**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Теплоенергетичний факультет**

**Кафедра автоматизації проектування  
енергетичних процесів і систем**

***Звіт***

***з лабораторної роботи №1***

***"Розробка продюсера даних"***

***з дисципліни***

***"Обробка потокової інформації"***

**Виконав:**

Студент групи ТІ-01мп

Стеценко Д. О.

**Перевірив:**

Д.ф.-м.н.

Матичин І.І.

**м. Київ – 2021 р.**

**Завдання**:

Написати код продюсера, що імітує потік даних, який генерується в певній предметній області (наприклад, показники масиву сенсорів, дані про угоди купівлі/продажу валюти або цінних паперів, логи тощо), та надсилає її до певного топіку Кафка.

**Передумови**:

Для реалізації продюсера даних була використана платформа **Node.js** і бібліотека **Kafka.js**. Також була використана бібліотека **Rx.js** для створення і генерації потоку даних. Припустимо, що Zookeeper і брокер Kafka вже запущені.

**Хід виконання**:

Спочатку створюємо екземпляр об’єкту Kafka з параметрами підключення, потім на основі нього ініціалізуємо продюсер. Створюємо потік даних за допомогою функції interval і на кожному новому значенні з потоку відправляємо дані в топік “**random-coords**”.

Код продюсера для надсилання даних координат (з частотою .5 секунди) наведений нижче:

const { Kafka } = require('kafkajs');

const { interval } = require('rxjs');

const { map } = require('rxjs/operators');

const { handleErrors } = require('./error-handling');

const { createCoords } = require('./utils');

let sub$;

let msgId = 0;

const kafka = new Kafka({

  clientId: 'coords-app',

  brokers: ['localhost:9092'],

});

const producer = kafka.producer();

const run = async () => {

  await producer.connect();

  sub$ = interval(500)

    .pipe(map(() => JSON.stringify(createCoords(msgId++))))

    .subscribe((*coords*) => {

      console.log('Sending message...', { coords });

      producer.send({

        topic: 'random-coords',

        messages: [{ value: coords }],

      });

    });

};

run().catch(console.error);

handleErrors([sub$]);

Результат виконання програми:

$ node ./lab-1/index.js

> {

coords: '{"lat":"56.1159","long":"74.2273","id":"giio4cpy07e","coordId":97}'

}

> {

coords: '{"lat":"-8.6607","long":"-62.8093","id":"dp2x5bzlgq","coordId":98}'

}

> {

coords: '{"lat":"-72.3148","long":"9.8680","id":"67nxmur0lb7","coordId":99}'

}

> {

coords: '{"lat":"-18.5801","long":"172.8648","id":"eu98a9xh1v","coordId":100}'

}

> {

coords: '{"lat":"-60.6310","long":"-130.2751","id":"w9pltdgvkt","coordId":101}'

}

> {

coords: '{"lat":"-42.6624","long":"-87.4582","id":"2xar5450qju","coordId":102}'

}

> {

coords: '{"lat":"81.3884","long":"-133.4794","id":"nps698stg3","coordId":103}'

}

> {

coords: '{"lat":"-0.2967","long":"-125.2990","id":"gknie93m0h","coordId":104}'

}

> {

coords: '{"lat":"-66.7979","long":"-81.8271","id":"e3s2p9bxxrd","coordId":105}'

}

> {

coords: '{"lat":"-50.7563","long":"106.0039","id":"binknmpknjj","coordId":106}'

}

...