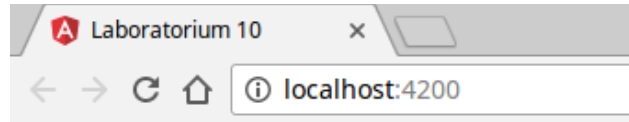


# Laboratorium 10

## Komponenty i usługi w Angular 7

### Zadanie 1 – uruchomienie prawie pustego projektu

Pod linkiem [lab10app.zip](http://lab10app.zip) dostępny jest niemal pusty projekt wyświetlający tylko tytuł aplikacji. Projekt trzeba pobrać, rozpakować, doinstalować pakiety, skompilować i otworzyć w przeglądarce:

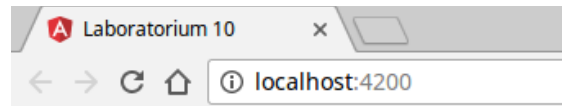


## Studenci z PCz

W zasadzie niestandardowe są tylko pliki `mock-students.ts` (tablica z listą studentów udająca dane z bazy) oraz `student.ts` (klasa `Student` wykorzystywana w całej aplikacji).

### Zadanie 2 – zrobienie prostej listy studentów

Proszę utworzyć za pomocą narzędzia Angular-CLI komponent `students` (po utworzeniu będzie się nazywał `StudentsComponent`) wyświetlający listę studentów z ich numerami indeksów. Proszę tak przygotować komponent, aby po kliknięciu na element z listy, wybrany student się zaznaczał, np. pogrubionym niebieskim kolorem:



## Studenci z PCz

Lista studentów:

- Marek Głowala (37014)
- Martyna Janicka (13840)
- Jan Kowalski (11234)
- **Wojciech Myśliwiec (37014)**
- Halina Siwiecka (28152)
- Tomasz Wroński (26024)

Będzie trzeba wykorzystać:

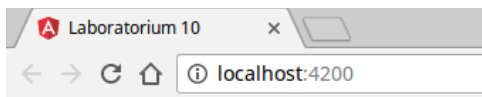
- `ngFor`,
- wiązanie zdarzeń do zaznaczania, np. `(click)="mojaMetoda(zmienna)"`
- wiązanie właściwości do przypisania stylu do elementu, np. `[class.selected]="aa===3"`

### Zadanie 3 – lista studentów wzbogacona o wyświetlanie szczegółów

Można od razu zrobić część a i b tego zadania jeśli ktoś potrafi.

#### a) Wyświetlanie szczegółów studenta po jego wybraniu

Proszę do komponentu `students` dodać wyświetlanie szczegółowych informacji o tym studentcie, który zostanie wybrany na liście. Jeśli nie został wybrany żaden student, to szczegółowe informacje nie powinny się pojawiać (dyrektywa `ngIf`), można ewentualnie wyświetlić odpowiednią informację, np. „nie wybrano studenta”. Użytkownik powinien mieć możliwość zmiany nazwy studenta oraz numeru jego indeksu (dyrektywa `ngModel`). W szczegółach proszę wyświetlić również wyliczoną średnią ocen i liczbę przedmiotów, które student posiada. Przykładowy wynik poniżej.

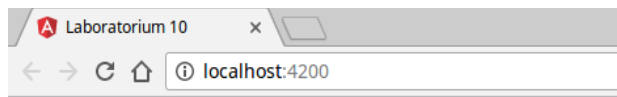


## Studenci z PCz

Lista studentów:

- Marek Głowala (37014)
- Martyna Janicka (13840)
- Jan Kowalski (11234)
- Wojciech Myśliwiecki (37014)
- Halina Siwiecka (28152)
- Tomasz Wroński (26024)

-- nie wybrano studenta --



## Studenci z PCz

Lista studentów:

- Marek Głowala (37014)
- Martyna Janicka (13840)
- Jan Kowalski (11234)
- **Wojciech Myśliwiecki (37015)**
- Halina Siwiecka (28152)
- Tomasz Wroński (26024)

Informacje o: WOJCIECH MYŚLIWIECKI

Imię i nazwisko:  Indeks:   
Średnia ocen: 4.85, liczba przedmiotów: 10

### b) Przeniesienie szczegółowych danych do osobnego komponentu

Proszę utworzyć komponent `student-detail`, który będzie wyświetlał szczegółowe informacje o wybranym studencie. Wystarczy przenieść szablon do nowo-utworzonego komponentu i wykonać kilka drobnych korekt. Należy też wykorzystać dekorator `@Input` oraz wiązanie właściwości, które pozwoli na przekazanie informacji z komponentu `students` do `student-detail`. Wygląd zasadniczo nie powinien ulec zmianie. Można pobawić się trochę zawartością pliku `student-detail.component.css`, np.:

- **Tomasz Wroński (26024)**

Informacje o: **TOMASZ WROŃSKI**

Imię i nazwisko:  Indeks:   
Średnia ocen: **4.1**, liczba przedmiotów: **10**

### Zadanie 4 – wczytywanie danych z usługi zamiast na stałe z pliku

Korzystając z Angular-CLI proszę utworzyć sobie usługę `student` (plik `student.service.ts`). Do usługi dodajemy metodę `getStudents()`, która zwróci całą zawartość tablicy `STUDENTS` z pliku `mock-students.ts`. Usługę należy wykorzystać w komponencie `StudentsComponent`. Aby to zrobić należy:

- zaimportować usługę `StudentService`,
- zastąpić w klasie `StudentsComponent` pole `students = STUDENTS`; pustą tablicą czyli `students: Student[]`;
- utworzyć w konstruktorze prywatne pole będące typu `StudentService` – dzięki temu mechanizm wstrzykiwania zależności udostępni usługę do komponentu, można dodać np. takie pole: `private studentService: StudentService`,
- utworzyć metodę `getStudents()`, która pobierze dane z `StudentService` za pomocą jej metody `getStudents()`, uzyskamy to za pomocą następującego kodu: `this.students = this.studentService.getStudents()`;
- wywołać metodę `getStudents()` komponentu przy jego inicjalizacji – wstawka programowa `ngOnInit`.

Jeśli wszystko jest właściwie, to powinniśmy mieć taki sam efekt jak w zadaniu 3 ale teraz mamy już działającą usługę dostarczaną za pomocą mechanizmu wstrzykiwania zależności.

### Zadanie 5 – dla chętnych, asynchronicznie pobieranie danych z usługi - subscribe

Zobacz: <https://angular.io/tutorial/toh-pt4#observable-data>.

Docelowo w metodzie `getStudents()` naszego komponentu powinniśmy mieć kod:

```
this.studentService.getStudents().subscribe(students => this.students = students);
```

czyli dopisujemy komponent do listy obserwujących zmiany danych w usłudze

`StudentService` – jeśli takie zmiany nastąpią, usługa odczyta dane z bazy, to nasz komponent otrzyma dane i wyświetli je w odpowiednim miejscu w szablonie.

### Zadanie 6 – dodatkowy komponent wyświetlający listę ocen z możliwością ich zmiany

Proszę utworzyć dodatkowy komponent o nazwie `student-grades`, w którym będzie się wyświetlać lista ocen zaznaczonego studenta z możliwością ich zmiany. Używamy dyrektywy `ngModel` i pola `@Input`. Zmiana oceny powinna od razu spowodować zmianę średniej danego studenta. Jedno z możliwych rozwiązań obok.

### Zadanie 7 – usuwanie przedmiotów

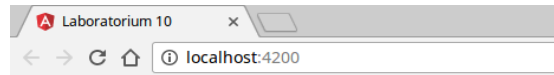
Proszę dodać do listy przedmiotów przyciski pozwalające na usuwanie ich z listy. Kliknięcie przycisku powinno usuwać przedmiot z listy co powinno prowadzić do automatycznej aktualizacji średniej i liczby wpisanych ocen dla danego studenta.

Imię i nazwisko:  Indeks:

Średnia ocen: **4.85**, liczba przedmiotów: **10**

Uzyskane oceny:

<input type="text" value="5"/>	- Tworzenie serwisów Web 2.0	<input type="button" value="USUŃ"/>
<input type="text" value="5"/>	- Eksploracja danych i hurtownie danych	<input type="button" value="USUŃ"/>
<input type="text" value="4.5"/>	- Szkielety tworzenia aplikacji	<input type="button" value="USUŃ"/>
<input type="text" value="5"/>	- Wzorce projektowe	<input type="button" value="USUŃ"/>



## Studenci z PCz

Lista studentów:

- Marek Glowala (37014)
- Martyna Janicka (13840)
- Jan Kowalski (11234)
- Wojciech Myśliwiec (37014)
- **Halina Siwiecka (28152)**
- Tomasz Wroński (26024)

Informacje o: **HALINA SIWIECKA**

Imię i nazwisko:  Indeks:

Średnia ocen: **3.9**, liczba przedmiotów: **10**

Uzyskane oceny:

<input type="text" value="3"/>	- Tworzenie serwisów Web 2.0
<input type="text" value="3"/>	- Eksploracja danych i hurtownie danych
<input type="text" value="4.5"/>	- Szkielety tworzenia aplikacji
<input type="text" value="4"/>	- Wzorce projektowe
<input type="text" value="4.5"/>	- Programowanie aplikacji dla Windows
<input type="text" value="5"/>	- Systemy baz danych
<input type="text" value="4"/>	- Metody dosępu do danych
<input type="text" value="3.5"/>	- Bezpieczeństwo komunikacji elektronicznej
<input type="text" value="4.5"/>	- Technologie zarządzania treścią
<input type="text" value="3"/>	- Aplikacje wielowarstwowe

### Zadanie 8 – dodawanie przedmiotów studentom

Można jeszcze dodać możliwość dodawania przedmiotów studentom. Można to zrobić zarówno w tym komponencie jak i w osobnym.

### Zadanie 9 – ładne ostylowanie tego Bootstrapem i CSSami :)

...