

Programowanie stron internetowych

Lab 8 – Ajax

1. Proszę o utworzenie strony internetowej, która będzie korzystać z biblioteki *jQuery* i frameworku *Bootstrap*.
2. Z racji, iż ćwiczenia wykonywane w ramach tego przedmiotu traktują o technologiach używanych po stronie klienta, a nie serwera, jako źródło danych wykorzystamy plik *json* znajdujący się pod następującym adresem: <https://raw.githubusercontent.com/wedeploy-examples/supermarket-web-example/master/products.json>
3. Proszę stworzyć przycisk służący do pobierania danych i nadanie mu odpowiedniego stylu za pomocą frameworku *Bootstrap*.
4. Następnie proszę o pobranie danych *json* po kliknięciu przycisku. Obsługę kliknięcia proszę zrealizować za pomocą *jQuery*. Do pobrania danych proszę wykorzystać następującą funkcję *fetch* oraz dokonać obsługi *promise* za pomocą metody *then*:

```
fetch('https://raw.githubusercontent.com/wedeploy-examples/supermarket-web-example/master/products.json')
  .then(
    response => {
      if (response.status !== 200) {
        console.log('Wystąpił problem. Kod statusu: ' + response.status);
        return;
      }
      response.json().then(data => {
        console.log(data);
      });
    }
  )
  .catch(err => {
    console.error(err);
  });
```

5. Następnie proszę o implementację metody *displayJsonData(data)*, która będzie wyświetlała pobrane dane w formie tabeli utworzonej z wykorzystaniem stylów *Bootstrap*. Informacje o pracy z formatem JSON można znaleźć pod adresem: https://www.w3schools.com/js/js_json.asp
6. Proszę nad tabelką dodać pole tekstowe umożliwiające filtrowanie danych po wybranej kolumnie i zaimplementować obsługę filtrowania. Pomocny artykuł na ten temat można znaleźć pod adresem: <https://www.vinta.com.br/blog/2015/javascript-lambda-and-arrow-functions/>
7. Proszę o dodanie funkcjonalności sortowania elementów w tabeli. Sortowanie ma następować po naciśnięciu na nagłówki kolumny. Pomocna może być tutaj metoda *sort* i tzw. *arrow functions*.
8. Następnie proszę dodać drugi przycisk na stronę i obsłużyć za jego pomocą pobieranie danych z wykorzystaniem funkcji asynchronicznej używającej operatory *await*. Artykuł na ten temat można znaleźć pod adresem: <https://kursjs.pl/kurs/es6/async-await.php>
9. Wykonywanie żądań asynchronicznych jest możliwe również poprzez funkcje *jQuery*, np. *\$.ajax* lub *\$.get*. Proszę o zapoznanie się z nimi i o pobranie danych za pomocą funkcji *\$.get*. W tym wypadku może okazać się konieczne parsowanie pobranych danych za pomocą metody *JSON.parse(data)*.