

Flink CEP基础学习与使用04----实际业务场景开发案例demo

目标：针对日志，做场景匹配demo，这个demo先发出来吧，感觉乱七八糟的，等过一段时间再思考一下怎么写，这个写的太乱了

```
1 import com.alibaba.fastjson.{JSON, JSONObject}
2 import org.apache.flink.api.common.serialization.SimpleStringSchema
3 import org.apache.flink.api.java.utils.ParameterTool
4 import org.apache.flink.cep.scala.{CEP, PatternStream}
5 import org.apache.flink.cep.scala.pattern.Pattern
6 import org.apache.flink.streaming.api.scala.{DataStream, StreamExecutionEnvironment, _}
7 import org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer010
8
9 import scala.collection.Map
10
11 object Storm2FlinkCEP_demo {
12   def main(args: Array[String]): Unit = {
13
14     val env = StreamExecutionEnvironment.getExecutionEnvironment
15
16     val proString = Array[String]("--input-topic",
17       "dianyou_wxgz_test2",
18       "--output-topic",
19       "dianyou_wxgz_test3",
20       "--bootstrap.servers",
21       "172.1.1.141:9092,172.1.1.142:9092,172.1.1.143:9092",
22       "--zookeeper.connect",
23       "172.10.4.63:2181,172.10.4.64:2181,172.10.4.65:2181",
24       "--group.id", "cc")
25
26     val parameterTool = ParameterTool.fromArgs(proString)
27
28
29     val kafkaDstream: DataStream[String] = env.addSource(new FlinkKafkaConsumer010[String]
30       ("dianyou_wxgz_test2",
31         new SimpleStringSchema,
32         parameterTool.getProperties)
33       )
34
35
36     //todo 这是模拟的Map数据
37     val redisMap = new util.HashMap[String, String]()
38     redisMap.put("row_01", "{ 'id':1, 'A': 'bbb', 'B': 'aaa' }")
39     redisMap.put("row_02", "{ 'id':2, 'A': 'bbb', 'B': 'ccc' }")
40     redisMap.put("row_03", "{ 'id':3, 'A': 'bbb', 'B': 'ddd' }")
41     redisMap.put("row_04", "{ 'id':4, 'A': 'aaa', 'B': 'ccc' }")
```

```

42
43 // 将kafka的数据转成JSON
44 val jsonDstream: DataStream[JSONObject] = kafkaDstream.map(line => {
45     val rs = JSON.parseObject(line)
46     rs
47 }).setParallelism(1)
48
49 // 定义模式
50 val pattern = Pattern.begin[JSONObject]("start")
51 // todo 第一次匹配
52 .where(log => {
53     val rowId = log.getString("rowId")
54     val user = log.getString("user")
55
56     if (redisMap.containsKey(rowId)) {
57         val redisStr = redisMap.get(rowId)
58         val redisJson = JSON.parseObject(redisStr)
59
60         redisJson.getString("A").equals(user) || redisJson.getString("B").equals(user)
61
62     } else {
63         false
64     }
65 })
66 .followedBy("middle") //todo 到这里的话只剩两条日志了。
67 .where(line => {
68     println("到middle的数据: "+line)
69
70     val rowId = line.getString("rowId")
71     val user = line.getString("user")
72
73     if (redisMap.containsKey(rowId)) {
74         val redisStr = redisMap.get(rowId)
75         val redisJson = JSON.parseObject(redisStr)
76
77         if (redisJson.getString("A").equals(user) || redisJson.getString("B").equals(user))
78     {
79         true
80     } else {
81         false
82     }
83
84 } else {
85     false
86 }
87
88 }).oneOrMore
89 .until(event=>{
90     event.containsKey("end") && event.getString("end").equals("true")
91 } )
92
93
94
95
96

```

```
97 // 拿到结果
98 val dataStreamPattern: PatternStream[JSONObject] = CEP.pattern(jsonDstream, pattern)
99
100 val rsDstream: DataStream[JSONObject] = dataStreamPattern.select((pat: Map[String,
Iterable[JSONObject]]) => {
101     val rs: JSONObject = pat.get("middle").get.iterator.next()
102     rs
103 })
104 rsDstream.print()
105
106
107
108 //执行
109 env.execute("execute Flink App")
110
111 }
112 }
```