

Specyfikacja Projektów 2019/2020

Wybrany projekt aplikacji powinien być wykonany przy pomocy Angular, według osobistych preferencji studenta. Projekt powinien być zgodny z przedstawioną specyfikacją. Ponadto, w sekcji „Walory dodatkowe” wyliczono opcjonalne warunki, których spełnienie podwyższy ocenę końcową za projekt.

Projekt należy wgrać na *GitHub*, a następnie wysłać prowadzącemu e-mail z linkiem.

Wyszukiwarko-przeglądarka artykułów

1. Wejdź na stronę <https://developer.nytimes.com/signup> i zarejestruj się. Na potrzeby rejestracji możesz podać tymczasowy e-mail. Celem jest uzyskanie klucza dostępu do API.
2. Spróbuj wykonać zapytanie HTTP typu GET pod adres podany poniżej. Pamiętaj, żeby wstawić we wskazane miejsce swój klucz API uzyskany w punkcie pierwszym: https://developer.nytimes.com/proxy/https/api.nytimes.com/svc/search/v2/articlesearch.json?api-key=KLUCZ_DO_API&q=president&fl=headline,web_url&begin_date=20171201&end_date=20180430&sort=newest.
3. Zauważ, że parametr *q* określa wyszukiwaną frazę, a parametr *fl* zawęża liczbę pól zawartych w odpowiedzi serwera do, np. *headline* i *web_url*. Parametr *begin_date* i *end_date* określa datę od-do publikacji szukanego artykułu w postaci *RRRRMMDD*. Parametr *sort* przyjmuje zaledwie dwie wartości: *newest* i *oldest*.
4. Pełna dokumentacja wspomnianego API *New York Times* znajduje się pod adresem developer.nytimes.com/article_search_v2.json#/Documentation/GET/articlesearch.json.
5. Napisz aplikację, która umożliwi użytkownikowi wyszukanie artykułu/-ów według frazy i daty od-do, a także wybór sortowania: od najnowszych lub od najstarszych. Wyniki wyszukiwania zaprezentuj w postaci listy nagłówek (*headline*), z których każdy będzie linkował do strony z właściwym artykułem (*web_url*).

Blog

1. Przygotuj plik *database.json* według poniższego przykładu:

```
{
  posts: [
    { id: 1, title: 'Post numer 1', text: 'Lorem ipsum...' },
    { id: 2, title: 'Post numer 2', text: 'Lorem ipsum...' }
  ]
}
```

2. *JSON Server* zapewni dostęp do danych wprowadzonych w *database.json* poprzez RESTowe API, z którego skorzystasz przy pomocy *fetch*, biblioteki *Axios*, serwisu *HttpClient* (Angular), albo innego wybranego klienta HTTP, .
3. Dokumentacja API *JSON Server*: <https://www.npmjs.com/package/json-server>.
4. Napisz aplikację typu blog, która umożliwi użytkownikowi dodawanie (formularz dodawania nowego posta), przeglądanie i usuwanie postów (lista postów). Pojedynczy post powinien składać się z tytułu, treści oraz pola id, które będzie wykorzystywane w zapytaniach HTTP.
5. Użytkownik powinien móc wyszukiwać posty po tytułach oraz opcjonalnie odfiltrowywać posty po tagach, które są im przypisane.

Walory dodatkowe

1. Autor aplikacji dołączył instrukcję dotyczącą jej uruchomienia.
2. Nazwy zmiennych, funkcji, klas, itp. są logiczne i zrozumiałe.
3. UI aplikacji jest schludny, przemyślany i funkcjonalny.
4. Zastosowano animacje, które polepszają komfort korzystania z aplikacji.
5. Aplikacja składa się z kilku widoków obsługiwanych przez Angular router.
6. Routing jest skonfigurowany w taki sposób, że przejście na nieistniejący widok powoduje wyświetlenie strony typu „Przepraszamy, ta strona nie istnieje.”
7. Do zarządzania stanem aplikacji zastosowano bibliotekę Redux lub NgRx.