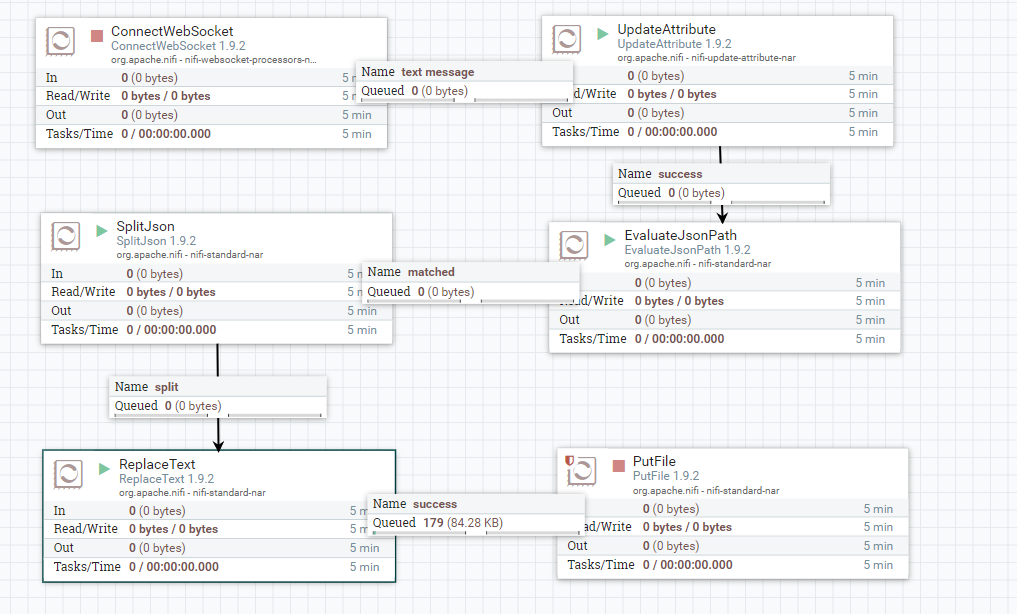
**Lab 2**

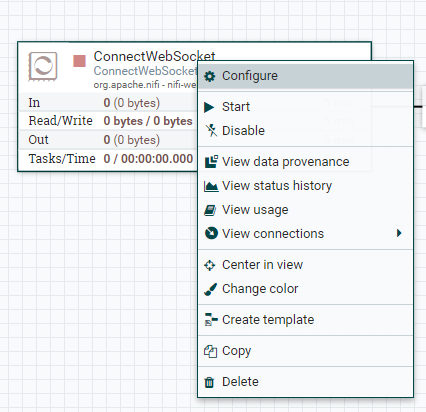
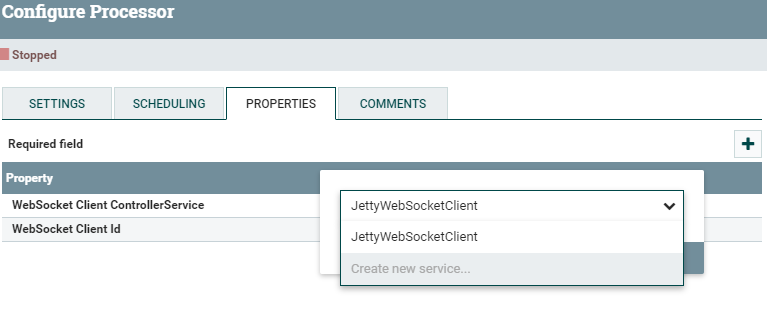
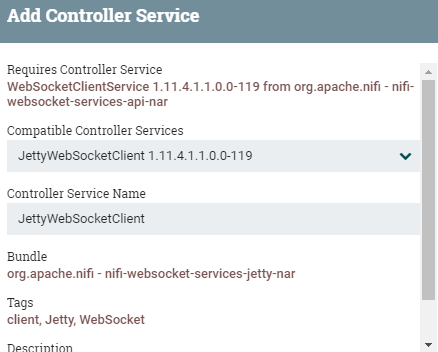
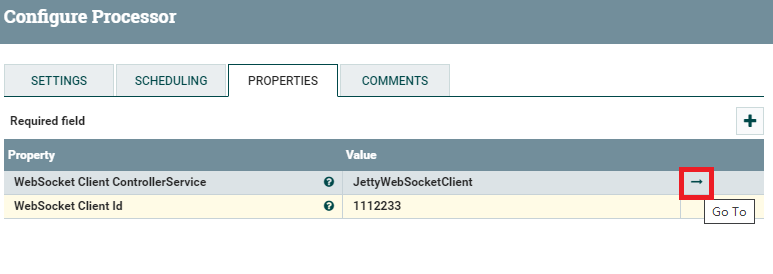
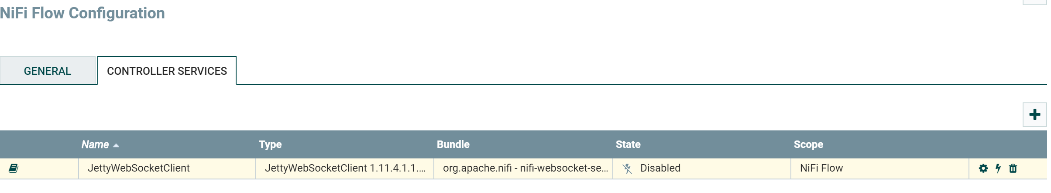
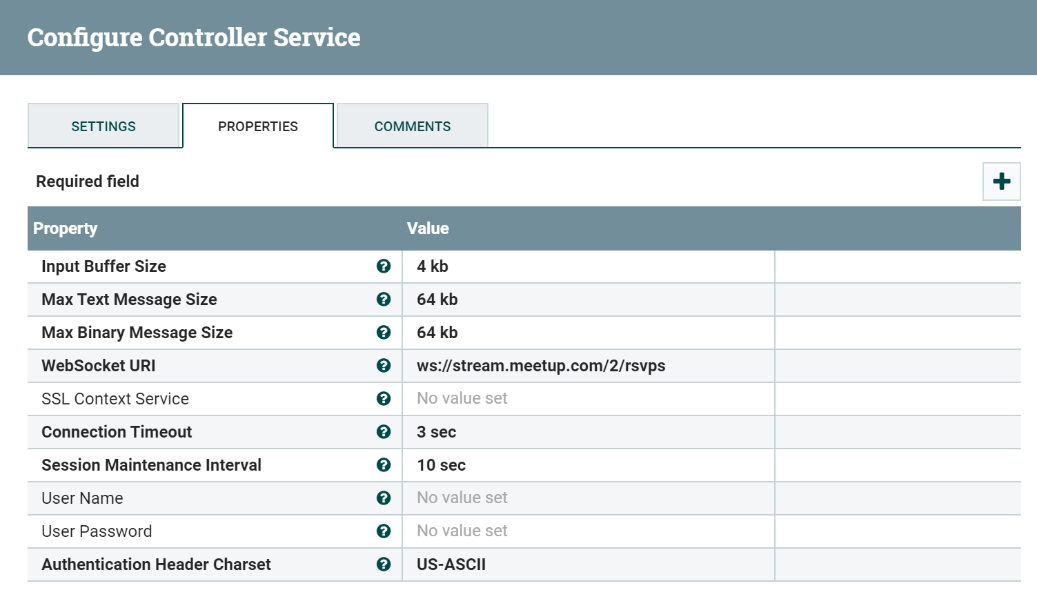
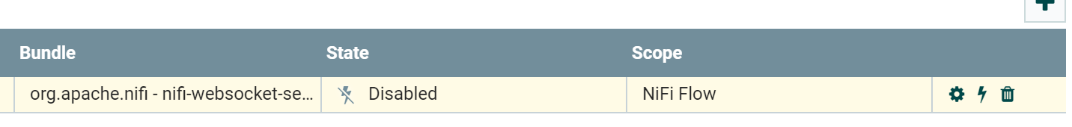
在本次实验中，我们将学到:

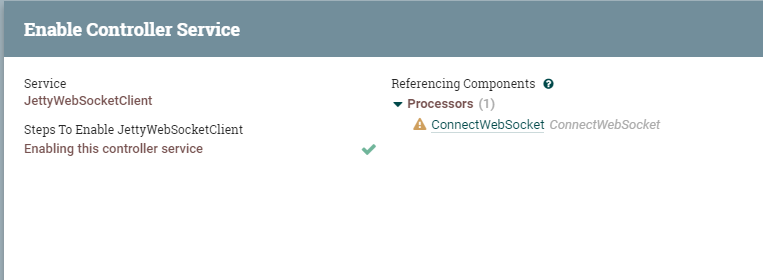
* 订阅Meetup网站的消息通知流Consume the Meetup RSVP stream
* 将消息的Json文本中我们需要的字段或对象提取出来
* 将提取后的Json文本拆分成更小的文件
* 将文件写入NIFI所在的系统存储上

**订阅Meetup RSVP数据**

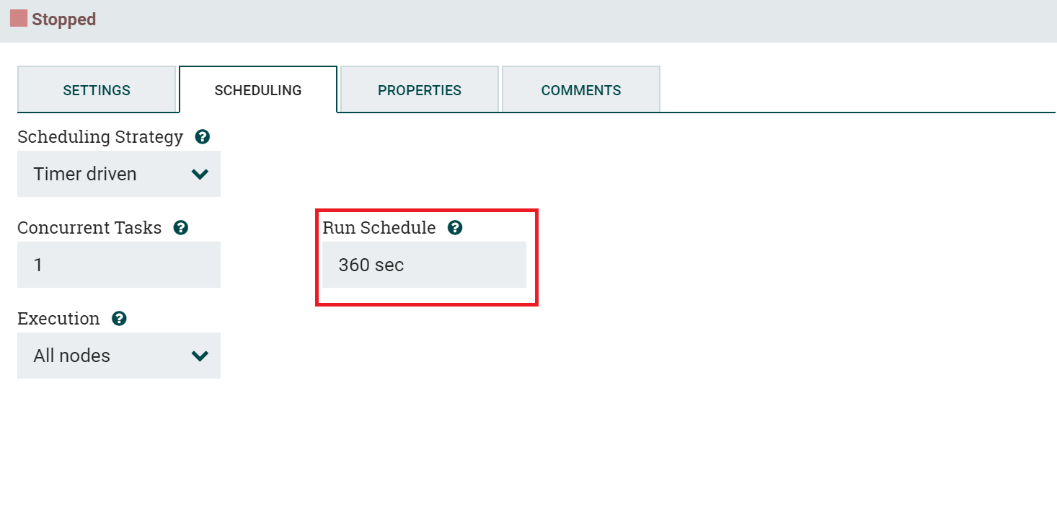
本次实验的最终成果应该如下Our final flow for this lab will look like the following:



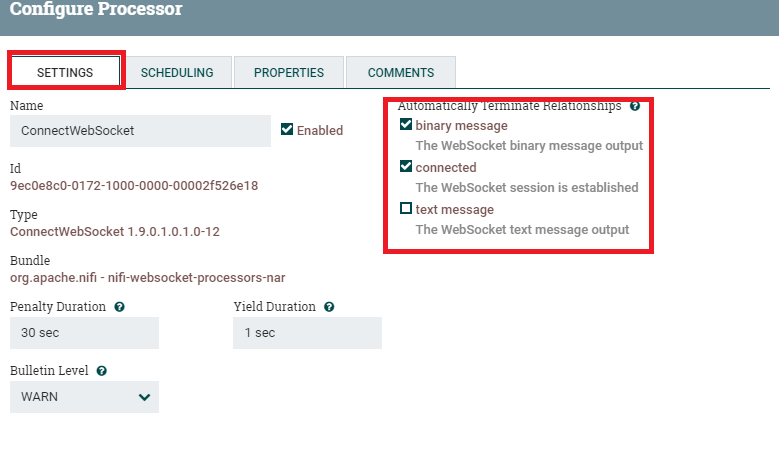
* 步骤1: 向画布上添加一个ConnectWebSocket 处理器
  + 在该处理器框内右键点击并选择Configure
  + 
  + 切换至Properties页面，并点击WebSocket Client Controller Service 然后选择Create New Service
  + 
  + 
  + 选择新建一个WebSocketClient服务，具体版本根据实验环境不同可能有差异
  + 点击箭头符号打开服务配置页
  + 
  + 
  + 点击右侧的齿轮按钮
  + 
  + 切换至Properties页面，在WebSocket URI对应的Value栏中输入如下地址ws://stream.meetup.com/2/rsvps
  + 点击闪电符号，启用该socket服务



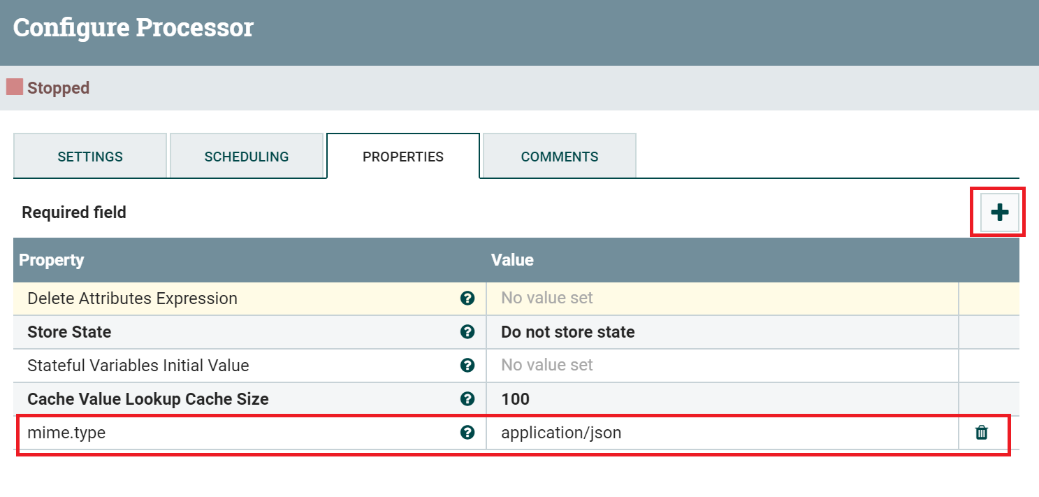
* + 返回处理器页面，再次打开该处理器的Properties页签，并设置一个任意的，随机的字符串作为Client ID
  + 切换至Scheduling页签并将 Run Schedule设置为360 sec



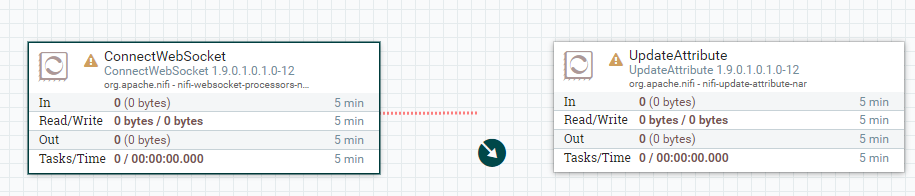
* + 切换至Settings页签并勾选自动丢弃的下游输出类型

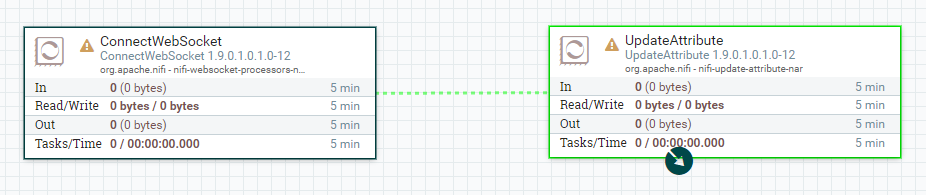


* 步骤 2: 添加一个Update Attribute 处理器
  + 打开Configuration页面

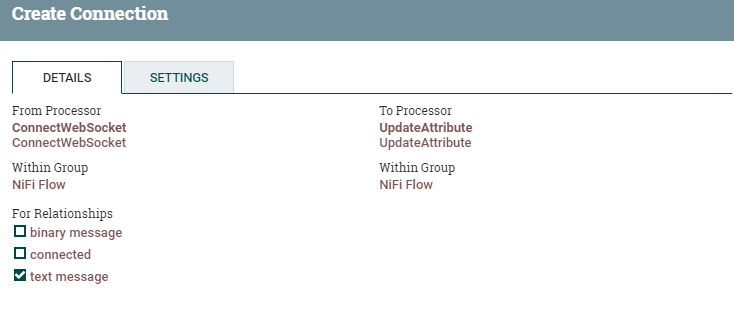


* + 点击+ 符号并在新增的一栏中填写property 为mime.type，value为application/json
  + 将Websocket处理器与Update Attribute 处理器链接起来

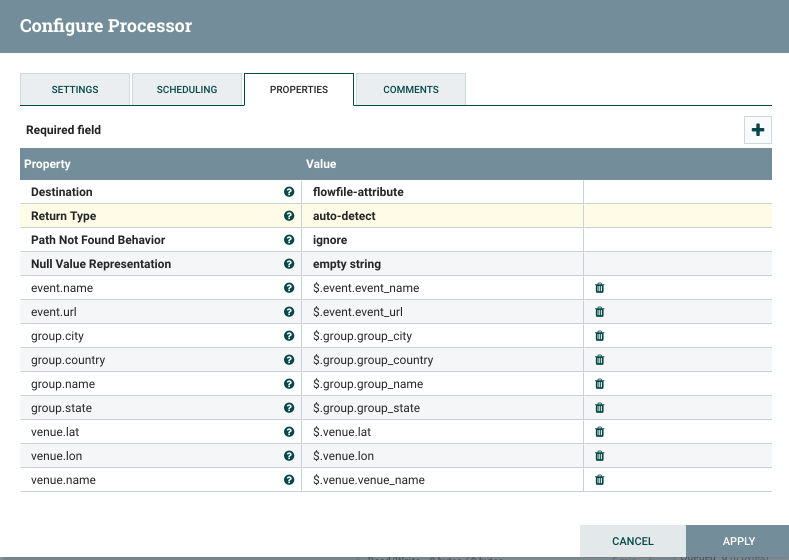




* + 勾选Text Message 作为上游输出类型



* 步骤 3. 添加一个EvaluateJsonPath 处理器
  + 打开Properties页面，并按下图进行配置, 注意如从该文档复制则需要注意不要复制空格

[](https://github.com/apsaltis/HDF-Workshop/raw/master/jsonpath.png)

需要添加的属性和值为：

event.name $.event.event\_name

event.url $.event.event\_url

group.city $.group.group\_city

group.state $.group.group\_state

group.country $.group.group\_country

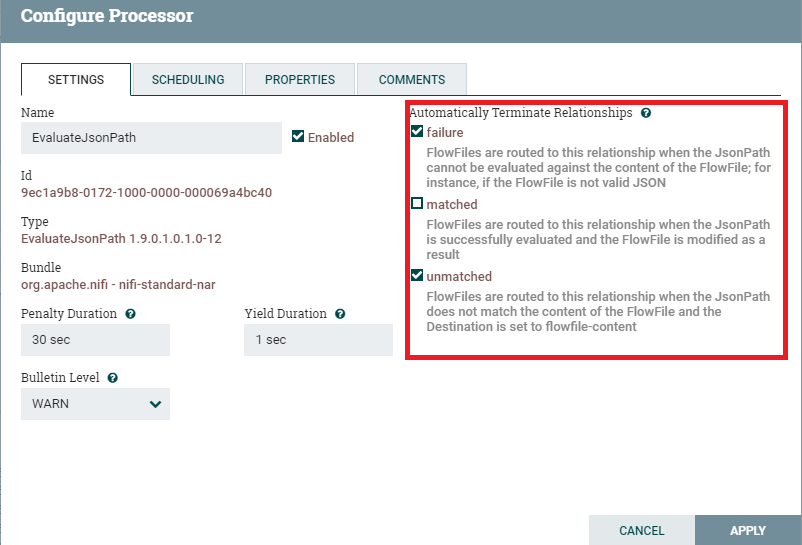
group.name $.group.group\_name

venue.lat $.venue.lat

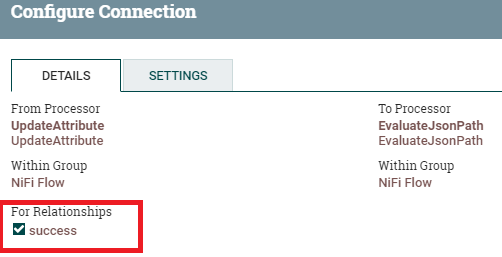
venue.lon $.venue.lon

venue.name $.venue.venue\_name

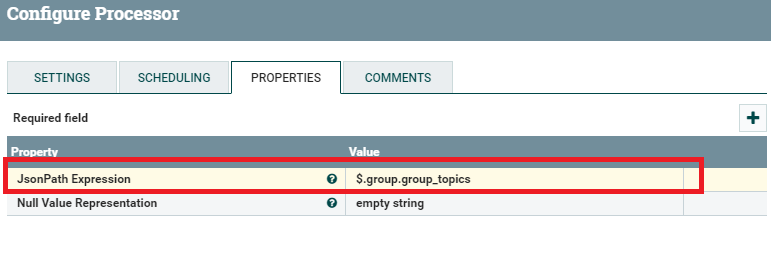
* + 切换至Settings页签，勾选failure和unmatched为自动丢弃的下游输出类型



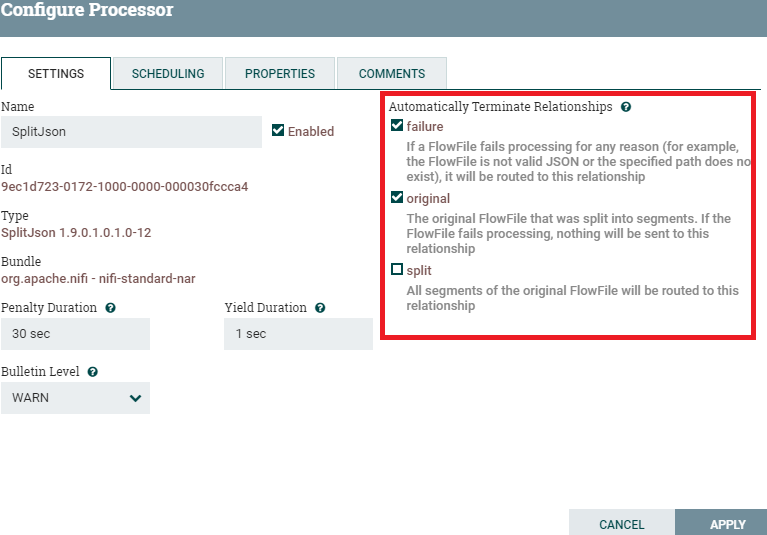
* + 将Update Attribute 处理器与EvaluateJsonPath处理器链接起来，并选择链接关系为success



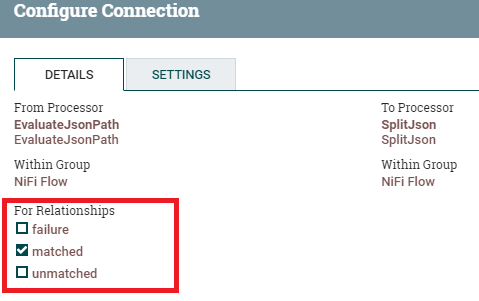
* 步骤4: 添加SplitJson 处理器
  + 打开Properties页面，并配置JsonPath Expression 为  $.group.group\_topics



* + 切换至Settings页签，勾选failure和original为自动丢弃的下游输出类型



将Evaluate JsonPath处理器与SplitJson链接起来，并选择链接关系为Matched



* 步骤5: 添加一个ReplaceText 处理器
  + 打开Configuration页面并配置Search Value 为([{])([\S\s]+)([}])  Replacement Value 为

{

"event\_name": "${event.name}",

"event\_url": "${event.url}",

"venue" : {

"lat": "${venue.lat}",

"lon": "${venue.lon}",

"name": "${venue.name}"

},

"group" : {

"group\_city" : "${group.city}",

"group\_country" : "${group.country}",

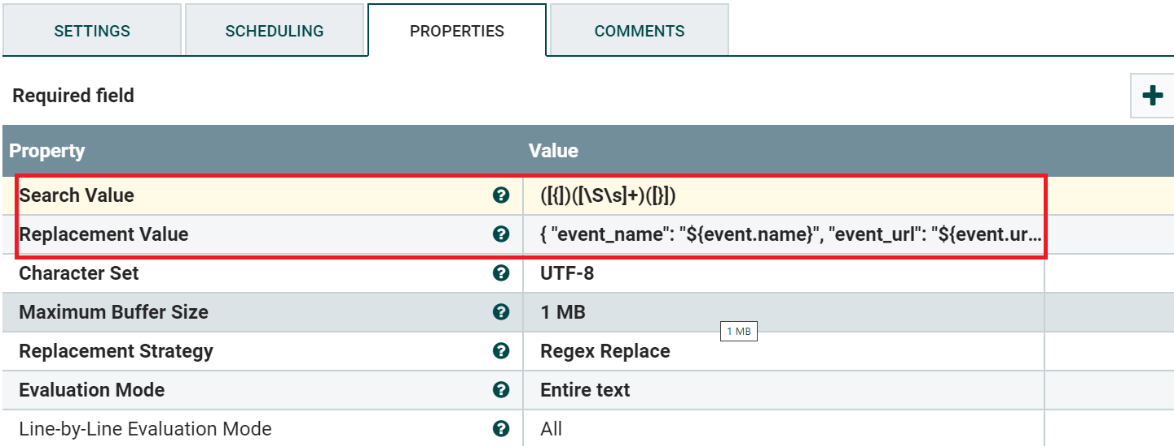
"group\_name" : "${group.name}",

"group\_state" : "${group.state}",

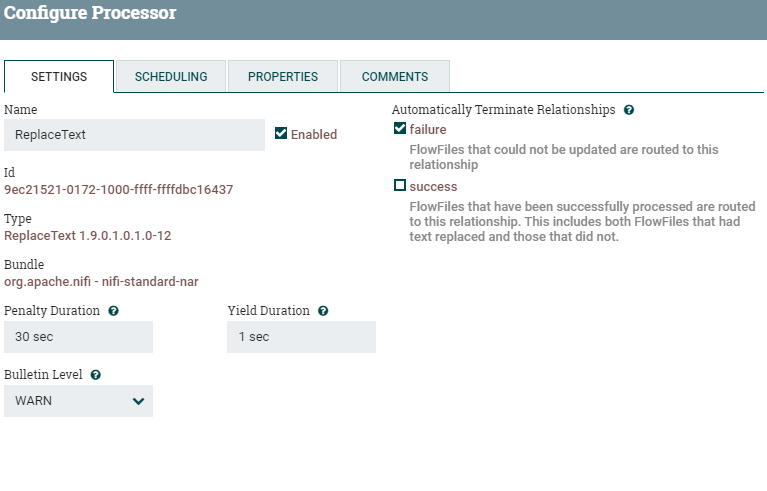
$2

}

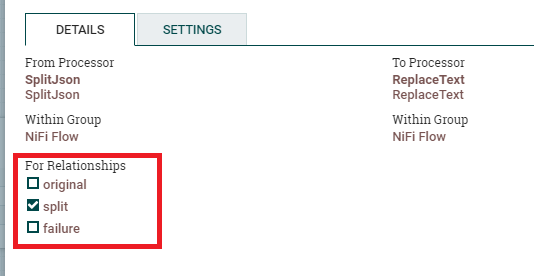
}



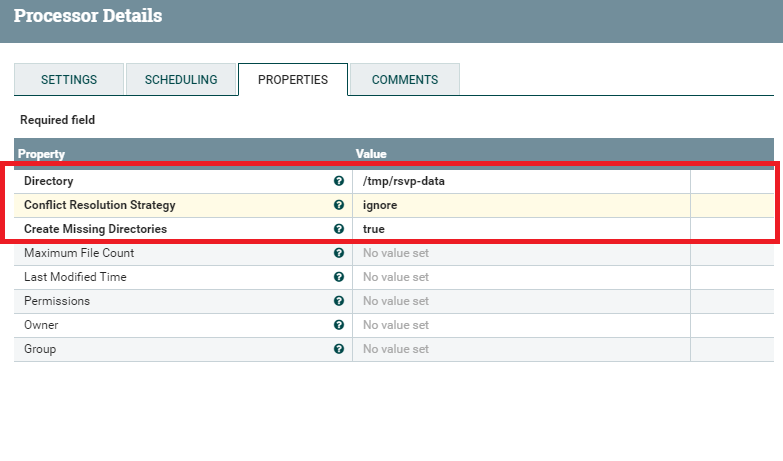
切换至Setting页签，勾选failure为自动丢弃的下游输出类型



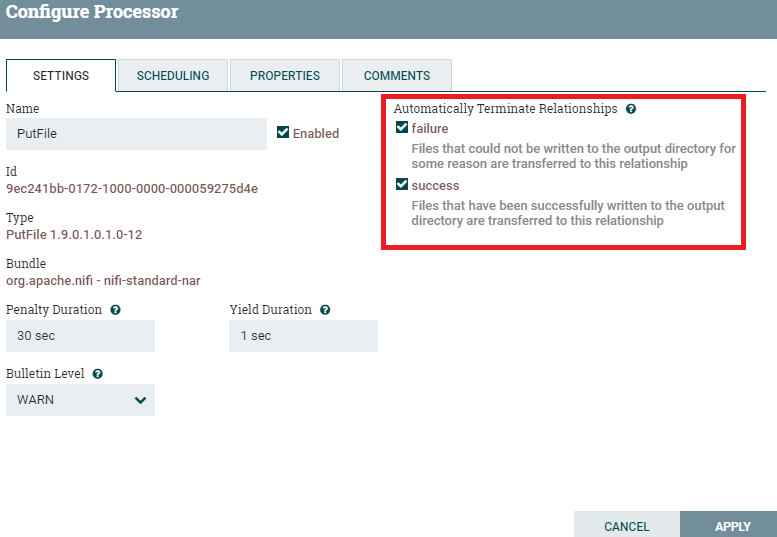
将SplitJson 与ReplaceText链接起来，并选择链接关系为split



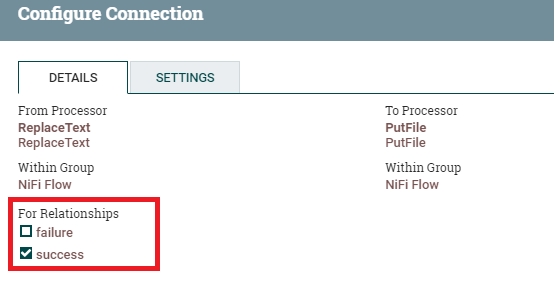
* 步骤6: 添加PutFile 处理器
  + 打开Properties页面，将如下属性的值设置为
    - Directory ： /tmp/rsvp-data
    - Conflict Resolution Strategy ： Ignore



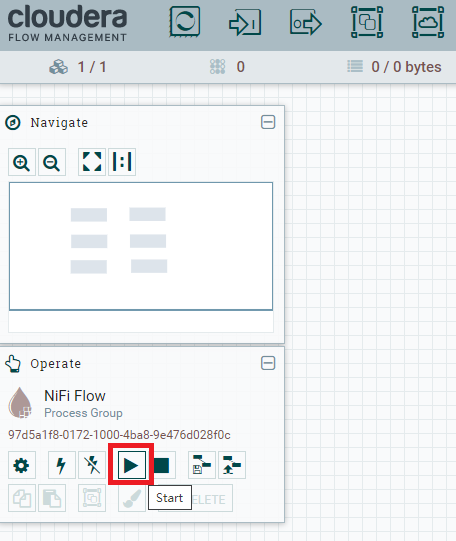
* + 切换至Settings页签，勾选failure和successful为自动丢弃的下游输出类型



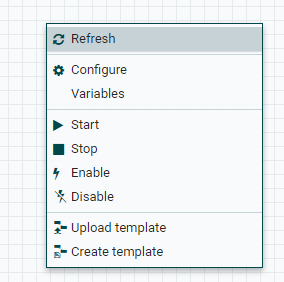
* + 将Replace Text处理器与PutFile处理器链接起来，并选择链接关系为Success



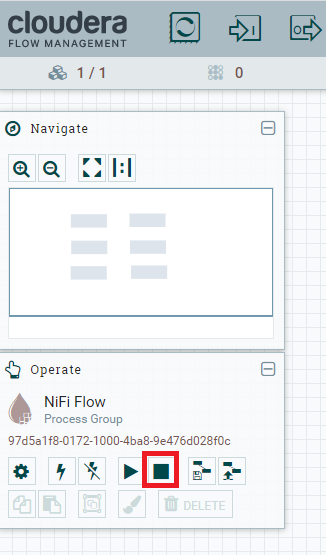
* 步骤7: 启动这个数据流



* + 如果需要尽快看到数据流运行情况，可以右键点击画布空白处并点击Refresh

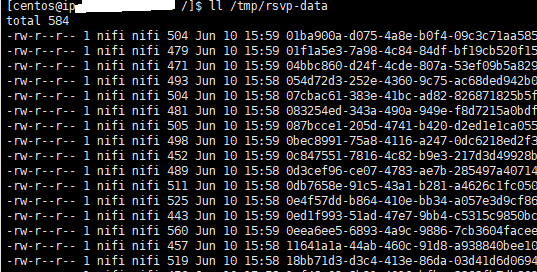


* + 执行一段时间后，可以停止该数据流



步骤8: 通过SSH客户端登陆你的系统，并查看对应目录是否有新的数据文件

* + ll /tmp/rsvp-data



* 本次Lab 2实验到此完成

**思考问题**

1. RSVP Json 对象在消费时该如何查看?
2. 最后的输出有多少个文件?
3. 能否变更PutFile处理器写入磁盘的文件名生成方式？
4. 你能否改变你设计的Dataflow并从Json中获取图片的地址并下载图片？
5. 你能否改变你的设计并将你喜欢的主题和图片通过邮件发送给其他人？