**说下schema（写Parquet格式数据需要schema，读取的话"自动识别"了schema）**

/\*

\* 每一个字段有三个属性：重复数、数据类型和字段名，重复数可以是以下三种：

\* required(出现1次)

\* repeated(出现0次或多次)

\* optional(出现0次或1次)

\* 每一个字段的数据类型可以分成两种：

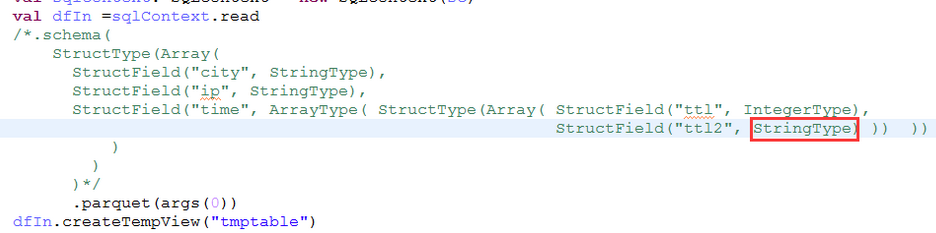
\* group(复杂类型)

\* primitive(基本类型)  
 \* 数据类型有  
 \* INT64, INT32, BOOLEAN, BINARY, FLOAT, DOUBLE, INT96, FIXED\_LEN\_BYTE\_ARRAY  
 \*/  
这个repeated和required 不光是次数上的区别，序列化后生成的数据类型也不同，  
比如repeqted修饰 ttl2 打印出来为 WrappedArray([7,7\_a])   
而 required修饰 ttl2 打印出来为 [7,7\_a]

**除了用MessageTypeParser.parseMessageType类生成MessageType** 还可以用下面方法  
（注意这里有个坑--spark里会有这个问题--ttl2这里 as(OriginalType.UTF8) 和 required binary city (UTF8)作用一样，加上UTF8，在读取的时候可以转为StringType，不加的话会报错 [B cannot be cast to java.lang.String ）

|  |
| --- |
| /\*MessageType schema = MessageTypeParser.parseMessageType("message Pair {\n" +  " required binary city (UTF8);\n" +  " required binary ip (UTF8);\n" +  "repeated group time {\n"+  "required int32 ttl;\n"+  "required binary ttl2;\n"+  "}\n"+  "}");\*/  //import org.apache.parquet.schema.Types;  MessageType schema = Types.buildMessage()  .required(PrimitiveTypeName.BINARY).as(OriginalType.UTF8).named("city")  .required(PrimitiveTypeName.BINARY).as(OriginalType.UTF8).named("ip")  .repeatedGroup().required(PrimitiveTypeName.INT32).named("ttl")  .required(PrimitiveTypeName.BINARY).as(OriginalType.UTF8).named("ttl2").named("time").named("Pair"); |

解决 [B cannot be cast to java.lang.String 异常：  
**1.要么生成parquet文件的时候加个UTF8  
2.要么读取的时候再提供一个同样的schema类指定该字段类型，比如下面:**



<dependency>

<groupId>org.apache.parquet</groupId>

<artifactId>parquet-hadoop</artifactId>

<version>1.7.0</version>

</dependency>