**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.Section6\_</groupId>

<artifactId>Section6\_</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-parquet</artifactId>

<version>2.19.0</version>

</dependency>

<dependency>

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

<artifactId>beam-sdks-java-io-amazon-web-services</artifactId>

<version>2.6.0</version>

</dependency>

<dependency>

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

<artifactId>beam-sdks-java-extensions-google-cloud-platform-core</artifactId>

<version>2.6.0</version>

</dependency>

</dependencies>

</project>

S3Example.java

**package** section6;

**import** org.apache.beam.sdk.Pipeline;

**import** org.apache.beam.sdk.io.TextIO;

**import** org.apache.beam.sdk.options.PipelineOptionsFactory;

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

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

**import** org.apache.beam.sdk.values.PCollection;

**import** com.amazonaws.auth.AWSCredentials;

**import** com.amazonaws.auth.AWSStaticCredentialsProvider;

**import** com.amazonaws.auth.BasicAWSCredentials;

**public** **class** S3Example {

**public** **static** **void** main(String[] args) {

Options myOption=PipelineOptionsFactory.*fromArgs*(args).withValidation().as(Options.**class**);

Pipeline p = Pipeline.*create*(myOption);

AWSCredentials awsCredObject = **new** BasicAWSCredentials(myOption.getAWSAccessKey(), myOption.getAWSSecretKey());

myOption.setAwsCredentialsProvider(**new** AWSStaticCredentialsProvider(awsCredObject));

PCollection<String> pInput=p.apply(TextIO.*read*().from("s3://beam-udemy-training/input/user\_order.csv"));

pInput.apply(ParDo.*of*(**new** DoFn<String, Void>() {

@ProcessElement

**public** **void** processElement(ProcessContext c) {

System.***out***.println(c.element());

}

}));

p.run();

}

}

Options.java

package section6;

import org.apache.beam.sdk.io.aws.options.S3Options;

import org.apache.beam.sdk.options.PipelineOptions;

public interface Options extends PipelineOptions,S3Options {

void setAWSAccessKey(String val);

String getAWSAccessKey();

void setAWSSecretKey(String val);

String getAWSSecretKey();

void setAwsRegion(String value);

String getAwsRegion();

}

ParquetIOExample.java

**package** section6;

**import** org.apache.avro.Schema;

**import** org.apache.avro.generic.GenericData;

**import** org.apache.avro.generic.GenericRecord;

**import** org.apache.beam.sdk.Pipeline;

**import** org.apache.beam.sdk.coders.AvroCoder;

**import** org.apache.beam.sdk.io.AvroIO;

**import** org.apache.beam.sdk.io.FileIO;

**import** org.apache.beam.sdk.io.TextIO;

**import** org.apache.beam.sdk.io.parquet.ParquetIO;

**import** org.apache.beam.sdk.transforms.MapElements;

**import** org.apache.beam.sdk.transforms.SimpleFunction;

**import** org.apache.beam.sdk.values.PCollection;

**class** BeamCustUtil{

**public** **static** Schema getSchema() {

String SCHEMA\_STRING =

"{\"namespace\": \"training.section6\",\n"

+ " \"type\": \"record\",\n"

+ " \"name\": \"ParquetExample\",\n"

+ " \"fields\": [\n"

+ " {\"name\": \"SessionId\", \"type\": \"string\"},\n"

+ " {\"name\": \"UserId\", \"type\": \"string\"},\n"

+ " {\"name\": \"UserName\", \"type\": \"string\"},\n"

+ " {\"name\": \"VideoId\", \"type\": \"string\"},\n"

+ " {\"name\": \"Duration\", \"type\": \"int\"},\n"

+ " {\"name\": \"StartedTime\", \"type\": \"string\"},\n"

+ " {\"name\": \"Sex\", \"type\": \"string\"}\n"

+ " ]\n"

+ "}";

Schema SCHEMA = **new** Schema.Parser().parse(SCHEMA\_STRING);

**return** SCHEMA;

}

}

**class** ConvertCsvToGeneric **extends** SimpleFunction<String, GenericRecord>{

@Override

**public** GenericRecord apply(String input) {

// **TODO** Auto-generated method stub

String arr[] = input.split(",");

Schema schema=BeamCustUtil.*getSchema*();

GenericRecord record= **new** GenericData.Record(schema);

record.put("SessionId",arr[0]);

record.put("UserId", arr[1]);

record.put("UserName", arr[2]);

record.put("VideoId", arr[3]);

record.put("Duration", Integer.*parseInt*(arr[4]));

record.put("StartedTime", arr[5]);

record.put("Sex", arr[6]);

**return** record;

}

}

**public** **class** ParquetIOExample {

**public** **static** **void** main(String[] args) {

Pipeline p = Pipeline.*create*();

Schema schema = BeamCustUtil.*getSchema*();

PCollection<GenericRecord> pOutput=p.apply(TextIO.*read*().from("C:\\Beam\\user.csv"))

.apply(MapElements.*via*(**new** ConvertCsvToGeneric()))

.setCoder(AvroCoder.*of*(GenericRecord.**class**, schema));

pOutput.apply(FileIO.<GenericRecord>*write*().via(ParquetIO.*sink*(schema)).to("C:\\Beam\\parquetexample.parquet")

.withNumShards(1).withSuffix(".parquet"));

p.run();

}

}

ParquetIOWriteExample.java

package section6;

import org.apache.avro.Schema;

import org.apache.avro.generic.GenericData;

import org.apache.avro.generic.GenericRecord;

import org.apache.beam.sdk.Pipeline;

import org.apache.beam.sdk.coders.AvroCoder;

import org.apache.beam.sdk.coders.Coder;

import org.apache.beam.sdk.io.FileIO;

import org.apache.beam.sdk.io.TextIO;

import org.apache.beam.sdk.io.parquet.ParquetIO;

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

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

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

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

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

class PrintElem extends SimpleFunction<GenericRecord, Void>{

@Override

public Void apply(GenericRecord input) {

// TODO Auto-generated method stub

System.out.println(input.get("SessionId"));

System.out.println("SessionId : "+input.get("SessionId"));

System.out.println("UserId" + input.get("UserId"));

System.out.println("UserName" + input.get("UserName"));

System.out.println("VideoId" + input.get("VideoId"));

System.out.println("Duration" + input.get("Duration"));

System.out.println("StartedTime" + input.get("StartedTime"));

System.out.println("Sex" + input.get("Sex"));

return null;

}

}

public class ParquetIOWriteExample {

public static void main(String[] args) {

Pipeline p = Pipeline.create();

Schema schema = BeamCustUtil.getSchema();

PCollection<GenericRecord> poutput=p.apply(ParquetIO.read(schema).from("C:\\Beam\\parquetexample1\\output-00000-of-00001.parquet"));

poutput.apply(MapElements.via(new PrintElem()));

p.run();

}

}