**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 №2***

***"Розробка конс’юмера 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 конс’юмера, що отримує потік даних, який генерується продюсером, та здійснює вибірку кортежів потоку для 1/10 всіх значень певного ключа.

**Передумови**:

Для реалізації конс’юмера даних була використана платформа **Node.js**, бібліотека **Kafka.js**, бібліотека **Rx.js** для створення і генерації потоку даних, та створений у Лабораторній роботі №1 продюсер. Припустимо, що Zookeeper і брокер Kafka вже запущені.

**Хід виконання**:

Спочатку створюємо екземпляр об’єкту Kafka з параметрами підключення, потім на основі нього ініціалізуємо конс’юмер. Для перевірки значення ключа використаємо хеш-функцію, яка повертає значення від 0 до 9. Конс’юмер слухає топік “**random-coords**” і зберігає повідомлення в пам’ять якщо значення ключа дорівнює 0.

Код конс’юмера для отримання даних координат (з частотою .5 секунди) наведений нижче та за посиланням:

<https://github.com/oddestdan/kafka-streams/tree/main/task-2>

const { Kafka } = require('kafkajs');

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

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

let processedCount = 0;

const KEY\_LENGTH = 10;

const savedMessages = [];

const kafka = new Kafka({

  clientId: 'coords-app',

  brokers: ['localhost:9092'],

});

const consumer = kafka.consumer({ groupId: 'random-coords' });

const run = async () => {

  console.log('Consumer is running...');

  await consumer.connect();

  await consumer.subscribe({ topic: 'random-coords', fromBeginning: false });

  await consumer.run({

    eachMessage: async ({ *topic*, *partition*, *message* }) => {

      processedCount++;

      const obj = JSON.parse(message.value.toString());

      const hash = getHashCode(obj.id, KEY\_LENGTH);

      if (hash === 0) {

        console.log('> ', { ...obj });

        savedMessages.push(obj);

      }

    },

  });

};

run().catch((*error*) => {

  console.error(error);

  process.exit(1);

});

process.on('SIGINT', function () {

  consumer.stop();

  console.log('\nTotal processed messages: ', processedCount);

  console.table(

    savedMessages.map((*m*) => (typeof m === 'object' ? m : JSON.parse(m)))

  );

});

handleErrors();

Результат виконання програми:

$ $ node index

Consumer is running...

> { lat: '-40.6548', long: '42.1080', id: 'lqpazflte9', coordId: 8 }

> { lat: '-44.3990', long: '-150.8088', id: '5l5pxolcj8', coordId: 22 }

> { lat: '4.8767', long: '-118.3238', id: 'duj7g2lqlae', coordId: 29 }

> { lat: '29.6558', long: '177.5708', id: 'aaj7pujytpm', coordId: 30 }

> { lat: '-46.3684', long: '111.0699', id: '5vvu9pltqea', coordId: 34 }

> { lat: '67.2041', long: '-22.2181', id: 'gp2mwzd9a1l', coordId: 48 }

> { lat: '84.5053', long: '-56.7451', id: 'dcip3o1eea', coordId: 51 }

> { lat: '-4.6680', long: '-20.5878', id: 'apyh53tilg', coordId: 55 }

> { lat: '58.5499', long: '3.0324', id: '1sphty1sda', coordId: 59 }

> { lat: '65.7665', long: '-161.6269', id: 'yxhmgm2lmef', coordId: 80 }

Total processed messages: 85

┌─────────┬────────────┬─────────────┬───────────────┬─────────┐

│ (index) │ lat │ long │ id │ coordId │

├─────────┼────────────┼─────────────┼───────────────┼─────────┤

│ 0 │ '-40.6548' │ '42.1080' │ 'lqpazflte9' │ 8 │

│ 1 │ '-44.3990' │ '-150.8088' │ '5l5pxolcj8' │ 22 │

│ 2 │ '4.8767' │ '-118.3238' │ 'duj7g2lqlae' │ 29 │

│ 3 │ '29.6558' │ '177.5708' │ 'aaj7pujytpm' │ 30 │

│ 4 │ '-46.3684' │ '111.0699' │ '5vvu9pltqea' │ 34 │

│ 5 │ '67.2041' │ '-22.2181' │ 'gp2mwzd9a1l' │ 48 │

│ 6 │ '84.5053' │ '-56.7451' │ 'dcip3o1eea' │ 51 │

│ 7 │ '-4.6680' │ '-20.5878' │ 'apyh53tilg' │ 55 │

│ 8 │ '58.5499' │ '3.0324' │ '1sphty1sda' │ 59 │

│ 9 │ '65.7665' │ '-161.6269' │ 'yxhmgm2lmef' │ 80 │

└─────────┴────────────┴─────────────┴───────────────┴─────────┘