EXERCICE : envoyer un csv via scala et consommer le contenu en DF avec spark streaming

1. Envoie de fichier
2. **object** ProducerInputProject **extends** App {  
     
    **import** java.util.Properties  
     
    **import** org.apache.kafka.clients.producer.\_  
     
    **val** *props* = **new** Properties()  
    *props*.put("bootstrap.servers", "localhost:9092")  
     
    *props*.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer")  
    *props*.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer")  
     
    **val** *producer* = **new** KafkaProducer[String, String](*props*)  
     
    **val** *TOPIC* = "topicDataRaw"  
     
    **val** *files* = **new** File("/partage/ble").listFiles().filter(\_.isFile)  
     
    **for**( file <- *files*){  
    **val** nameFile = file.getName  
    **for**( line <- Source.*fromFile*(file).getLines()){  
    **val** record = **new** ProducerRecord(*TOPIC*, nameFile, line)  
    *println*(line)  
    *producer*.send(record)  
    }  
    }  
     
    *producer*.close()  
   }
3. Consommation du fichier

**import** org.apache.spark.sql.SparkSession  
  
**import** org.apache.spark.sql.functions.\_  
  
  
**object** StreamingSample **extends** App {  
  
 **val** *spark* = SparkSession.*builder*.appName("Spark-Kafka-Integration").master("local").getOrCreate()  
 *spark*.sparkContext.setLogLevel("ERROR")  
  
 **val** *df* = *spark* .readStream  
  
 .format("kafka")  
  
 .option("kafka.bootstrap.servers", "localhost:9092")  
  
 .option("subscribe", "topicDataRaw")  
  
 .load()  
  
 **import** *spark*.implicits.\_  
  
 //V1  
  
 **val** *df1* = *df*.selectExpr("CAST(value AS STRING)").as[String]  
  
 .withColumn("data",*split*(*col*("value"),"\t")(0))  
 .withColumn("data1",*split*(*col*("value"),"\t")(1))  
 .withColumn("data2",*split*(*col*("value"),"\t")(2))  
 .withColumn("data3",*split*(*col*("value"),"\t")(3))  
 .withColumn("data4",*split*(*col*("value"),"\t")(4))  
 .withColumn("data5",*split*(*col*("value"),"\t")(5))  
 .withColumn("data3",*split*(*col*("value"),"\t")(6)).drop("value")  
  
  
  
 *df1*.writeStream  
 .format("console")  
 .option("truncate","true")  
 .start()  
  
 .awaitTermination()  
// df1.take(10).foreach(println)  
}