

## Javascript, AJAX i PHP

### Wstęp

Javascript umożliwia dostęp do drzewa dokumentu HTML, ale może być również zastosowany do wysyłania asynchronicznych zapytań (AJAX) do serwera.

### Przykład

W poniższym przykładzie wykorzystany zostanie Javascript do wczytania dokumentu JSON zapytaniem AJAX, a następnie na podstawie danych w nim zawartych zbudowana zostanie tabela HTML. Dokument JSON będzie dynamicznie generowany przez PHP na podstawie zawartości bazy danych.

### Dokument HTML

W pliku HTML zawarta jest tabela (wraz z nagłówkiem), która będzie wypełniana danymi pobranymi z serwera. Dodatkowo został stworzony przycisk, po którego kliknięciu zostaną uruchomione funkcje odpowiedzialne za wczytanie danych oraz wypełnienie tabeli danymi. W sekcji HEAD podłączona została biblioteka Bootstrap CSS (<https://getbootstrap.com/>) oraz plik *funkcje.js* zawierający opisywaną funkcjonalność.

### Plik JS

Plik Javascript został podzielony na kilka części.

Na początku znajdująca się w strukturze HTML tabela i domyślnie ukrywana:

```
const ksiazkiElem = document.getElementById('ksiazki')
const samochodyTable = document.getElementById('samochody')
samochodyTable.style.display = 'none'
```

W kolejnym kroku podłączane jest zdarzenie click do przycisku „Pobierz”. W ramach obsługi zdarzenia wysyłane jest zapytanie AJAX, pobierające dokument JSON z serwera. Na otrzymanym obiekcie wykonywana jest pętla *forEach* i dla każdego elementu tablicy uruchamiana jest funkcja *pokazSamochod()*.

```
/**
 * Podłączenie zdarzenia click do przycisku pobierz
 */
document.getElementById('btn-pobierz').addEventListener('click', async () => {
  const resp = await fetch('pobierz.php')
  const json = await resp.json()

  json.forEach(s => pokazSamochod(s))
  samochodyTable.style.display = ''
})
```

## Zajęcia 4 - AJAX

Funkcja *pokazSamochod* () przetwarza element tablicy otrzymanej z serwera i generuje na jego podstawie kod HTML, który następnie jest dołączany do tabeli:

```
/**
 * Wyświetla dane pojedynczego samochodu w tabeli HTML.
 */
function pokazSamochod(samochod) {
    const { id, marka, model, rok, pojemnosc } = samochod

    const wiersz = `
        <tr>
            <td>${id}</td>
            <td>${marka}</td>
            <td>${model}</td>
            <td>${rok}</td>
            <td>${pojemnosc}</td>
            <td><button class="btn btn-link" data-
id="${id}">pokaż szczegóły</button>
        </tr>
    `

    const template = document.createElement('template')
    template.innerHTML = wiersz.trim()
    samochodyTable.querySelector('tbody').appendChild(template.content.firstChi
ld)
}
```

Ostatnim elementem funkcjonalności jest podłączenie zdarzenia *click* do przycisków „Pokaż szczegóły” w każdym wierszu tabeli:

```
/**
 * Podłączenie zdarzenia click dla przycisku pokaż szczegóły
 */
document.body.addEventListener('click', (e) => {
    if (e.target.matches('[data-id]')) {
        const id = e.target.dataset.id

        pokazSzczegoly(id)
    }
})

/**
 * Pokazuje szczegóły samochodu.
 *
 * @param {string} id
 */
function pokazSzczegoly(id) {
    alert('Szczegóły samochodu o ID: ' + id)
}
```

### Plik `pobierz.php`

Plik `pobierz.php` jest uruchamiany przez asynchroniczne zapytanie AJAX. Jego zadaniem jest pobranie danych z tabeli `samochody` w bazie danych, a następnie zserializowanie ich do formatu JSON.

```
<?php
```

```
const DB_HOST = 'localhost';
const DB_NAME = 'test';
const DB_USERNAME = 'root';
const DB_PASSWORD = '';

function polacz(): PDO
{
    return new PDO(
        sprintf('mysql:host=%s;dbname=%s', DB_HOST, DB_NAME),
        DB_USERNAME,
        DB_PASSWORD
    );
}

$pdo = polacz();
$stmt = $pdo->prepare('SELECT * FROM samochody ORDER BY marka, model');
$stmt->execute();
$samochody = $stmt->fetchAll(PDO::FETCH_ASSOC);

header('Content-Type: application/json');
echo json_encode($samochody);
```

### Zadania

1. Proszę ukryć przycisk „Pobierz samochody” po kliknięciu w niego i wygenerowaniu tabeli HTML.
2. Proszę zmodyfikować funkcję `pokazSzczegoly()` tak, by szczegóły samochodu (wszystkie pozostałe kolumny tabeli *samochod*) zostały wyświetlone w prawej kolumnie kontenera (element o ID *szczegoly*)