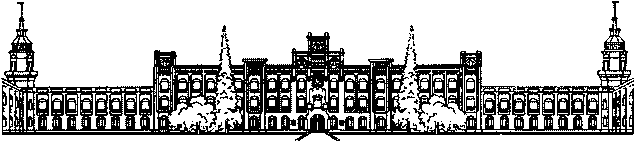
****

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №4

з дисципліни

Сучасні методи обробки масивів даних

|  |  |  |
| --- | --- | --- |
| Виконав(ла): |  | Перевірила: |
|  |  |  |
| студент(ка) групи ІМ-42мп: |  | ст. викладач |
| Бардін В. Д. |  | Тимофєєва Ю.С. |
|  |  |  |

Київ 2024

**1 КОД ЗАСТОСУНКУ**

public class Lab4Streams : IStreamTopologyBuilder  
{  
 private const string InputTopic = "int.k-connect.csv.plastic-pollution";  
  
 private static JsonSerializerSettings *GetJsonSerializerSettings*()  
 {  
 return new JsonSerializerSettings  
 {  
 Converters = { new NaIntConverter() },  
 ContractResolver = new DefaultContractResolver  
 {  
 NamingStrategy = new SnakeCaseNamingStrategy(),  
 }  
 };  
 }  
  
 public StreamBuilder BuildTopology(StreamBuilder streamBuilder)  
 {  
 JsonConvert.DefaultSettings = *GetJsonSerializerSettings*;  
  
 var records = streamBuilder.Stream(InputTopic, new StringSerDes(), new JsonSerDes<PlasticPollutionInfo>())  
 .Filter((\_, v, \_) => v is not null);  
  
 *BuildVolunteersCountTopology*(records);  
 *BuildUkraineTotalCountTopology*(records);  
  
 return streamBuilder;  
 }  
  
 private static void *BuildVolunteersCountTopology*(IKStream<string, PlasticPollutionInfo> records)  
 {  
 const int minEventsThreshold = 10;  
 var volunteersCountStream = records  
 .Filter((\_, v, \_) => v.NumEvents < minEventsThreshold)  
 .Map<string, int>((\_, v, \_) => KeyValuePair.*Create*("passed", v.Volunteers))  
 .GroupByKey<StringSerDes, Int32SerDes>()  
 .Aggregate<int, Int32SerDes>(  
 () => 0,  
 (\_, v, agg) => agg + v  
 )  
 .ToStream();  
  
 volunteersCountStream.Print(Printed<string, int>.*ToOut*());  
 }  
  
 private static void *BuildUkraineTotalCountTopology*(IKStream<string, PlasticPollutionInfo> records)  
 {  
 const string ukraineEventsKey = "Ukraine";  
 var totalCollectedInUkraine = records.MapValues<long>((\_, v, \_) => v.NumEvents)  
 .Filter((k, \_, \_) => k is ukraineEventsKey)  
 .GroupByKey()  
 .Aggregate<long, Int64SerDes>(  
 () => 0,  
 (\_, v, agg) => agg + v  
 )  
 .ToStream();  
   
 totalCollectedInUkraine.Print(Printed<string, long>.*ToOut*());  
 }  
}

**2 СКРІНШОТИ З РЕЗУЛЬТАТАМИ РОБОТИ**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**3 ВИСНОВКИ**

У цій роботі було використано Kafka Streams для обробки вхідного потоку даних та їх агрегації за допомогою використання операцій зі збереженням стану.