



Міністерство освіти та науки України
Національний технічний університет України “КПІ ім. Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

Звіт
до лабораторної роботи з дисципліни
“ Доступ до зовнішнього API ”
Лабораторна робота №7

Перевірив:
Попенко В. Д.

Виконала:
студентка групи ІС-83
Мороз М.А.

м. Київ-2020

Лабораторна робота 7. Доступ до зовнішнього API

Завдання:

Підключити зовнішній сервіс до свого рішення. Сервіс можна використовувати існуючий або створити власний. Приклади зовнішніх сервісів наведені в Додатку.

Вправа 7.1. Обрати зовнішній сервіс, і описати сценарій його використання. Додати до існуючого в GitHub репозиторію з ЛР6 опис сценарію використання API зовнішнього застосування.

Вправа 7.2. Реалізувати на будь-якій мові програмування виклик API зовнішнього застосування і візуалізацію відповіді. Додати в GitHub код і опис прикладу застосування API.

Для виконання лабораторної роботи було обрано зовнішній сервіс API

<https://ghibliapi.herokuapp.com/films>, який містить інформацію про фільми.

Ідея лабораторної – створити сайт “Multiplex”, можна дізнатися діректора та продюсера фільму, а також переглянути його рейтинг.

Проект складається з трьох файлів: index.html, scripts.js та style.css.

Посилання на гітхаб з файлами проекту, на якому також зберігається відео з результатами роботи сайту (назва відео – “Multiplex_video of work.mp4”):

<https://github.com/mash-mallow/is-83-008>

Файл index.html, який містить html-код сторінки сайту:

```
index.html x style.css x scripts.js x
1 <!DOCTYPE html>
2 <head>
3   <meta charset="UTF-8">
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5
6   <title>Multiplex</title>
7   <link href="https://fonts.googleapis.com/css?family=Dosis:400,700" rel="stylesheet">
8   <link href="style.css" rel="stylesheet">
9 </head>
10
11 <body>
12   <div id="root"></div>
13   <script src="scripts.js"></script>
14 </body>
```

Фрагмент файлу style.css, який містить стилі, що передаються у файл index.html
(повний вміст файлу можна переглянути на прикріпленому вище [github-посиланні](#)):

```
style.css x index.html x scripts.js x
43
44 .card {
45   margin: 1rem;
46   background: #221F1F;
47   box-shadow: 2px 4px 25px rgba(0, 0, 0, .1);
48   border-radius: 12px;
49   overflow: hidden;
50   transition: all .2s linear;
51 }
52
53 .card:hover {
54   box-shadow: 2px 8px 45px rgba(0, 0, 0, .15);
55   transform: translate3D(0, -2px, 0);
56 }
57
58 @media screen and (min-width: 600px) {
59   .card {
60     flex: 1 1 calc(50% - 2rem);
61   }
62 }
63
64 @media screen and (min-width: 900px) {
65   .card {
66     flex: 1 1 calc(33% - 2rem);
67   }
68 }
```

Фрагмент файлу `scripts.js`, що містить код на мові javascript, який реалізує підключення до зовнішнього серверу (повний вміст файлу можна переглянути на прикріпленому вище github-посиланні):

```
1  const app = document.getElementById( elementId: 'root');
2  const logo = document.createElement( tagName: 'img');
3  logo.src = 'logo.png';
4
5  const container = document.createElement( tagName: 'div');
6  container.setAttribute( qualifiedName: 'class', value: 'container');
7  app.appendChild(logo);
8  app.appendChild(container);
9
10 let request = new XMLHttpRequest();
11 request.open( method: 'GET', url: 'https://ghibliapi.herokuapp.com/films', async: true);
12 request.onload = function () {
13     let data = JSON.parse(this.response);
14     if (request.status >= 200 && request.status < 400) {
15         data.forEach(movie => {
16             const card = document.createElement( tagName: 'div');
17             card.setAttribute( qualifiedName: 'class', value: 'card');
18
19             const h1 = document.createElement( tagName: 'h1');
20             h1.textContent = movie.title;
21             const p = document.createElement( tagName: 'p');
22             p.setAttribute( qualifiedName: 'style', value: 'white-space: pre;');
23
24             movie.director = movie.director.substring(0, 100);
25             p.textContent = `Director: ${movie.director}\r\n`;
26             movie.producer = movie.producer.substring(0, 100);
```

Підключення API проводиться за допомогою завантаження зовнішнього JavaScript-файлу, що містить компоненти API, в данному випадку

<https://ghibliapi.herokuapp.com/films>. Далі створюється клас, який буде містити дані про фільм movie, а саме title(назву), director(директора), producer(продюсера), release_date(дату виходу) та rt_score(рейтинг).

Нижче наведено повний код виклику API зовнішнього застосування:

```
let request = new XMLHttpRequest();
request.open( method: 'GET', url: 'https://ghibliapi.herokuapp.com/films', async: true);
request.onload = function () {
    let data = JSON.parse(this.response);
    if (request.status >= 200 && request.status < 400) {
        data.forEach(movie => {
            const card = document.createElement( tagName: 'div');
            card.setAttribute( qualifiedName: 'class', value: 'card');
            const h1 = document.createElement( tagName: 'h1');
            h1.textContent = movie.title;
            const p = document.createElement( tagName: 'p');
            p.setAttribute( qualifiedName: 'style', value: 'white-space: pre;');
            movie.director = movie.director.substring(0, 100);
            p.textContent = `Director: ${movie.director}\r\n`;
            movie.producer = movie.producer.substring(0, 100);
            p.textContent += `Producer: ${movie.producer}\r\n`;
            movie.release_date = movie.release_date.substring(0, 100);
            p.textContent += `Release date: ${movie.release_date}\r\n`;
            movie.rt_score = movie.rt_score.substring(0, 100);
            p.textContent += `Rating IMDb: ${movie.rt_score}\r\n`;
            container.appendChild(card);
            card.appendChild(h1);
            card.appendChild(p);
        });
    } else {
        const errorMessage = document.createElement( tagName: 'marquee');
        errorMessage.textContent = `No answer, please, try again :(`;
        app.appendChild(errorMessage);
    }
}
request.send();
```