS formularz.component.spec.ts > describe('FormularzComponent') ca

```
5 describe('FormularzComponent', () => {  
19  
20   it('should create', () => {  
21     expect(component).toBeTruthy();  
22   });  
23  
24   it('should have Tytuł filmu', () => {  
25     const compiled = fixture.nativeElement as HTMLElement;  
26     expect(compiled.querySelectorAll('label')[0].textContent).toContain('Tytuł filmu');  
27   });  
28  
29   it('should have Rodzaj', () => {  
30     const compiled = fixture.nativeElement as HTMLElement;  
31     expect(compiled.querySelectorAll('label')[1].textContent).toContain('Rodzaj');  
32   });  
33 });  
34
```



The screenshot shows a web form with two input fields and a button. The first input field is labeled "Tytuł filmu" and the second is labeled "Rodzaj". A blue button labeled "Dodaj" is positioned below the second input field. Red circles highlight the labels "Tytuł filmu" and "Rodzaj". A red arrow points from the first test case in the code to the "Tytuł filmu" label, and another red arrow points from the second test case to the "Rodzaj" label. A large red circle encompasses the entire code block and the form screenshot.

FormularzComponent test

Karma v 6.4.4 - connected; test: complete;

Firefox 131.0 (Windows 10) is idle

 Jasmine 4.6.1

• • • • •

6 specs, 0 failures, randomized with seed 58148

FormularzComponent

- should have Rodzaj
- should create
- should have Tytuł filmu

AppComponent

- should have the 'film' title
- should render title
- should create the app

Safe Navigation Operator

Template Syntax with Safe Navigation Operator:

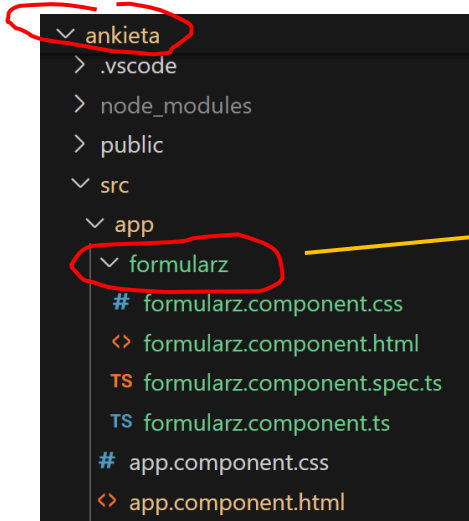
In Angular templates, the question mark can be used as part of the "safe navigation operator" (`?.`). It works similarly to optional chaining in TypeScript, allowing you to safely access properties on potentially undefined or null objects:

```
<div>{{ user?.name }}</div> <!-- Will not throw an error if user is null or undefined -->
```

zadanie

napisać ankietę w komponencie *formularz*

dane są uzupełniane na bieżąco



Ankieta

Mirosław	imię
Kowalski	nazwisko
kowal@op.sad.pl	e-mail

pleć

mężczyzna kobieta

zainteresowania

sport film elektronika informatyka

szkoła

Technikum

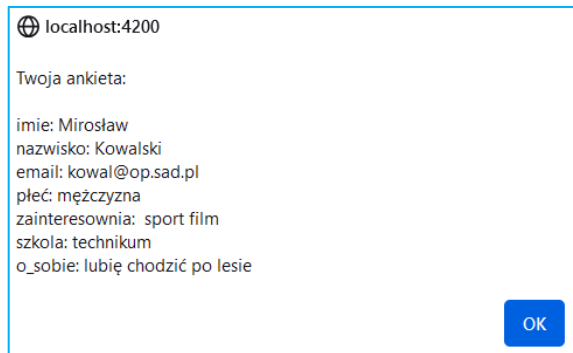
krótko o sobie

lubię chodzić po lesie

submit reset

imię: Mirosław
nazwisko: Kowalski
email: kowal@op.sad.pl
pleć: mezczyzna
sport: true
film: true
elektronika:
informatyka:
szkola: technikum
o_sobie: lubię chodzić po lesie

alert



zadanie - dokumentacja

Wykonaj dokumentację do aplikacji



Listing 1. Wzór dokumentacji klasy

```
*****  
nazwa klasy: <tu wstaw nazwę klasy>  
pola:        <nazwa pola> - <co przechowuje>  
             <nazwa pola> - <co przechowuje>  
metody:      <nazwa metody>, <co zwraca> - <opis>  
informacje: <opis>  
autor:       <numer zdającego>  
*****
```

zadanie - test

przetestować w
komponencie
formularz
występowanie
napisów na
stronie

Ankieta

Mirosław	imię
Kowalski	nazwisko
kowal@op.sad.pl	e-mail

pleć

mężczyzna kobieta

zainteresowania

sport film elektronika **informatyka**

szkoła

Technikum

krótko o sobie

lubię chodzić po lesie

submit reset

imię: Mirosław
nazwisko: Kowalski
email: kowal@op.sad.pl
plec: mezczyzna
sport: true
film: true
elektronika:
informatyka:
szkola: technikum
o_sobie: lubię chodzić po lesie

Karma v 6.4.4 - connected; test: complete;

Chrome 130.0.0.0 (Windows 10) is idle

Jasmine 4.6.1

6 specs, 0 failures, randomized with seed 48111

AppComponent

- should create the app
- should have the 'ankieta' title

FormularzComponent

- should create
- should have h1 Ankieta
- should have h3 pleć
- should have label informatyka