МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Лабораторна робота №8**

з дисципліни «Веб технології та веб-дизайн»

**Виконав:**

студент гр. БС-81

Тарабара С.В.

**Перевірив:**

Ас. каф БМК Матвійчук О.В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання:**

Створити 1 сторінковий сайт який має містити наступну інформацію:

Час з моменту останнього оновлення даних. (Бажано в header, та в секундах з точністю до 0.1)

Дані по позиціях на біржі для 8 найменувань (на ваш вибір)

Посилання на тест: [testmoz.com/2985692](https://testmoz.com/q/2985692)

Пароль до тесту: fbmi2020

**Пункт 1:**

Доступ до наних біржі відбувається за допомогою API описаного тут: <https://iexcloud.io/docs/api/>

Ключ доступу: pk\_ba9bb61a36a24ec3a5ae70362941f41c (вже не працює :D)

Частота оновлень: 20 сек

p.s: даний ключ має задавитись конфігураційно та не повинен фігурувати на gitHub

**Пункт 2:**

Таблиця з результатами повинна містити наступні поля:

Назву компанії (Приклад: "Apple Inc.")

Назву позиції (Приклад: "AAPL")

Останню ціну позиції (Приклад: "158.73")

Різницю з моменту останнього оновлення (Приклад: "-1.67")

**При виконанні роботи обов'язково використовувати бібліотеку RXjs**

**Код:**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>laba 8</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h3 id="last-update">Loading...</h3>

<table id="my-table">

<tr>

<th>Company Name</th>

<th>Symbol</th>

<th>Latest Price</th>

<th>Change</th>

</tr>

</table>

<script src="https://unpkg.com/rxjs/bundles/rxjs.umd.min.js"></script>

<script src="script.js"></script>

</body>

</html>

**Script.js**

const {

timer

} = rxjs;

const {

filter,

map,

switchMap

} = rxjs.operators;

const {

fromFetch

} = rxjs.fetch;

// pk\_57b5ebfc1c98439baf983eb6367db5de

const apiBase = 'https://cloud.iexapis.com/stable';

const token = prompt("Enter Your API key : ");

function quote(symbol, field) {

return fromFetch(`${apiBase}/stock/${symbol}/quote/${field}?token=${token}`)

.pipe(switchMap(res => res.json()));

}

const symbols = ['ibm', 'aaon', 'kodk', 'gmed', 'hack', 'hcacw', 'tsla', 'twtr', 'icad', 'iec', 'igov', 'iim', 'goog', 'kodk', 'msft', 'nflx', 'sbux', 'tsla', 'twtr'];

console.log(quote('aapl', 'companyName'))

function main() {

const timer$ = timer(250, 100);

const update$ = timer$.pipe(filter(x => x % 200 == 0));

const table = document.getElementById('my-table');

for (const symbol of symbols) {

const tr = document.createElement('tr');

const tdCompanyName = document.createElement('td');

tr.appendChild(tdCompanyName);

const tdSymbol = document.createElement('td');

tr.appendChild(tdSymbol);

const tdLatestPrice = document.createElement('td');

tr.appendChild(tdLatestPrice);

const tdChange = document.createElement('td');

tr.appendChild(tdChange);

table.appendChild(tr);

// static

console.log(quote(symbol, 'companyName')

.subscribe(text => tdCompanyName.innerHTML = text));

quote(symbol, 'symbol')

.subscribe(text => tdSymbol.innerHTML = text);

// update

update$.subscribe(() => {

quote(symbol, 'latestPrice')

.subscribe(text => tdLatestPrice.innerHTML = text);

quote(symbol, 'change')

.subscribe(text => tdChange.innerHTML = text);

});

}

const lastUpdateElement = document.getElementById('last-update');

let lastUpdate = 0;

update$.subscribe(() => {

lastUpdate = 0.0;

});

timer$.subscribe(x => {

lastUpdate += 0.1;

lastUpdateElement.innerHTML = `Last Update: ${lastUpdate.toFixed(1)}s ago`;

});

}

main();

**Style.css**

table,

td,

th {

border: 1px solid #ddd;

text-align: left;

}

table {

border-collapse: collapse;

width: 50%;

margin: 0 auto;

}

th,

td {

padding: 15px;

}

tr:nth-child(odd) {

background: gray;

}

#last-update {

text-align: center;

margin: 50px auto;

}