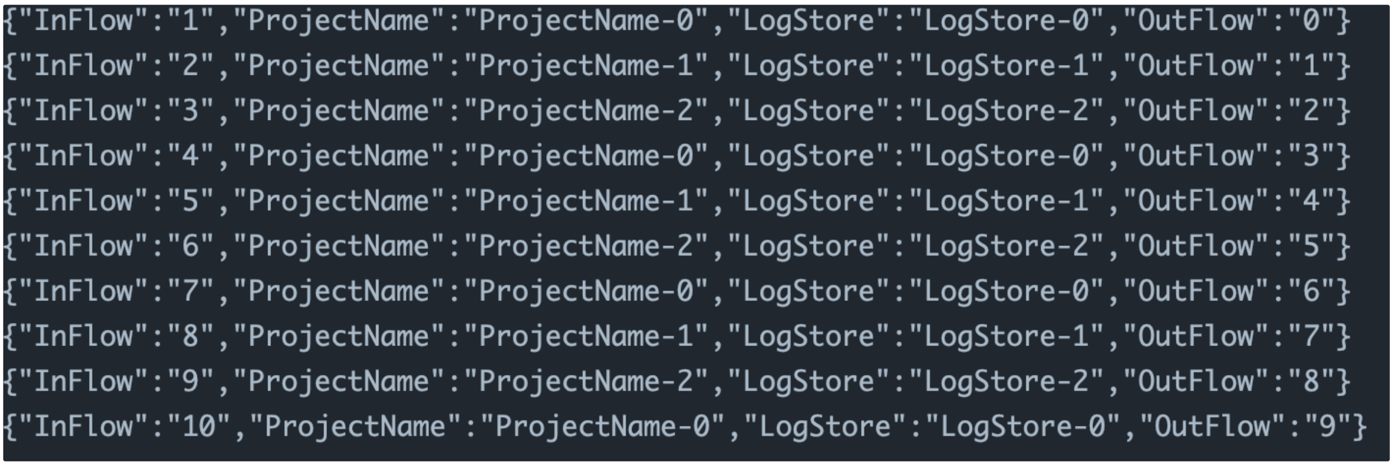
rocketmq-streams作业

> 2101210318
>
> 吕浩毅

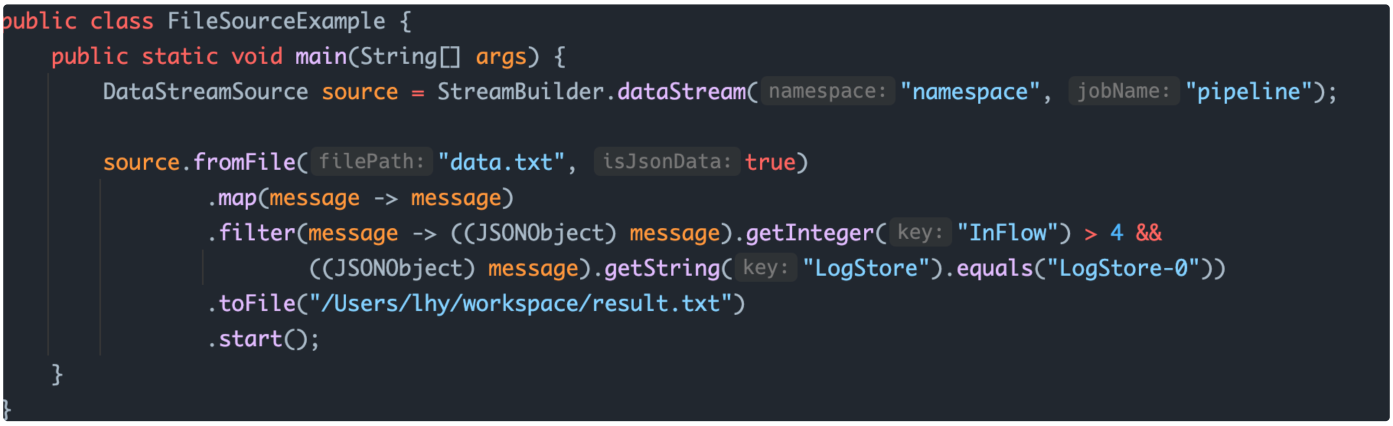
源文件`data.txt`内容



```
{"InFlow":"1","ProjectName":"ProjectName-0","LogStore":"LogStore-0","OutFlow":"0"}
{"InFlow":"2","ProjectName":"ProjectName-1","LogStore":"LogStore-1","OutFlow":"1"}
{"InFlow":"3","ProjectName":"ProjectName-2","LogStore":"LogStore-2","OutFlow":"2"}
{"InFlow":"4","ProjectName":"ProjectName-0","LogStore":"LogStore-0","OutFlow":"3"}
{"InFlow":"5","ProjectName":"ProjectName-1","LogStore":"LogStore-1","OutFlow":"4"}
{"InFlow":"6","ProjectName":"ProjectName-2","LogStore":"LogStore-2","OutFlow":"5"}
{"InFlow":"7","ProjectName":"ProjectName-0","LogStore":"LogStore-0","OutFlow":"6"}
{"InFlow":"8","ProjectName":"ProjectName-1","LogStore":"LogStore-1","OutFlow":"7"}
{"InFlow":"9","ProjectName":"ProjectName-2","LogStore":"LogStore-2","OutFlow":"8"}
{"InFlow":"10","ProjectName":"ProjectName-0","LogStore":"LogStore-0","OutFlow":"9"}
```

输出完整的数据

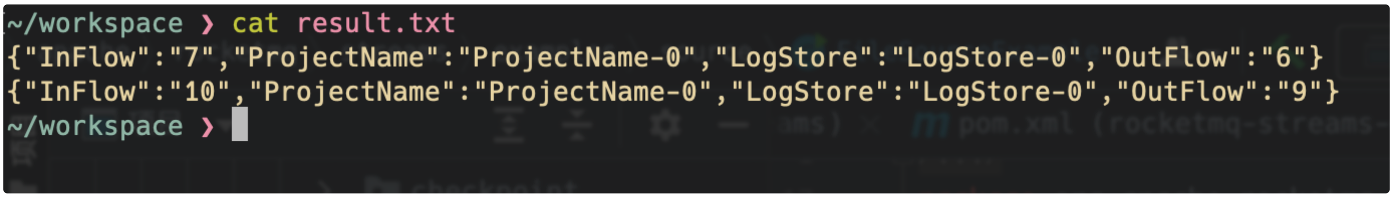
代码



```
public class FileSourceExample {
    public static void main(String[] args) {
        DataStreamSource source = StreamBuilder.dataStream( namespace: "namespace",  jobName: "pipeline");

        source.fromFile( filePath: "data.txt",  isJsonData: true)
                .map(message -> message)
                .filter(message -> ((JSONObject) message).getInteger( key: "InFlow") > 4 &&
                        ((JSONObject) message).getString( key: "LogStore").equals("LogStore-0"))
                .toFile("/Users/lhy/workspace/result.txt")
                .start();
    }
}
```

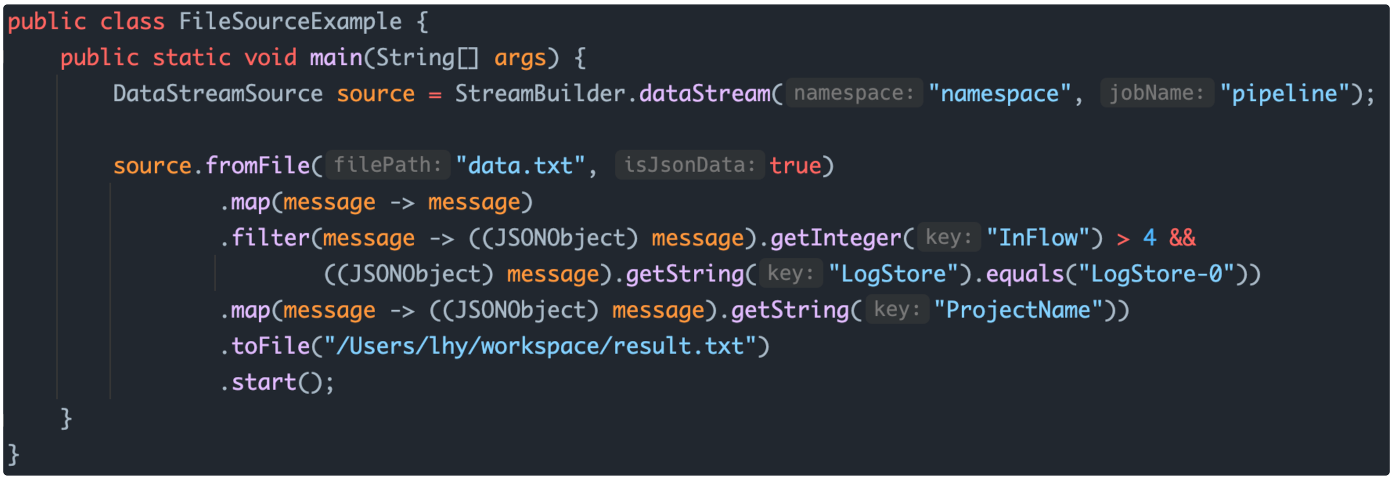
结果



```
~/workspace ❯ cat result.txt
{"InFlow":"7","ProjectName":"ProjectName-0","LogStore":"LogStore-0","OutFlow":"6"}
{"InFlow":"10","ProjectName":"ProjectName-0","LogStore":"LogStore-0","OutFlow":"9"}
~/workspace ❯
```

只输出ProjectName

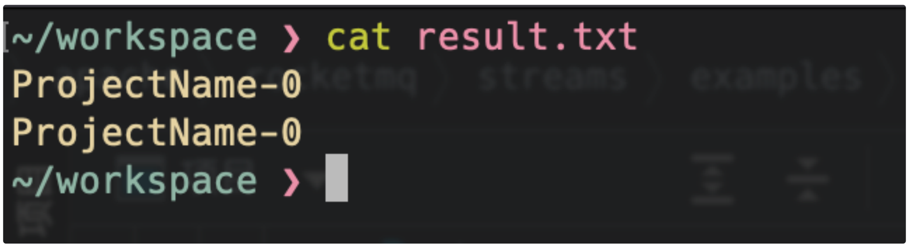
代码



```java
public class FileSourceExample {
    public static void main(String[] args) {
        DataStreamSource source = StreamBuilder.dataStream( namespace: "namespace",  jobName: "pipeline");

        source.fromFile( filePath: "data.txt",  isJsonData: true)
                .map(message -> message)
                .filter(message -> ((JSONObject) message).getInteger( key: "InFlow") > 4 &&
                        ((JSONObject) message).getString( key: "LogStore").equals("LogStore-0"))
                .map(message -> ((JSONObject) message).getString( key: "ProjectName"))
                .toFile("/Users/lhy/workspace/result.txt")
                .start();
    }
}
```

结果



```
~/workspace ❯ cat result.txt
ProjectName-0
ProjectName-0
~/workspace ❯
```

思考题

> 如果消息处理需要基于**事件时间**进行处理，那么对于乱序的消息窗口需要添加什么额外的设计?

我认为，可以对每一个消息窗口，在增加对真实发生时间的范围限制，`[earleastOccurTime ~ latestOccurTime]`。然后再根据发生时间进行排序。