Assignment Solution

Week11: Apache Spark - Structured API Part-1

IMPORTANT

Self-assessment enables students to develop:

1. A sense of responsibility for their own learning and the ability & desire to continue learning,
2. Self-knowledge & capacity to assess their own performance critically & accurately, and
3. An understanding of how to apply their knowledge and abilities in different contexts.

All assignments are for self assessment. Solutions will be released on every subsequent week.Once the solution is out, evaluate yourself.

No discussions/queries allowed on assignment questions in slack channel.

Note *You can raise your doubts once the solution is released*

Solution 1:

```
//Assignemnet-Problem 1

import org.apache.log4j.Level
import org.apache.log4j.Logger
import org.apache.spark.SparkConf
import org.apache.spark.sql.SaveMode
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.types.DoubleType
import org.apache.spark.sql.types.IntegerType
import org.apache.spark.sql.types.StringType
import org.apache.spark.sql.types.StructField
import org.apache.spark.sql.types.StructType

object Spark_Assignment_windowdata extends App

{ //creating sparkConf object

  val sparkConf = new SparkConf()

sparkConf.set("spark.master","local[2]")
```

```
//Step1 -creating a spark session
 val spark = SparkSession.builder()
 .config(sparkConf)
 .getOrCreate()
 //Step 2 - Setting the logging level to Error
 Logger.getLogger(“org”).setLevel(Level.ERROR)

 // Step 3 Explicit schema definition programmatically using
 StructType val windowdataSchema = StructType(List(
     StructField(“Country”,StringType),
     StructField(“weeknum”,IntegerType),
     StructField(“numinvoices”,IntegerType),
     StructField(“totalquantity”,IntegerType),
     StructField(“invoicevalue”,DoubleType)


    ))
```

```
// Step 3 contd.. Loading the file and creation of dataframe using dataframe reader API, using explicitly specified schema
   val windowdataDF = spark.read
   .format(“csv”)
   .schema(windowdataSchema)
   .option(“path”, “C:/xyz/TrendyTech/Spark_data/structuredAPI/windowdata.csv”)
   .load()

   //print first 20 records of the dataframe
   windowdataDF.show()

//Step 4: Saving the data in Parquet format using Dataframe Writer API
   //Data is two-level partitioned on Country and weeknum column , these columns have low cardinality //Default output format is parquet
  /* windowdataDF.write
   .partitionBy(“Country”, “weeknum”)
   .mode(SaveMode.Overwrite)
   .option(“path”,”C:/xyz/TrendyTech/Spark_data/structuredAPI/Output/windowdata_output”)
   .save()
```

```
  //Step 5: Save the Dataframe to Avro Format and also partitioning data by Country column
  windowdataDF.write
  .format("avro")
  .partitionBy("Country")
  .mode(SaveMode.Overwrite)
  .option("path","C:/xyz/TrendyTech/Spark_data/structuredAPI/Output/windowdata_avrooutput")
  .save()

}
```

Solution 2:

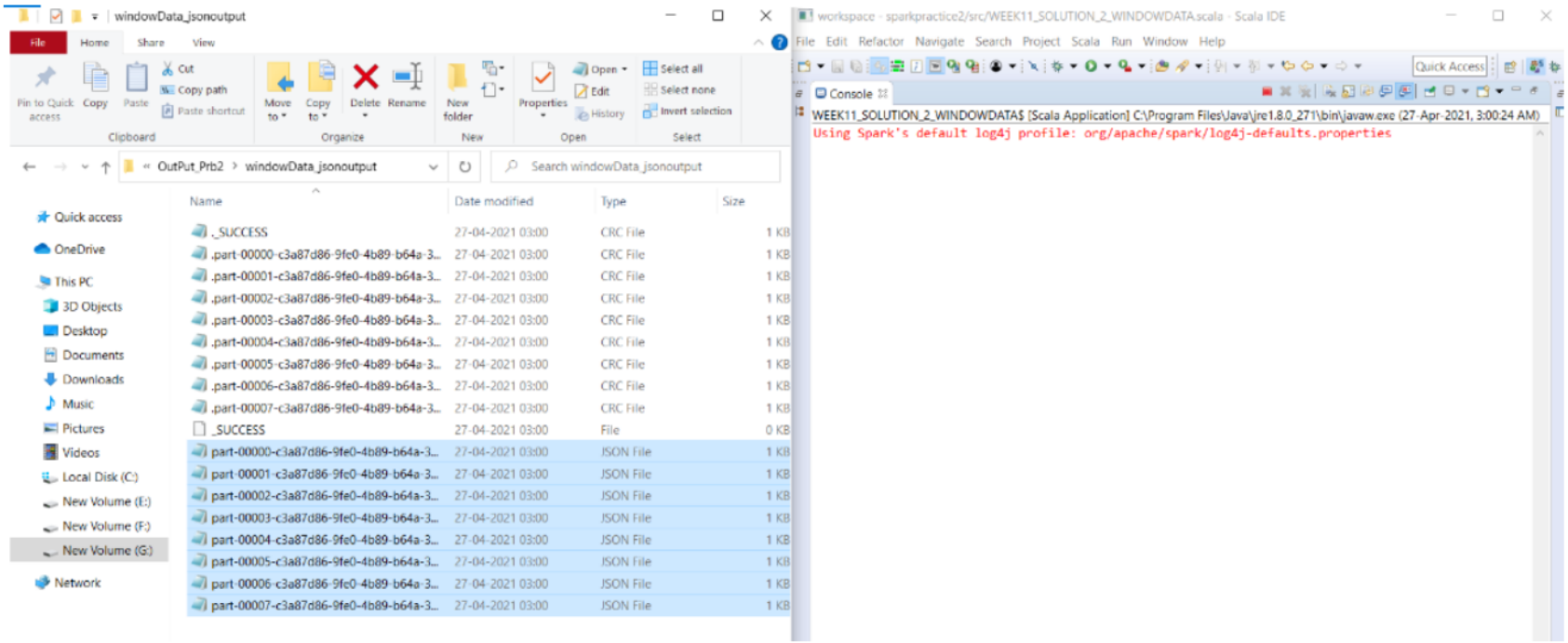

```
import org.apache.log4j.Level
import org.apache.log4j.Logger
import org.apache.spark.SparkConf
import org.apache.spark.sql.SaveMode
import org.apache.spark.sql.SparkSession
object WEEK11_SOLUTION_2_WINDOWDATA extends App {
  // Setting the Logging~Level To ERROR
  Logger.getLogger("org").setLevel(Level.ERROR)

  // define a schema for employee record data using a case class
  case class windowData(Country: String, Weeknum: Int, NumInvoices: Int, TotalQuantity: Int,
InvoiceValue: String)
  //Create Spark Config Object
  val sparkConf = new SparkConf()
  sparkConf.set("spark.app.name", "WEEK11_SOLUTION_2_WINDOWDATA")
  sparkConf.set("spark.master", "local[2]")
  //Create Spark Session
  val spark = SparkSession.builder()
.config(sparkConf)
.getOrCreate()
```

```
  import spark.implicits._

  val windowDataDF = spark.sparkContext.textFile("G:/TRENDY~TECH/WEEK-
11/Assignment_Dataset/windowdata-201021-002706.csv") //.toDF
    .map(_.split(","))
    .map(e => windowData(e(0), e(1).trim.toInt, e(2).trim.toInt, e(3).trim.toInt, e(4)))
    .toDF()
    .repartition(8)

  windowDataDF.write
    .format("json")
    .mode(SaveMode.Overwrite)
    .option("path", "G:/TRENDY~TECH/WEEK-
11/Assignment_Dataset/OutPut_Prb2/windowData_jsonoutput")
    .save()
  //windowDataDF.show()
  spark.stop()
  scala.io.StdIn.readLine()
}
```



Google REVIEWS

5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT

9108179578

Call for more details