StarterPipeline.java

**/\***

**\* Licensed to the Apache Software Foundation (ASF) under one**

**\* or more contributor license agreements. See the NOTICE file**

**\* distributed with this work for additional information**

**\* regarding copyright ownership. The ASF licenses this file**

**\* to you under the Apache License, Version 2.0 (the**

**\* "License"); you may not use this file except in compliance**

**\* with the License. You may obtain a copy of the License at**

**\***

**\* http://www.apache.org/licenses/LICENSE-2.0**

**\***

**\* Unless required by applicable law or agreed to in writing, software**

**\* distributed under the License is distributed on an "AS IS" BASIS,**

**\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.**

**\* See the License for the specific language governing permissions and**

**\* limitations under the License.**

**\*/**

**package com.training.section10;**

**import java.util.ArrayList;**

**import java.util.List;**

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

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

**import org.apache.beam.sdk.io.gcp.bigquery.BigQueryIO;**

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

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

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

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

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

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

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

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**import com.google.api.services.bigquery.model.TableFieldSchema;**

**import com.google.api.services.bigquery.model.TableRow;**

**import com.google.api.services.bigquery.model.TableSchema;**

**/\*\***

**\* A starter example for writing Beam programs.**

**\***

**\* <p>The example takes two strings, converts them to their upper-case**

**\* representation and logs them.**

**\***

**\* <p>To run this starter example locally using DirectRunner, just**

**\* execute it without any additional parameters from your favorite development**

**\* environment.**

**\***

**\* <p>To run this starter example using managed resource in Google Cloud**

**\* Platform, you should specify the following command-line options:**

**\* --project=<YOUR\_PROJECT\_ID>**

**\* --stagingLocation=<STAGING\_LOCATION\_IN\_CLOUD\_STORAGE>**

**\* --runner=DataflowRunner**

**\*/**

**public class StarterPipeline {**

**private static final Logger LOG = LoggerFactory.getLogger(StarterPipeline.class);**

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

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

**myOption.setTempLocation("gs://udemy\_training\_beam/input");**

**myOption.setStagingLocation("gs://udemy\_training\_beam/input");**

**myOption.setProject("beam-276305");**

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

**List<TableFieldSchema> columns = new ArrayList<TableFieldSchema>();**

**columns.add(new TableFieldSchema().setName("userId").setType("STRING"));**

**columns.add(new TableFieldSchema().setName("orderId").setType("STRING"));**

**columns.add(new TableFieldSchema().setName("name").setType("STRING"));**

**columns.add(new TableFieldSchema().setName("productId").setType("STRING"));**

**columns.add(new TableFieldSchema().setName("Amount").setType("INTEGER"));**

**columns.add(new TableFieldSchema().setName("order\_date").setType("STRING"));**

**columns.add(new TableFieldSchema().setName("country").setType("STRING"));**

**TableSchema tblSchema = new TableSchema().setFields(columns);**

**PCollection<String> pInput =p.apply(TextIO.read().from("gs://udemy\_training\_beam/input/user.csv"));**

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

**@ProcessElement**

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

**String arr[] = c.element().split(",");**

**if(arr.length==7) {**

**if(arr[6].equalsIgnoreCase("India")) {**

**TableRow row = new TableRow();**

**row.set("userId", arr[0]);**

**row.set("orderId", arr[1]);**

**row.set("name", arr[2]);**

**row.set("productId", arr[3]);**

**row.set("Amount", Integer.valueOf(arr[4]));**

**row.set("order\_date", arr[5]);**

**row.set("country", arr[6]);**

**c.output(row);**

**}**

**}**

**}**

**}))**

**.apply(BigQueryIO.writeTableRows().to("udemy\_training.user\_tbl")**

**.withSchema(tblSchema)**

**.withWriteDisposition(BigQueryIO.Write.WriteDisposition.WRITE\_APPEND)**

**.withCreateDisposition(BigQueryIO.Write.CreateDisposition.CREATE\_IF\_NEEDED)**

**);**

**p.run();**

**}**

**}**

MyOption.java

**package** com.training.section10;

**import** org.apache.beam.runners.dataflow.options.DataflowPipelineOptions;

**public** **interface** MyOption **extends** DataflowPipelineOptions {

}