Test Report 2

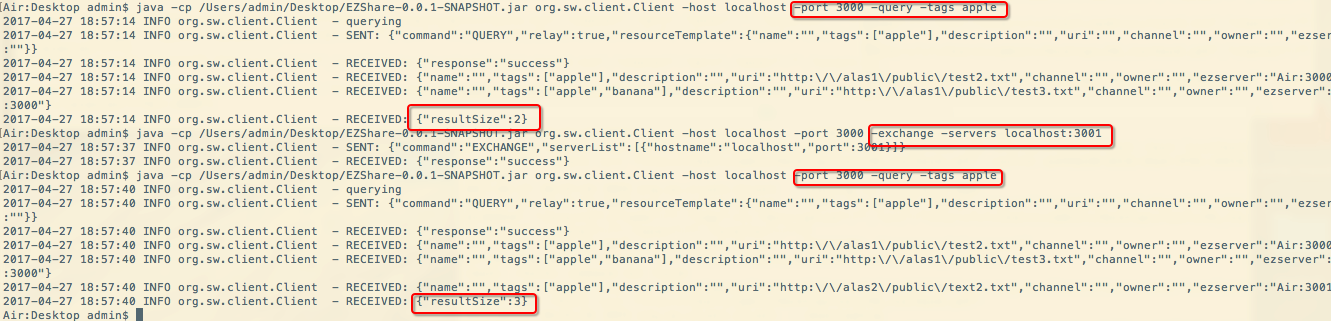
Nectar今天不知道干嘛一直跑不了，我用本机两个server测的。

Problem:

* Query：下图中，中间上方那个小图是server2中我publish的两个文件，第一个文件,tag🡪apple， 第二个文件 tag->apple,banana。 中间下方的执行是client在server1中query –tags apple, 红色部分是server1中本身的文件，绿色部分是server2中的文件，但是结果应该有两个server2中的文件才对，目前只收到一个。



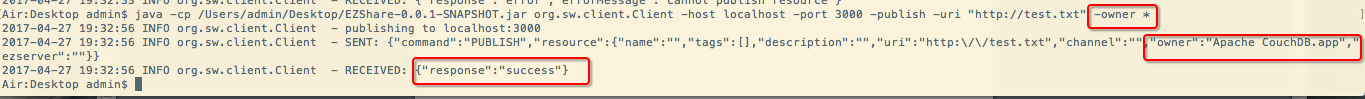
* Query: 下图一开始Query结果是2，Exchange之后结果是3。但是在一开始之前我已经Exchange过，result结果已经是3了，可能过了几分钟再去测结果变成2需要再次Exchange。两个server都没有宕机。分析可能是server端自主exchange过程有问题，将serverlist还原。



我测试过了，就是自主exchange过程有问题，还原了。 调了-exchangeinterval = 10s, 然后10s后就变了。理论上不宕机serverList应该不变：



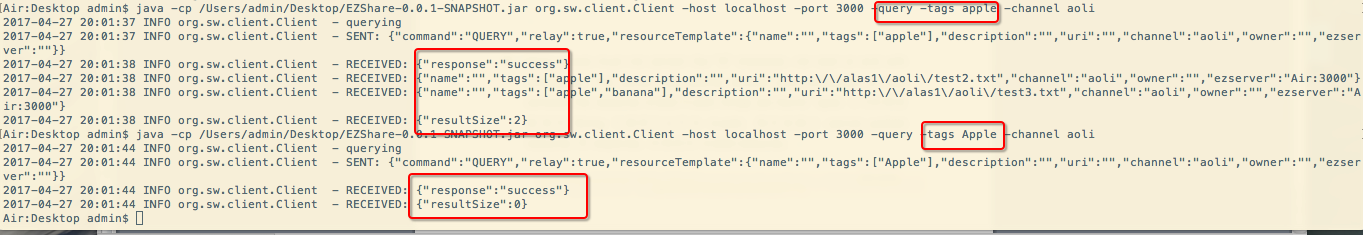
* Server 端的 –connectionintervallimit 我重设成5s后，client依然可以5s内多次重复访问。spec上说“An incoming request that violates this rule will be closed immediately with no response”
* Publish: 设置owner 为 \* ， 理论上应该是invalid,但是现在是success, 并且将owner改成 Apache CouchDB.app.



* Share 的时候如果uri不是绝对路径，我觉得应该输出 invalid resource, “If the resource contained incorrect information that could not be recovered from”, “cannot share resource”是除了incorrect information之外的情况



* Query: tags 的检索是 case insensitive!!!! 目前的程序是 case sensitive。



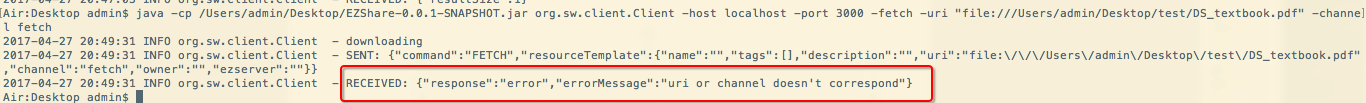
* Query 的 name 和 description的rules写的有问题。



四种情况对于name 和 description是正确的：

1. 第一步先判断某一条resource是否name 和 description都为空串，若均为空串，满足条件。
2. 若第一步条件不满足，判断query 中是否name 和 description两部分均含有。如果均含有，分别判断每条resource 中的name 和 description 是不是含有query 中的 name 和 description的子串。
3. 若第二步条件不满足，判断query中是否只有name，模板的name一定要是resource的子串，description此时随意。
4. 若第三步条件不满足，判断query中是否只有description，模板的description一定要是resource的子串，name此时随意。

* Fetch: 当channel 或者 uri 不匹配的时候，不用像老师那样输出invalid resourceTemplate吗？我觉得老师那个情况就是说的不匹配。



Question(这几个问题你wechat上跟我说一下)

* 你程序运行内存设置的是多少，让Lilian在Report “Other Challenges”中写出来，并且说明如果不设置Chunk,当文件大于内存的时候，内存溢出。
* 你哪几个方法用了synchronize修饰，我觉得只有publish/share/remove三个需要，query & fetch 一般都是shared lock，可以同时进行的。
* 你在自主Exchange的时候是每次随机选择一个server交换信息不是选择server List 上全部server exchange 吧？

To Lilian:

For some optional problem 🡪 “Sting values must not contain the “\0” character, nor start or end with whitespace. The server may silently remove such characters or may consider the resource invalid if such things are found.” In this case, our system chose to silently remove.