

Serwisy

Zadanie 1.

1. Stwórz projekt o nazwie **AJAX**.
2. Zainstaluj **json-server**:
<https://github.com/typicode/json-server#install>
3. Pobierz plik [db.json](#) i skopiuj go do katalogu **AJAX**.
4. Stwórz serwis **Test**:
ng g s Test
5. Zaimportuj serwis do **app.module.ts** do tablicy **providers**.
6. Stwórz komponent **Testowy**, który będzie zawierać:
 - a. import serwisu **Test** w pliku TS komponentu;
 - b. utworzenie prywatnej zmiennej w konstruktorze odnoszącej się do serwisu;
 - c. formularz:
 - i. pole tekstowe z etykietą **name**;
 - ii. pole select (na razie bez wartości) z etykietą **species**;
 - iii. pole radio z etykietą **gender** i możliwościami: **male, female, n/a**;
 - iv. pole tekstowe z etykietą **homeworld**;
 - v. przycisk zapisujący formularz:
 - w pliku TS należy stworzyć metodę, która będzie zapisywała dane z formularza do postaci jednego obiektu z polami kolejno jak etykiety formularza a wartościami są ciągi znaków;
 - d. tabelę, w której:
 - i. **thead** ma 1 wiersz i komórki kolejno jak etykiety formularza;
 - ii. wiersze w części **tbody** będą generowane przez **ngFor** na tablicy obiektów **characters**, którą należy zadeklarować w pliku TS – na razie tablica jest pusta – na każdy wiersz w części **tbody** będą 4 komórki (w jednym obiekcie są pola kolejno jak etykiety formularza);
 - iii. kiedy w tablicy **characters** nie ma elementów zamiast tabeli należy wyświetlić informację tekstową o braku elementów w tablicy.

7. Uruchom projekt.
8. Uruchom w drugim oknie terminala serwer:
`json-server --watch db.json`

Zadanie 2.

1. W pliku TS serwisu **Test**:
 - a. przygotuj metodę **getData**, która będzie zwracać wszystko z serwera w **characters** jako JSON;
 - b. przygotuj metodę **getSpecies**, która będzie zwracać wszystko z serwera w **species** jako JSON;
 - c. przygotuj metodę **postData**, która będzie wysyłać do serwera nowy element do **characters**.
2. Dla komponentu **Testowy**:
 - a. dopisz metodę, która będzie wypełniać pole *select* formularza wartościami pobranymi z serwera przez metodę z serwisu **getSpecies**;
 - b. dopisz metodę, która pobierze do tablicy **characters** dane z serwera przez metodę z serwisu **getData**;
 - c. metody z ppkt. a i b wywołaj w **ngOnInit()**;
 - d. w metodzie zapisującej formularz do obiektu dopisz instrukcję wysyłającą ten obiekt przez metodę serwisu **postData**;
 - e. w metodzie **ngOnDestroy()** napisz instrukcję usuwającą wszystkie subskrypcje do metod serwisu.
3. Przetestuj działanie aplikacji.

Zadanie 3.

Dorób obsługę usuwania konkretnego elementu z **characters**. Usuwanie odbywa się przy pomocy **id** konkretnego elementu oraz metody HTTP **DELETE**.

Widok w przeglądarce jest na bieżąco odświeżany.

Zadanie dla chętnych (+100% do oceny)

Dorób obsługę edytowania konkretnego elementu z **characters**. Edytowanie odbywa się przy pomocy **id** konkretnego elementu oraz metody HTTP **PUT** (z ciałem wysyłanego elementu jak w **POST**). Widok w przeglądarce jest na bieżąco odświeżany. Do samej edycji wykorzystujemy utworzony wcześniej formularz do dodawania elementów.

*Być może przyda się tutaj rozdzielenie kodu na więcej komponentów, komunikacja pomiędzy równoległymi komponentami, użycie **ngOnChanges** i **SimpleChanges**.*

Po wykonaniu wszystkich zadań cały projekt należy wysłać do repozytorium GitHuba do katalogu ANGULAR-CW03.