Dokumentacja projektu
Books Library
Webowe Biblioteki Programistyczne PHP

Damian Kwaśniok, Grupa 1. PI

14 czerwca 2019

1 Część I

1.1 Opis programu

Program (aplikacja) jest to własna mini biblioteka, w której można przechowywać informacje o posiadanych książkach oraz ich dostępności

1.2 Opis działania

Po uruchomieniu aplikacji widoczny jest formularz do wprowadzania nowych książek oraz książki zapisane wcześniej.

Po dodaniu książki można ustawić jej dostępność/niedostępność lub ją usunąć z bazy.

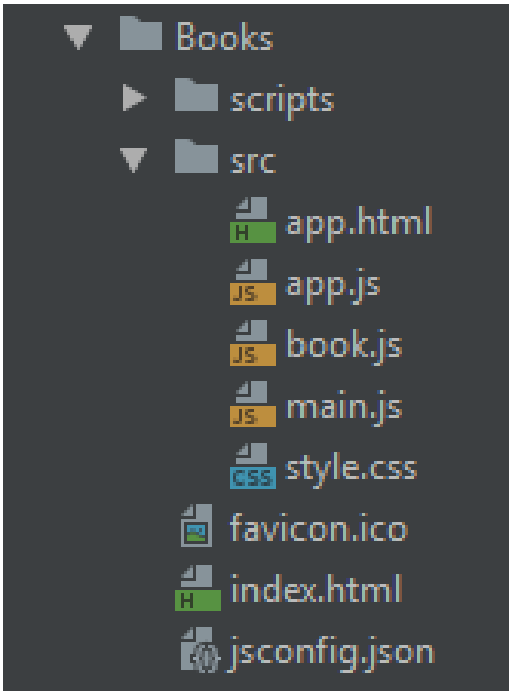
1.3 Instrukcja obsługi

Aby uruchomić projekt należy:

- Rozpakować archiwum z projektem na dowolne miejsce na dysku
- Uruchomić plik index.html w **Przeglądarce Firefox**. Jest to o tyle ważne, by odtworzyć plik w przeglądarce Firefox, gdyż Firefox jest na tyle elastyczny, że aplikacja będzie działać bez uruchomienia serwera internetowego.

2 Część II

2.1 Schemat projektu



2.2 Część techniczna

Opis poszczególnych komponentów projektu:

- **index.html**
  Główny, uruchomieniowy plik strony, która jest uruchamiana ze startem aplikacji. Zawiera podlinkowane arkusze styli, definicję w **body**, że jest aplikacji typu **Aurelia**. Plik ma również dołączone skrypty, które odpowiadają za całe działanie aplikacji.
- **book.js**
  Skrypt zawierający klasę **Book**, która reprezutuje posiadane książki.
- **app.js**
  Skrypt zawierająct główną klasę **App** odpowiedzialną za całe działanie aplikacji, a

więc dodawanie książek, usuwanie ich, zmienianie ich dostępności oraz zapis i odczyt książek do/z pamięci lokalnej.

- **main.js**
  Skrypt zawierający funkcję odpowiedzialą za konfigurację aplikacji.

- **app.html**
  Szablon strony, która jest renderowana i "wstawiana"do znacznika *body* w pliku **index.html**. Zawiera wyśrodkowany kontener, w którym znajduje się formularz z podpiętą akcją dodania nowej książki oraz tabelę wyświetlającą wprowadzone książki. Całość działa dynamicznie w zależności od przekazanej tablicy książek z pliku **app.js**.

- **style.css**
  Arkusz styli odpowiedzialny za wygląd strony (aplikacji)

3 Pełen kod programu

Pełen kod z podziałem na pliki:

- **index.html**

```
<!DOCTYPE html>
<html>
<head>
    <title>My library</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
                                skeleton/2.0.4/skeleton.css" />
    <link rel="stylesheet" href="src/style.css" />
</head>
<body aurelia-app="src/main">
    <script src="scripts/system.js"></script>
    <script src="scripts/config-esnext.js"></script>
    <script src="scripts/aurelia-core.min.js"></script>
    <script>
        System.import('aurelia-bootstrapper');
    </script>
</body>
</html>
```

- **book.js**

```
export class Book {
    constructor(title, author, isAvailable) {
        this.isAvailable = isAvailable;
        this.title = title;
        this.author = author;
    }
}
```

- **app.js**

```
import {Book} from './book';

export class App {
    constructor() {
        this.heading = "My library";
        this.books = this.getBooksFromStorage();
        this.bookTitle = '';
        this.bookAuthor = '';
        this.bookAvailability = true;
    }

    addBook()
    {
        if(this.bookTitle && this.bookAuthor)
        {
            this.books.push(new Book(this.bookTitle, this.bookAuthor,
                                    this.bookAvailability));

            //Store in LS
            this.storeBook(this.bookTitle, this.bookAuthor,
                           this.bookAvailability);

            //Clear fields
            this.bookAuthor = '';
            this.bookTitle = '';
            this.bookAvailability = true;
        }
    }

    storeBook(title, author, availability)
    {
        let books;
        if(localStorage.getItem('books') === null)
        {
            books = [];
        }
        else
        {
            books = JSON.parse(localStorage.getItem('books'));
        }
        books.push({title: title, author: author,
                                  isAvailable: availability});
        localStorage.setItem('books',JSON.stringify(books));
    }
```

```
    getBooksFromStorage()
    {
        let books;
        if(localStorage.getItem('books') === null)
        {
            books = [];
        }
        else
        {
            books = JSON.parse(localStorage.getItem('books'));
        }
        return books;
    }

    deleteBook(book)
    {
        let index = this.books.indexOf(book);
        if(index !== -1)
        {
            this.books.splice(index, 1);
        }
        this.removeBookFromStorage(index);
    }

    removeBookFromStorage(index)
    {
        let books = JSON.parse(localStorage.getItem('books'));
        books.splice(index,1);
        localStorage.setItem('books',JSON.stringify(books));
    }

    changeBookAvailability(book)
    {
        let index = this.books.indexOf(book);
        book.isAvailable = !book.isAvailable;
        localStorage.setItem('books',JSON.stringify(this.books));
    }
}
```

- **main.js**

```
export function configure(aurelia) {
    aurelia.use.basicConfiguration();
    aurelia.start().then(() => aurelia.setRoot());
}
```

- **app.html**

```
<template>
    <div class="container">
        <h1>${heading}</h1>

        <form submit.trigger = addBook()>
            <label for="bookTitle">Book Title</label>
            <input type="text" value.bind="bookTitle" placeholder="Book
                    Title" class="u-full-width" id="bookTitle">
            <label for="bookAuthor">Book Author</label>
            <input type="text" value.bind="bookAuthor" placeholder="Book
                    Author" class="u-full-width" id="bookAuthor">
            <button type="submit" class="u-full-width">Add Book</button>
        </form>

        <table class="u-full-width">
            <tr>
                <th>Title</th>
                <th>Author</th>
                <th>Change availability</th>
                <th>Delete</th>
            </tr>
            <tr repeat.for="book of books">
                <td> ${book.title}</td>
                <td> ${book.author}</td>
                <td>
                    <button click.trigger="changeBookAvailability(book)">
                        <span if.bind="book.isAvailable">Available</span>
                        <span if.bind="!book.isAvailable">Unavailable</span>
                    </button>
                </td>
                <td>
                    <button click.trigger="deleteBook(book)">Delete</button>
                </td>
            </tr>
        </table>
    </div>
</template>
```

- **style.css**

```
body
{
    background-color: #f6e8a1;
    color: #6e3033;
```

```
}

tr:first-child
{
    background-color: #85deb2;
}

tr
{
    background-color: #e07761;
}

button
{
    background-color: #6e3033;
    color: #e07761;
}

button:hover
{
    background-color: #e07761;
    color: #6e3033;
}

td,th,h1
{
    text-align: center;
}
```