**DataShuffle2.0使用说明文档**

前言

DataShuffle是一个基于SparkStreaming做实时数据清洗的SDK，保留了1.0高效的类型监测方法，在1.0的基础之上做了进一步的抽象，彻底将数据的清洗规则从运行逻辑中剥离出来，用户可以使用自带的数据清洗规则，也可以根据业务逻辑做进一步的定制化开发。

1. **概述**

**1.1整体流程分析**



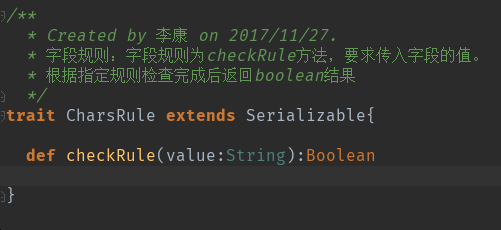
**1.2包目录结构**

项目的包目录结构按照功能进行划分总共分为四个部分：dataRule，inputFormat，shuffle，util。dataRule包含了CharsRule接口以及CharsRule的实现类；inputFormat则包含了InputFormat接口及其实现类；shuffle包下是数据清洗的单例对象ShuffleDataToDstream；util包则是对一些通用到的类型检测方法的封装。

1. **各功能模块介绍**

**2.1dataRule**

dataRule是实现清洗逻辑与流处理流程分离的重要一环。dataRule包下主要是CharsRule接口及其实现类。其中CharsRule代码如下所示：



charsRule很简单只是继承了scala的可序列化以及一个checkRule抽象方法。

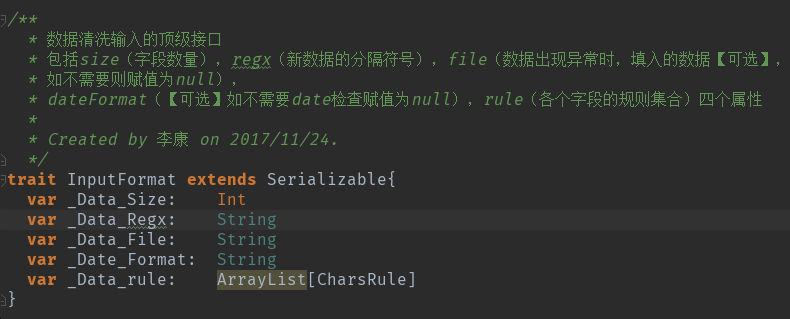
CharsRule的实现类通过继承charsRule接口并重写checkRule方法来实现检查规则的自定义，在2.0中提供了很多基础的CharsRule的实现类，IPRule，StringRule，IntRule，DateRule等等。都是在上文提到的util中工具类具体功能的包装。通过Input对象传给清洗程序。

例：IPRule的代码如下图所示：



**2.2inputFormat**

inputFort是数据清洗的规则进行包装的地方。主要包含InputFormat对象及其实现类。InputFormat对象定义了数据清洗方法中需要传入的属性，代码如下图所示：



同样的，InputFormat也给出了两个经常使用的日志类型的输入对象，FlowInputFormat以及DNSInputFormat。如需对这两种数据格式的日志进行数据清洗则只需要创建对象即可无须手写InputFormat对象，如果需要定制日志格式，则需要自己实现InputFormat对象并重新赋值所有属性。

**2.3shuffle**

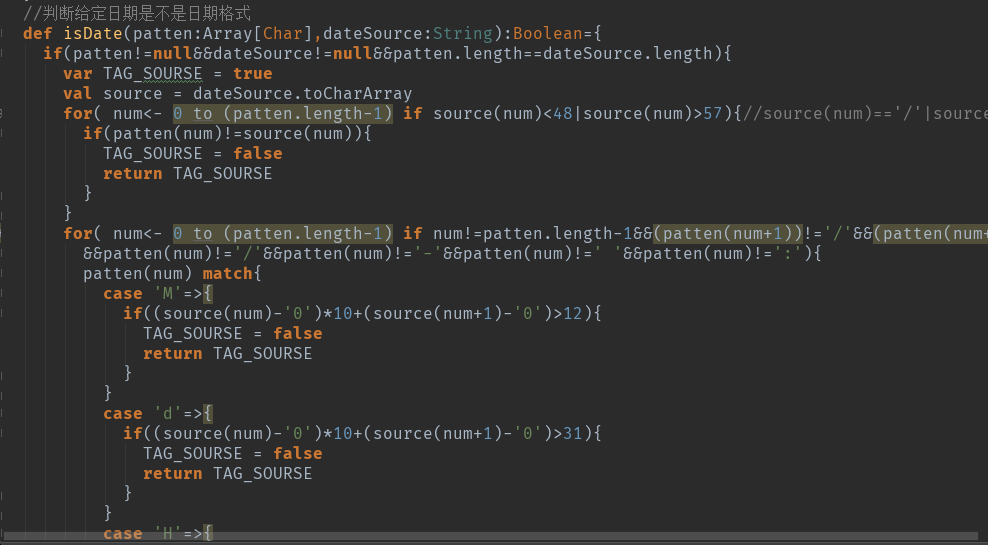
Shuffle包是数据清洗发生的地方，只包含一个单例对象DataShuffleToDstream对象，并对外提供shuffle2Dstream方法，接收dstream，inputFormat对象返回一个包含了最终结果的Dstream对象，需要注意的是传入的Dstream对象必须是Dstream[String],返回的Dstream是Dstream[tuple2[String,String]]。Tuple.\_1为最终正确的数据，Tuple.\_2为出现异常的数据。用户可以根据自己的需要取不同的数据。

**2.4util**

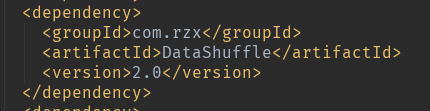
Util包封装了工具类，DataCheckUtil，是整个项目的基石。DatacheckUtil主要包含了数值类型检查，日期类型检查等数据格式检测的方法。

例：数值类型检查以及日期类型检查方法如下图所示：

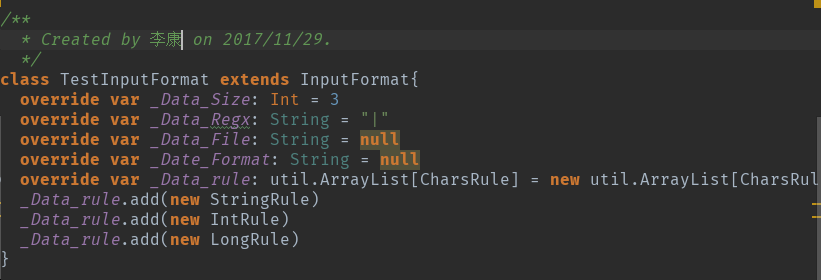




1. **具体开发实例**
2. 程序的jar会传至私服，主需要将jar导入mvn依赖即可。

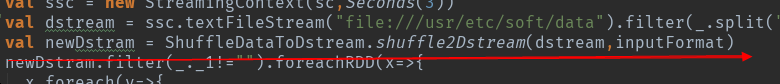


1. 初始化InputFormat对象。如果是flow或者dns日志，只需要在程序中直接new FlowInputFormat对象或DNSInputFormat对象。如果想要自定义自己的日志格式则需要手写inputformat类编辑日志属性。



其中如果不需要file或者dateformat则可以设置为null，其他属性则为必选属性，在datarule中如果有没有赋值的数据则默认为StringRule。

1. 调用方法进行清洗：



通过单例对象ShuffleDataToDstream的shuffle2Dstream方法传入dstream[String]和inputFormat进行清洗。最终返回Dstream[Tuple2[String,String]].包含最终正确数据以及错误数据。