# 性能测试脚本录制注意事项

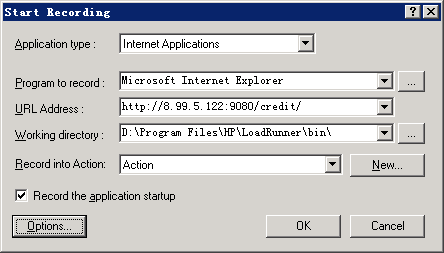
## 命名规范

建议使用如下格式——“XDXT01\_TX01\_登录”、“XDXT01\_TX02\_待办展示”、“XDXT01\_TX03\_退出” ——命名事务，这样在进行数据整理的时候可以使用Excel的排序功能，减少低效的“体力劳动”；

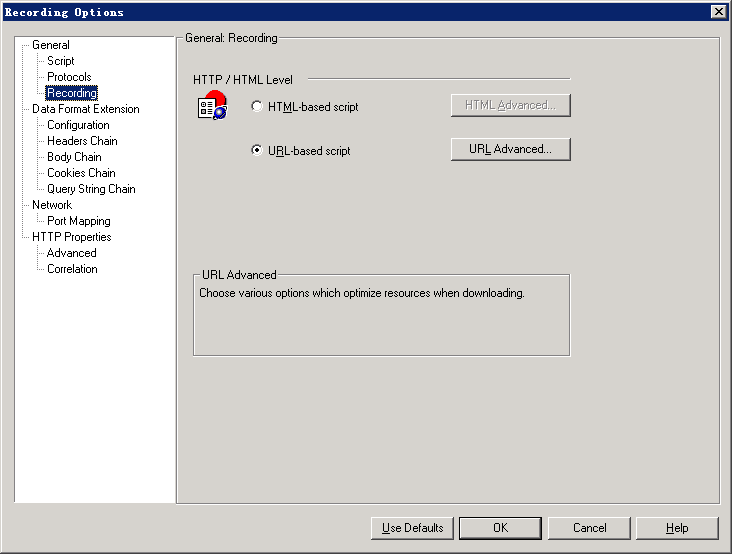
## 录制级别设置

由于系统中有多处Redirector，录制脚本时需设置录制级别为URL-based script，否则会有部分脚本执行出错。操作步骤为：

* 1. 在开始录制对话框点击“Options...”打开录制选项窗口；



* 1. 从左侧列表里点击“General”|“Recording”，在右侧面板里选择URL-based script。



## 脚本的参数化和关联

### 参数化的概念及应用

参数化主要是给HTTP请求中的静态参数进行赋值。

为模拟不同用户的操作，需要对LoginID、密码、用户名、用户姓名、用户所属机构号等进行参数化。测试环境里有很多用户的密码都是000000als，因此可以选择密码为000000als用户的相关信息来参数化，由于同一LoginID有可能对应多个UserID，为简化工作，同时还要求该LoingID对应的UserID唯一，因此构建如下的SQL来选择需要参数化的LoginID：

Select LoginID

From (Select Count(1) As UserCount, LoginID

From NBCBCREDIT.User\_Info

where Password = 'CC357616CC513B4896F8FED824B3E191'

Group By Loginid)

Where UserCount = 1;

具体到不同的事务，还要求该用户对应业务列表不为空，因此还需要根据具体的业务构建业务表的查询，对LoginID进行筛选。

**参数化脚本需要深入了解系统的数据结构、业务规则。**

### 关联的概念及应用

关联是指从动态的HTTP响应中抓取后续HTTP请求所需的参数，并在后续的HTTP请求中替换脚本录制时所使用的参数。

#### 自动关联

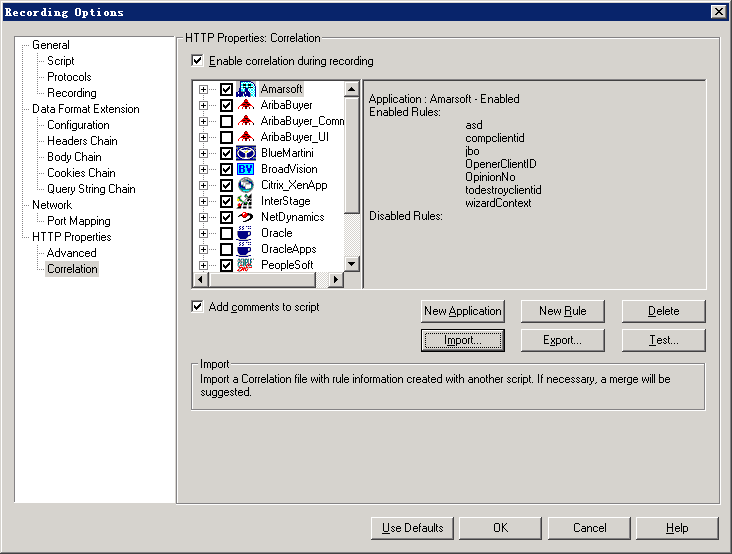
LR可以实现自动关联，在录制并生成脚本的过程中自动分析HTTP响应和请求并关联符合关联规则的HTTP请求参数。ALS7.cor文件配置了在ALS7中常用的几条关联规则，关联规则略举一例：

<Rule Name="compclientid" LeftBoundText="sCompClientID=&quot;" LeftBoundType="1" LeftBoundInstance="0" RightBoundText="&quot;" RightBoundType="1" AltRightBoundText="" AltRightBoundType="1" Flags="8" ParamPrefix="CompClientID\_" Type="8" SaveOffset="0" SaveLen="-1" CallbackName="" CallbackDLLName="" FormField="" ReplaceLB="" ReplaceRB=""/>

这条规则指示LR，在录制并生成脚本的过程中，当HTTP响应中有以“sCompClientID=””为左边界、以“””为右边界的内容时，将被**根据需要**抓取出来作为后续HTTP请求的参数，被抓取出来的参数以“CompClientID\_”为前缀命名，例如当HTTP响应中有“sCompClientID=” C2013062115484424966094174279312””的内容时，*有可能*会在该条HTTP请求之前自动生成相应的语句，抓取 “C2013062115484424966094174279312”并赋值给以“CompClientID\_”为前缀命名的变量，假设该变量被命名为“CompClientID\_35”，则后续脚本中出现“C2013062115484424966094174279312”的地方都会以“{CompClientID\_35}”替换。

之所以说是“有可能”是因为，若经过分析，在录制的脚本里，“C2013062115484424966094174279312”没有在后续的HTTP请求中使用，则不会自动生成上述语句。

要实现自动关联，只要在HP Virtual User Generator的录制选项中导入关联文件ALS7.cor。操作步骤为：在开始录制对话框点击打开录制选项窗口，从左侧列表里点击“HTTP Properties”/“Correlation”，在右侧面板里点击“Import...”，导入上述文件即可。



#### 用户相关参数的自动关联技巧

透过对系统数据结构、业务规则的深入了解，有些HTTP请求参数的赋值可以通过参数化，也可以通过脚本关联实现，但这两种方式的难易程度不同。下面以用户相关参数为例说明如何通过一定的技巧来减少参数化的工作量，降低难度，提升效率。

除了LoginID外，UserID、UserName以及OrgID也会作为HTTP请求的参数，对于系统数据结构了解的话可以通过简单的数据库查询获得对应的UserID、OrgID、UserName对脚本进行参数化，其中UserName参数化前所需的两次URL编码，可使用Excel自定义函数进行转换；但这一过程涉及到数据准备以及每个参数的设置、赋值、取值方式设置和取值时机设置等操作，整个过程繁琐且容易出错。

若在LR里导入关联文件PTComm.cor，并在性能测试环境的MainAppEntry.jsp文件的HEAD段添加如下代码，则可以实现UserID、UserName和OrgID的自动关联，减少参数化的工作量，*该段代码给后台带来的额外负担可忽略不计*。

<script type="text/javascript">

//PTVar001='<%=CurUser.getOrgID()%>';

//PTVar002='<%=CurUser.getUserID()%>';

//PTVar003='<%=CurUser.getUserName()%>';

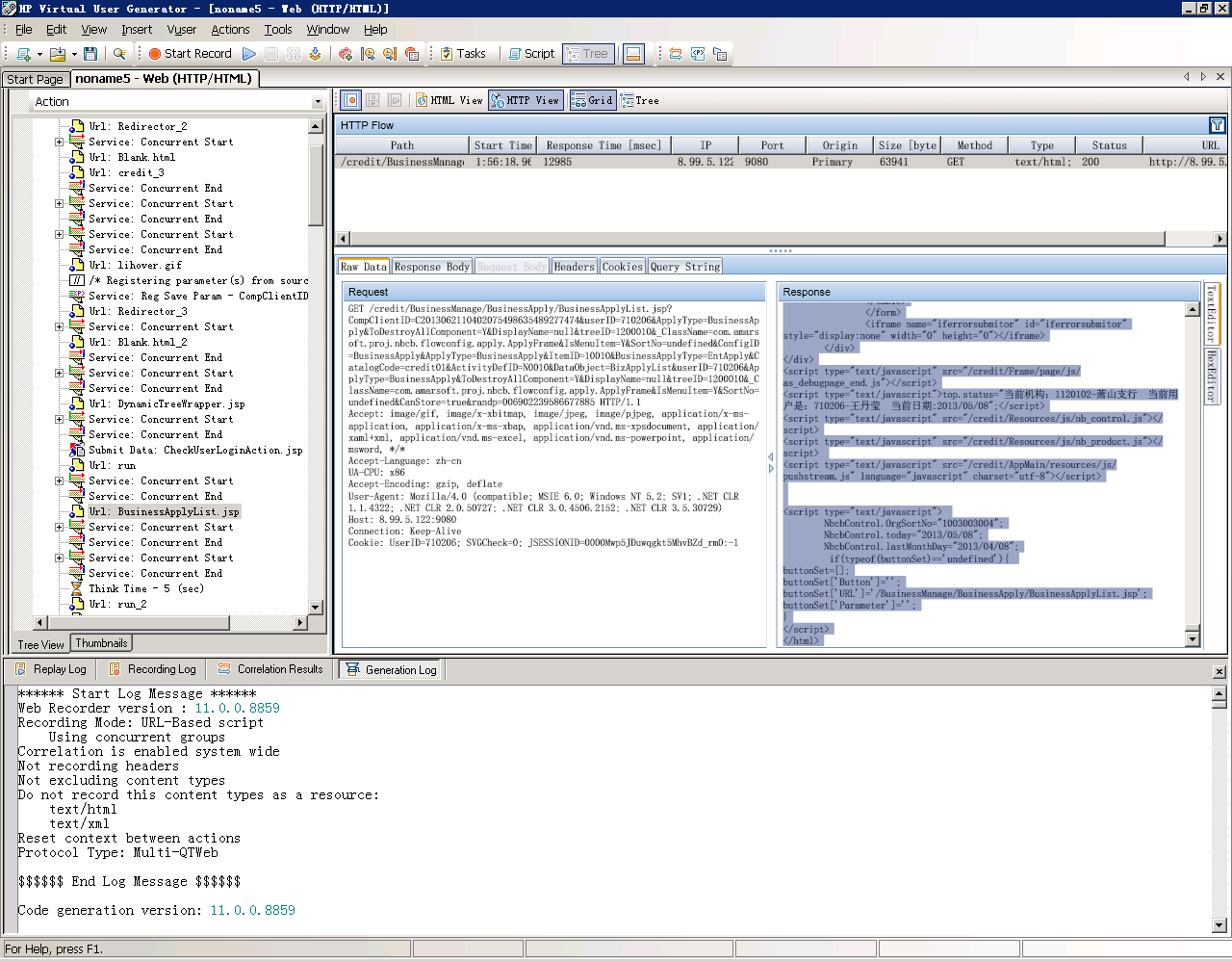
//PTVar103='<%=java.net.URLEncoder.encode(java.net.URLEncoder.encode(CurUser.getUserName(),"UTF-8"),"UTF-8")%>';

</script>

#### 手工关联实例

具体业务列表下的进一步事务操作如查看业务详情、签署意见等还需要关联到序列号、第三方应用的事务参数等，需要具体问题具体分析，下面以申请业务的意见签署为例演示一下如何进行手工关联：

* 1. 录制脚本，注意录制到签署意见时选择列表里的第一条记录，录制完生成脚本后切换到HP Virtual User Generator的Tree视图；



* 1. 在Tree视图点击列表页面，这里是“Url: BusinessApplyList.jsp”，从右边的面板选择HTTP View，为操作方便，全选Response的内容并复制粘贴到文本编辑器中，另存为BusinessApplyList\_Snapshot.txt；
  2. 忽略其后的“Service: Concurrent Start”块和“Service: Concurrent End”语句，找到以“Url:”开头的HTTP请求；以“Url: run\_2”为例，该请求的URL为“http://8.99.5.122:9080/credit/servlet/run?ClassName=com.amarsoft.proj.nbcb.businessmanage.businessapply.action.BusinessApplyCheckAction&MethodName=checkIsBaseInfoSave&Args=applyNo=AP06042007490999”，其中的“AP06042007490999”表示该请求涉及到的是列表中的第一条申请，可见该请求与当前选择的申请有关；
  3. 脚本关联的实质是从HTTP响应中抓取后续HTTP请求的参数，因此我们在前面保存的BusinessApplyList\_Snapshot.txt文件中找到“AP06042007490999”并分析得出其抓取规则；
  4. 在BusinessApplyList\_Snapshot.txt文件中搜索“AP06042007490999”，得到如下一行内容；

DZ[i][2][0]=new Array('AP06042007490999','--','63611','BA06042007490999','FO2013060400007207','EntApply','80457','','63611','N0010','02','zygu对公客户1005','01','CI2013031300000014','101','2012082200000008','一般中长期贷款','2','01','','10,000','0010','业务经办','A00045','萧山支行信贷经理','1120102','萧山支行');

* 1. 由此分析出APPLYNO的抓取规则为：左边界“DZ[i][2][0]=new Array('”，右边界“,”，在脚本中“Url: BusinessApplyList.jsp”前插入下面的关联语句，查找并替换后续HTTP请求中所有的“AP06042007490999”为“{APPLYNO}”；

web\_reg\_save\_param("APPLYNO",

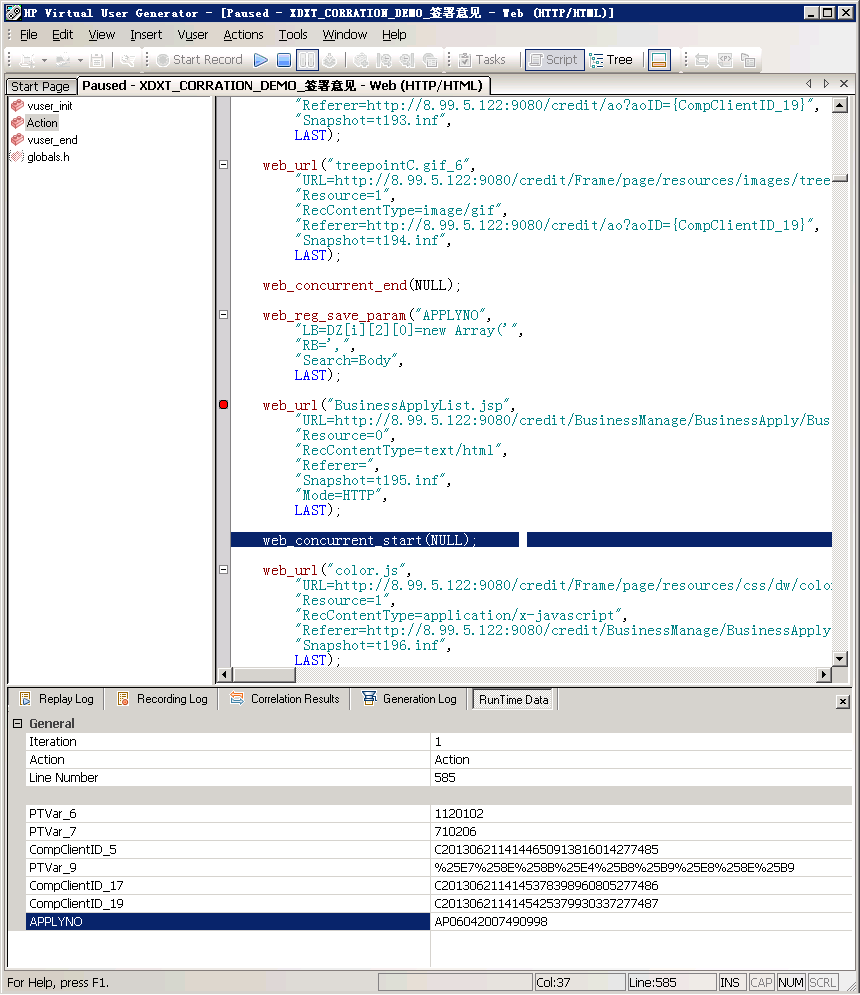
"LB=DZ[i][2][0]=new Array('",

"RB=',",

"Search=Body",

LAST);

* 1. 设置断点执行脚本，从下图可见确实抓取到了正确的APPNO并替换后续HTTP请求的参数；



* 1. 再次分析后续的HTTP请求，发现还需要关联“workItemID =80457”，该参数值在BusinessApplyList\_Snapshot.txt文件中与APPNO同一行，由于其抓取规则的左边界中含动态内容，关联语句需要使用正则表达式，其关联语句如下，查找并替换后续HTTP请求中所有的“80457”为“{WORKITEMID }”；

web\_reg\_save\_param\_ex("ParamName=WORKITEMNO",

"LB/RE=DZ\\[i\\]\\[2\\]\\[0\\]=new Array\\('[^']+','[^']+','[^']+','[^']+','[^']+','[^']+','",

"RB=',",

LAST);

* 1. 类似地，可以通过正则表达式关联到其他参数，关联语句见下：

//Field 4

web\_reg\_save\_param\_ex("ParamName=OLDLIMITID",

"LB/RE=DZ\\[i\\]\\[2\\]\\[0\\]=new Array\\('[^']\*','[^']\*','[^']\*','",

"RB=',",

LAST);

//Field 7

web\_reg\_save\_param\_ex("ParamName=WORKITEMID",

"LB/RE=DZ\\[i\\]\\[2\\]\\[0\\]=new Array\\('[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','",

"RB=',",

LAST);

//Field 14

web\_reg\_save\_param\_ex("ParamName=CUSTOMERID",

"LB/RE=DZ\\[i\\]\\[2\\]\\[0\\]=new Array\\('[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','",

"RB=',",

LAST);

//Field 15

web\_reg\_save\_param\_ex("ParamName=OCCURTYPE",

"LB/RE=DZ\\[i\\]\\[2\\]\\[0\\]=new Array\\('[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','",

"RB=',",

LAST);

//Field 16

web\_reg\_save\_param\_ex("ParamName=PRODUCTID",

"LB/RE=DZ\\[i\\]\\[2\\]\\[0\\]=new Array\\('[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','[^']\*','",

"RB=',",

LAST);

* 1. 采用类似的方法分析后面的HTTP响应和HTTP请求，处理签署、提交意见所需要的其他参数；

web\_reg\_save\_param("PROCESSINSTID",

"LB/IC=processInstID=",

"RB/IC=,",

"Ord=1",

"Search=Body",

LAST);

web\_reg\_save\_param("BIZPROCESSOBJECTID",

"LB/IC=bizProcessObjectID=",

"RB/IC=\"",

"Ord=1",

"Search=Body",

LAST);

对于某些在当前脚本事务中恒定不变的参数如展示列表时的“ActivityDefID=N0010”、提交流程时的“activityDefID=N0020”、“opinionAction=02”、“personID=710204”等，可以不用关联；

必须使用web\_reg\_save\_param\_ex才能在关联时使用正则表达式作为边界条件，同时还要注意在正则表达式中，正常的“[”、“]”、“(”等字符必须使用转义符“\”进行转义，而转义符“\”在C语言中要写成“\\”。

#### 列表相关参数的手工关联的技巧

由于我们对列表的操作参数来自列表中记录的字段，在录制脚本时更是有意识地针对业务列表的第一条记录进行操作，因此可以通过对后台Jsp的改造来降低手工关联的复杂度。与用户相关参数的自动关联技巧类似，通过在后台Jsp中添加一段代码将第一条记录的所有字段使用更易于抓取的方式包装起来发送到前端，则在手工关联时，不论是关联脚本的难度、语句的复杂程度还是易出错性都得以极大地降低，并能提升关联语句的可读性和可维护性。该段代码可适用于大部分业务列表的Jsp，具体方法见下：

1. 在业务列表的jsp文件中找到

Vector vTemp = dwTemp.genHTMLDataWindow(type);

for(int i=0;i<vTemp.size();i++) out.print((String)vTemp.get(i));

1. 将for循环修改为：

for(int i=0;i<vTemp.size();i++){

out.print((String)vTemp.get(i));

String tmpStr = (String)vTemp.get(i);

int idxOfDZLine1Begin = tmpStr.indexOf("DZ[i][2][0]=new Array(");

if(idxOfDZLine1Begin>=0){

idxOfDZLine1Begin += 23;

int idxOfDZLine1End = tmpStr.indexOf("');", idxOfDZLine1Begin);

String[] DZLine1 = tmpStr.substring(idxOfDZLine1Begin, idxOfDZLine1End).split("','");

for(int idxOfDZField = 0; idxOfDZField < DZLine1.length; idxOfDZField++){

out.print(String.format("//PTField0%02d='", idxOfDZField) + DZLine1[idxOfDZField] + "';\n");

out.print(String.format("//PTField1%02d='", idxOfDZField) + java.net.URLEncoder.encode(java.net.URLEncoder.encode(DZLine1[idxOfDZField])) + "';\n");

}

}

}

1. 根据需要关联PTField0XX或PTField1XX，其中XX表示字段序号；
2. 上一节i)、j)标题下的关联语句可修改为如下内容，与上节的语句相比，其难度、语句复杂程度和易出错程度都降低了，而效率、可读性和可维护性都得到极大的提升。

//Field 1

web\_reg\_save\_param("APPLYNO",

"LB=PTField000='",

"RB=';",

"Search=Body",

LAST);

//Field 4

web\_reg\_save\_param("OLDLIMITID",

"LB=PTField003='",

"RB=';",

"Search=Body",

LAST);

//Field 7

web\_reg\_save\_param("WORKITEMID",

"LB=PTField006='",

"RB=';",

"Search=Body",

LAST);

//Field 14

web\_reg\_save\_param("CUSTOMERID",

"LB=PTField013='",

"RB=';",

"Search=Body",

LAST);

//Field 15

web\_reg\_save\_param("OCCURTYPE",

"LB=PTField014='",

"RB=';",

"Search=Body",

LAST);

//Field 16

web\_reg\_save\_param("PRODUCTID",

"LB=PTField015='",

"RB=';",

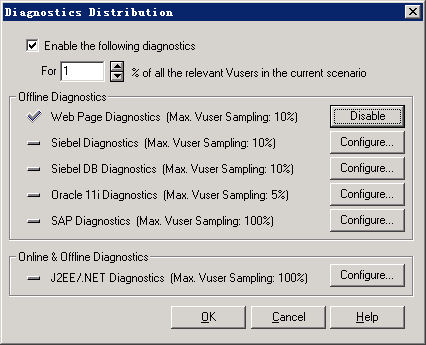
"Search=Body",

LAST);

1. 该段代码虽然给后台带来少量额外负担，但经实测验证，对事务响应时间的影响极其微小，推荐使用。

## 其他注意事项

* 由于LR对MMSG\_CREDIT消息服务采用同步机制，会明显拖慢事务响应时间，因此在测试过程中应将消息服务的web\_url请求语句删除。
* 由于HTTP请求中的参数数量巨大，在查找替换时要小心换错位置导致脚本执行失败。
* 使用Controller进行超过1小时的稳定性测试时需要从“Diagnostics”菜单的“Configuration...”禁用“Web Page Diagnostics”，如下图所示，点击“Web Page Diagnostics”右边的“Disable”进行禁用，否则1小时左右时会报大量错误并导致事务失败。



* 如果不忽略思考时间，则要注意将思考时间移到事务边界外。
* 建议录制完脚本后再根据实际发送的请求和执行的操作确定事务数和事务范围。
* TO BE ENRICHED