**POM.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.training.section8</groupId>

<artifactId>section8</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-sdks-java-core</artifactId>

<version>2.19.0</version>

</dependency>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-runners-direct-java</artifactId>

<version>2.19.0</version>

</dependency>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-sdks-java-io-kafka</artifactId>

<version>2.17.0</version>

</dependency>

<dependency>

<groupId>org.apache.kafka</groupId>

<artifactId>kafka-clients</artifactId>

<version>2.5.0</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.46</version>

</dependency>

<dependency>

<groupId>org.apache.beam</groupId>

<artifactId>beam-sdks-java-io-jdbc</artifactId>

<version>2.4.0</version>

</dependency>

</dependencies>

</project>

RealTimeStreamingETL.java

package section8;

import java.sql.PreparedStatement;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.io.jdbc.JdbcIO;

import org.apache.beam.sdk.io.jdbc.JdbcIO.PreparedStatementSetter;

import org.apache.beam.sdk.io.kafka.KafkaIO;

import org.apache.beam.sdk.transforms.Count;

import org.apache.beam.sdk.transforms.DoFn;

import org.apache.beam.sdk.transforms.ParDo;

import org.apache.beam.sdk.transforms.Values;

import org.apache.beam.sdk.transforms.windowing.FixedWindows;

import org.apache.beam.sdk.transforms.windowing.Window;

import org.apache.beam.sdk.values.KV;

import org.apache.kafka.common.serialization.LongDeserializer;

import org.joda.time.Duration;

public class RealTimeStreamingETL {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

p.apply(KafkaIO.<Long,IotEvent>read()

.withBootstrapServers("localhost:9092")

.withTopic("beamtopic")

.withKeyDeserializer(LongDeserializer.class)

.withValueDeserializer(IotDeserializer.class)

.withoutMetadata()

)

.apply(Values.<IotEvent>create())

.apply(Window.<IotEvent>into(FixedWindows.of(Duration.standardSeconds(10))))

.apply(ParDo.of(new DoFn<IotEvent, String>() {

@ProcessElement

public void processElement(ProcessContext c) {

if(c.element().getTemperature()>80.0) {

c.output(c.element().getDeviceId());

}

}

}))

.apply(Count.perElement())

.apply(JdbcIO.<KV<String,Long>>write().withDataSourceConfiguration(JdbcIO.DataSourceConfiguration

.create("com.mysql.jdbc.Driver", "jdbc:mysql://127.0.0.1:3306/beamdb?useSSL=false")

.withUsername("root").withPassword("root"))

.withStatement("insert into event values (?,?) ")

.withPreparedStatementSetter(new PreparedStatementSetter<KV<String,Long>>() {

public void setParameters(KV<String,Long> element, PreparedStatement preparedStatement) throws Exception {

// TODO Auto-generated method stub

preparedStatement.setString(1, element.getKey());

preparedStatement.setLong(2, element.getValue());

}

})

);

p.run();

}

}

IotDeserializer.java

package section8;

import java.util.Map;

import org.apache.kafka.common.serialization.Deserializer;

import com.fasterxml.jackson.databind.ObjectMapper;

public class IotDeserializer implements Deserializer<IotEvent> {

@Override

public void close() {

// TODO Auto-generated method stub

Deserializer.super.close();

}

@Override

public void configure(Map<String, ?> configs, boolean isKey) {

// TODO Auto-generated method stub

Deserializer.super.configure(configs, isKey);

}

@Override

public IotEvent deserialize(String args, byte[] args1) {

// TODO Auto-generated method stub

ObjectMapper om = new ObjectMapper();

IotEvent iotEvent=null;

try {

iotEvent = om.readValue(args1, IotEvent.class);

}

catch(Exception e) {

System.out.println(e.getMessage());

}

return iotEvent;

}

}

IotEvent.java

**package** section8;

**import** java.io.Serializable;

**public** **class** IotEvent **implements** Serializable {

**private** String deviceId;

**private** String name;

**private** String description;

**private** Long eventtime;

**private** Double temperature;

**private** String unit;

**public** IotEvent() {

}

**public** IotEvent(String deviceId, String name, String description,Long eventtime,Double temperature,String temperature\_unit) {

**this**.deviceId = deviceId;

**this**.name = name;

**this**.description = description;

**this**.eventtime = eventtime;

**this**.temperature = temperature;

**this**.unit = unit;

}

**public** String getDeviceId() {

**return** deviceId;

}

**public** **void** setDeviceId(String deviceId) {

**this**.deviceId = deviceId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** Long getEventtime() {

**return** eventtime;

}

**public** **void** setEventtime(Long eventtime) {

**this**.eventtime = eventtime;

}

**public** Double getTemperature() {

**return** temperature;

}

**public** **void** setTemperature(Double temperature) {

**this**.temperature = temperature;

}

**public** String getUnit() {

**return** unit;

}

**public** **void** setUnit(String unit) {

**this**.unit = unit;

}

}