spark命令：

|  |
| --- |
| spark-submit --master yarn --deploy-mode client --driver-memory 1g --executor-memory 4g --executor-cores 2 --num-executors 10 --class com.jollychic.example.HiveDataToEsCookieidpro /home/userprofile/yubo/eshivedata/target/spark-streaming-demo-2.1.1-1.0-SNAPSHOT-jar-with-dependencies.jar |

scala文件：

|  |
| --- |
| **package** com.jollychic.example  **import** java.time.LocalDate **import** java.time.format.{DateTimeFormatter, DateTimeFormatterBuilder, SignStyle} **import** java.time.temporal.ChronoField.{*DAY\_OF\_MONTH*, *MONTH\_OF\_YEAR*, *YEAR*} **import** java.util.Map.Entry  **import** org.apache.spark.sql.SparkSession **import** org.elasticsearch.spark.rdd.EsSpark  */\*\*  \* Author: Boris  \* Date: 2018/11/6 14:31  \* Description:   \*/* **object** HiveDataToEsCookieid {   **var** *FORMATTER*: DateTimeFormatter = **new** DateTimeFormatterBuilder()  .appendValue(*YEAR*, 4, 10, SignStyle.*EXCEEDS\_PAD*)  .appendValue(*MONTH\_OF\_YEAR*, 2)  .appendValue(*DAY\_OF\_MONTH*, 2)  .toFormatter  // val conf: SparkConf = new SparkConf() // .setAppName("appName").setMaster("local[\*]") // conf.set("es.index.auto.create", "true") // conf.set("es.nodes", "172.31.2.215")   **def** main(args: Array[String]): Unit = {    **val** spark = SparkSession.*builder*()  .appName("EsData")  //.master("local[\*]")  .config("spark.serializer", "org.apache.spark.serializer.KryoSerializer")  .config("spark.dynamicAllocation.enabled", "false")  .config("es.index.auto.create", "true")  .config("es.nodes", "10.121.26.235")  .config("es.batch.write.retry.count", "2")  .config("es.batch.write.retry.wait", "30")  .config("thread\_pool.write.queue\_size", "1000")  .config("thread\_pool.write.size", "50")  .config("thread\_pool.write.type", "blocking")  .config("es.batch.size.bytes", "10mb")  //.config("es.batch.write.refresh", "false")  .enableHiveSupport()  .getOrCreate()   **val** dataStr = *getDateStr*();   **import** spark.sql;    **val** hiveDF = sql(  s"""  | SELECT cookieid, tagsmap FROM dw.dw\_profile\_user\_tag\_cookie where data\_date = '**$**{dataStr}' limit 10  """.stripMargin)   **val** rdd = hiveDF.*rdd*.map {  row => {  **val** cookieid = row.getAs[String]("cookieid")  **val** tags = scala.collection.mutable.Map(row.getAs[Map[String, Any]]("tagsmap").toSeq:\_\*)  **for** ((key, value)<-tags ) {  **try** {  **val** newValue = value.toString.toFloat  tags.+((key, newValue))  }**catch**{  **case** e:Exception => { }  }  }  *Map*("cookieid" -> cookieid, "tags" -> tags)  }  }.coalesce(1, **true**)  // rdd/\*saveToEs("20181106\_userid/tags", Map("es.mapping.id"->"userid"))\*/  EsSpark.*saveToEs*(rdd , dataStr+"\_cookieid/tags", *Map*[String,String]("es.mapping.id"->"cookieid"))  spark.stop()  }   */\*\*  \* 获取昨天日期字符串 比如：20181105  \** ***@return*** *\*/* **def** getDateStr(): String = {  **val** yesterDay = LocalDate.*now*().minusDays(1);  **val** date = yesterDay.format(*FORMATTER*);  **return** date;  } } |