Spark解析Json数据(非Sql方式)

Spark支持读取很多格式的文件，其中包括了所有继承了Hadoop的InputFormat类的输入文件，以及平时我们常用的Text、Json、CSV (Comma Separated Values) 以及TSV (Tab Separated Values)文件。本文主要介绍如何通过Spark来读取Json文件。很多人会说，直接用Spark SQL模块的jsonFile方法不就可以读取解析Json文件吗？是的，没错，我们是可以通过那个读取Json数据，但是本文本着学习的心态，来自己解析Json数据。

本文通过play-json_2.10依赖包进行解析的，如果你用Java的话，可以使用Jackson（当然Scala也是可以使用Jackson解析Json数据的）；如果你是用python，可以使用内置的Json解析函数。我们可以这样来使用：

```
import play.api.libs.json._

val input = sc.parallelize(List( """{"name":"过往记忆","website":"www.iteblog.com"}""",
                                 """{"other":"过往记忆"}"""))
val parsed = input.map(Json.parse)
parsed.collect

output:
{"name":"过往记忆","website":"www.iteblog.com"}
{"other":"过往记忆"}
```

这样很就解析出Json了，但是我们只是简单将它转换成字符串了，相当于还是没解析。上面的Json数据每条Json格式不一样，如果你的Json数据格式都一样，比如每条Json最多只包含了name和website属性，那么可以这样解析：

```
import play.api.libs.json._

val input = sc.parallelize(List( """{"name":"过往记忆","website":"www.iteblog.com"}""",
                                 """{"name":"过往记忆"}"""))
val parsed = input.map(Json.parse)

case class Info(name: String, website: String) {
   override def toString: String = name + "₩t" + website
}

implicit val personReads = Json.format[Info]

val result = parsed.flatMap(record => personReads.reads(record).asOpt)
result.collect
```

output:
过往记忆 www.iteblog.com

也就是把Json数据解析出来了，并存储在Info类型中，这样就便于下面我们的处理。细心的同学可能会说，这个Json不是有两条数据吗？{"name":"过往记忆"}这条数据为什么没打出了？这是因为，我们不能保证输入的Json数据格式都是包含了name和website属性，如果不包含这两个属性的Json数据我们认为其是错误的数据，也就是要过滤掉。我们在程序中使用到asOpt和flatMap，它的功能是当解析失败的时候将那条失败的数据过滤掉。

如果我们需要将计算的结果保存成Json的数据可以如下操作：

```
val data = sc.parallelize(List(Info("过往记忆", "www.iteblog.com")))
data.map(Json.toJson(_)).collect.foreach(println)
```

结果是{"name":"过往记忆","website":"www.iteblog.com"}。

需要正常运行上面的程序，需要引入相关的依赖包：如果你是用Maven，请在你的pom.xml文件加入以下依赖：

```
<dependency>
        <groupId>com.typesafe.play</groupId>
        <artifactId>play-json_2.10</artifactId>
        <version>2.4.0-M1</version>
</dependency>
```

如果你是用sbt，请在build.sbt文件加入以下依赖：

```
"com.typesafe.play" % "play-json_2.10" % "2.2.1"
```

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接: 【】（）