接口测试指导方案

1. 接口测试简介

接口测试是测试系统组件间接口的一种测试。接口测试主要用于检测外部系统与系统之间以及内部各个子系统之间的交互点。测试的重点是要检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。

**2. 接口测试流程**

接口测试的流程和功能测试流程类似，依据的对象是需求说明书和接口需求，接口测试流程如下：



3. 接口测试范围

1) 业务功能（包括正常、异常场景是否实现）

2) 业务规则（覆盖度是否全面）

3) 参数验证（边界、业务规则是否达到要求）

4) 异常场景（重复提交、并发提交、事务中断、多机环境、大数据量测试）

5) 性能测试（响应时间、吞吐量、并发数、资源要求）

6) 安全测试（权限验证、SQL注入等）

4. 接口测试重点

（1） 检查接口的功能：检查接口的功能有没有实现，也就是请求会不会成功，如果不成功会不会返回错误代号（或错误信息）

（2） 检查接口返回的数据：检查接口返回的数据、数据格式、数据类型是否与预期一致（正向且传递的参数正常）；

（3） 检查接口的容错性：接口是否可以正常处理（假如传递的参数足够大或者为负、空值时）

（4） 检查接口的性能：http请求接口大多与后端执行的SQL语句性能、算法等比较相关。

（5） 检查接口的安全性：外部调用的接口尤为重要。

**5. 接口测试需求分析**

首先根据接口设计的技术架构方案，了解清楚被测接口对应的公共入参、入参、出参及返回数据的Json结构规范，根据测试场景进行测试。

1) 理解接口参数，熟悉接口参数的输入要求、输入值范围、必填项等；

2) 理解接口输出，熟悉返回json的结构构成、返回值类别、返回值范围、返回data的不同类型等。

3) 理解接口的逻辑、接口的业务关联，熟悉技术方案中的接口相互关联、依赖的关系，接口与接口之间的数据传递等。

4) 寻找测试点，根据输入(参数名、取值范围)、输出(参数名、返回值范围)、关联关系，进行测试点分析；

6. 接口测试用例设计

接口测试的主要测试对象是接口，但是随着系统复杂度越来越高，接口越来越多，完全覆盖所有接口是很难的一件事情，并且实际过程中任意内部接口的变动都可能导致我们的测试用例的不可用。

1） 接口用例设计优先级

n 优先级-->针对所有接口

a) 暴露在外面的接口，因为通常该接口会给第三方调用；

b) 供系统内部调用的核心功能接口；

c) 供系统内部调用非核心功能接口；

n 优先级-->针对单个接口

a) 正向用例优先测试，逆向（异常）用例次之(通常情况，非绝对)；

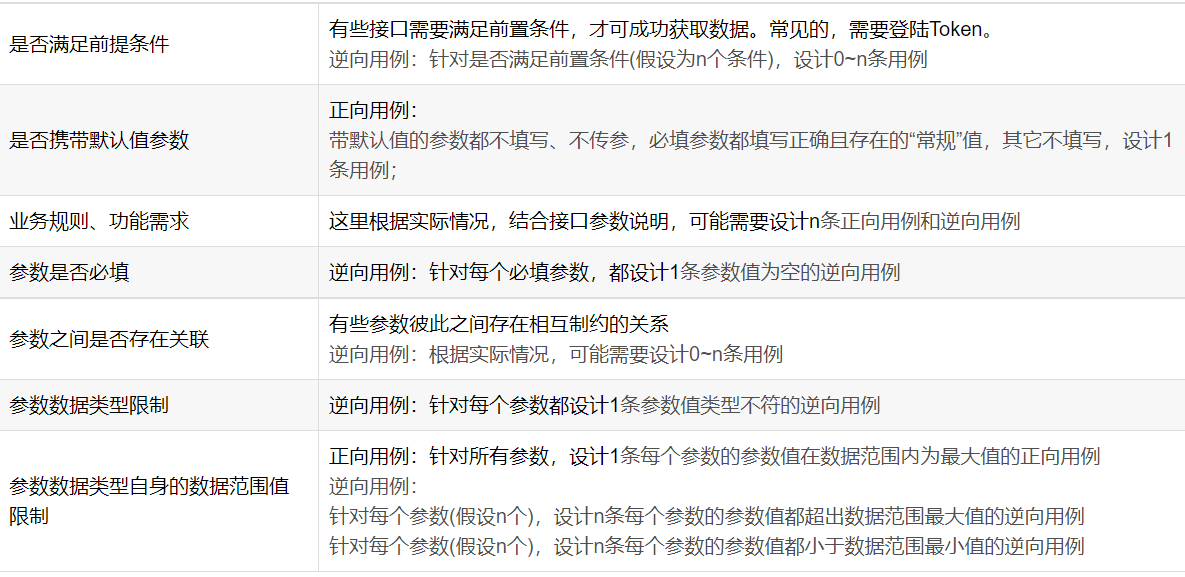
b) 是否满足前提条件 > 是否携带默认参值参数 > 参数是否必填 > 参数之间是否存在关联 > 参数数据类型限制 > 参数数据类型自身的数据范围值限制

2） 接口用例设计方法



具体可参考《接口测试用例设计》

3） 测试用例编写注意事项



进行测试执行编写时，有如下的原则：

1.不同的接口参数覆盖不同的业务场景；

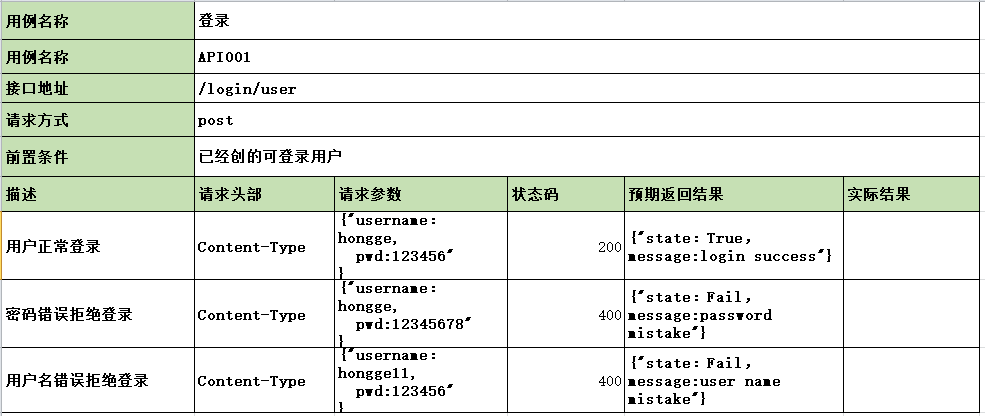
2.在后台构造合适的数据来满足接口的测试用例；

3.根据接口的返回值，断言其是否返回期望结果，并查看数据库验证；

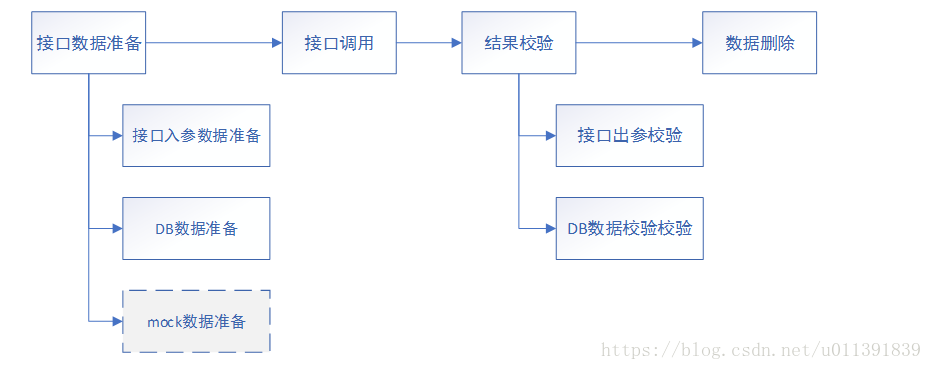
4.测试用例涉及多个步骤的，应对涉及的步骤都验证

5.删除测试过程中产生的结果，确保每个用例执行前都是一个清洁的环境

7. 接口测试用例模板



8. 接口数据准备



9. 接口测试工具

1) 接口测试工具选择

Jmeter：可以测试各种类型的接口，不支持的也可以通过网上或自己编写的插件进行扩展。

10. 接口测试原则

（1） 基础配置，如域名、环境配置等，单出文件配置，方便不同环境测试、脚本维护

（2） 明确接口实现什么样的功能，实际需要什么样的功能，是否一致

（3） 接口测试数据太多，用该数据驱动模式更有层次，且易于维护

（4） 要在众多测试用例中选出冒烟测试用例及可用于性能测试的用例

（5） 先单接口测试，在多接口业务测试

（6） 测试完成以后，需要清洗脏数据

11. 接口测试执行实例

11.1 使用Jmeter测试接口实例

n 接口测试环境准备

ü Jdk1.8或以上：http://www.oracle.com/technetwork/java/javase/downloads/index.html

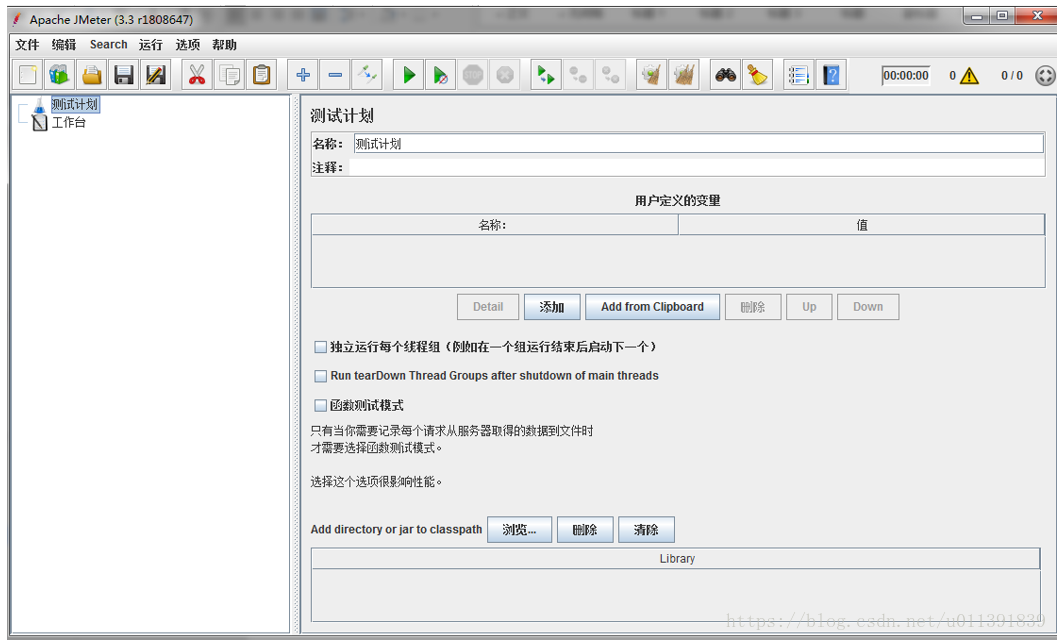
ü Jmeter下载址址：http://jmeter.apache.org/download\_jmeter.cgi

ü 插件的下载安装地址：http://www.jmeter-plugins.org/

n 接口测试过程

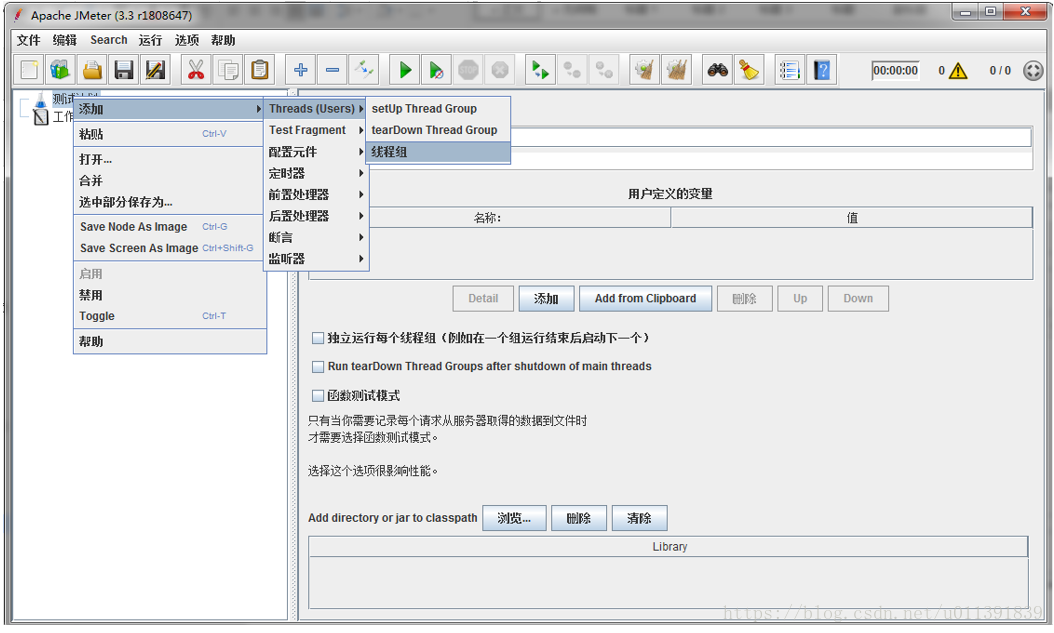
1) 新增一个测试计划

下载好Jmeter后，双击bin目录下的jmeter.bat文件



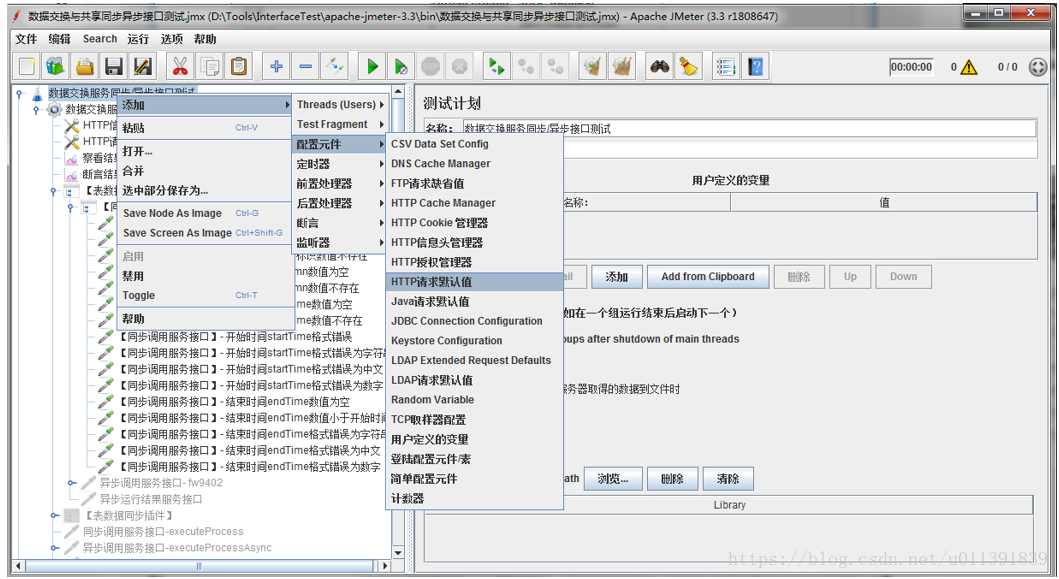
2) 　添加线程组

在“测试计划”上点击鼠标右键-->添加-->threads(Users)-->线程组，添加测试场景设置组件，接口测试中一般设置为1个“线程数”，根据测试数据的个数设定“循环次数”。



3) 添加“Http请求默认值”

在上步的线程组上右键添加-->配置元件-->HTTP请求默认值。

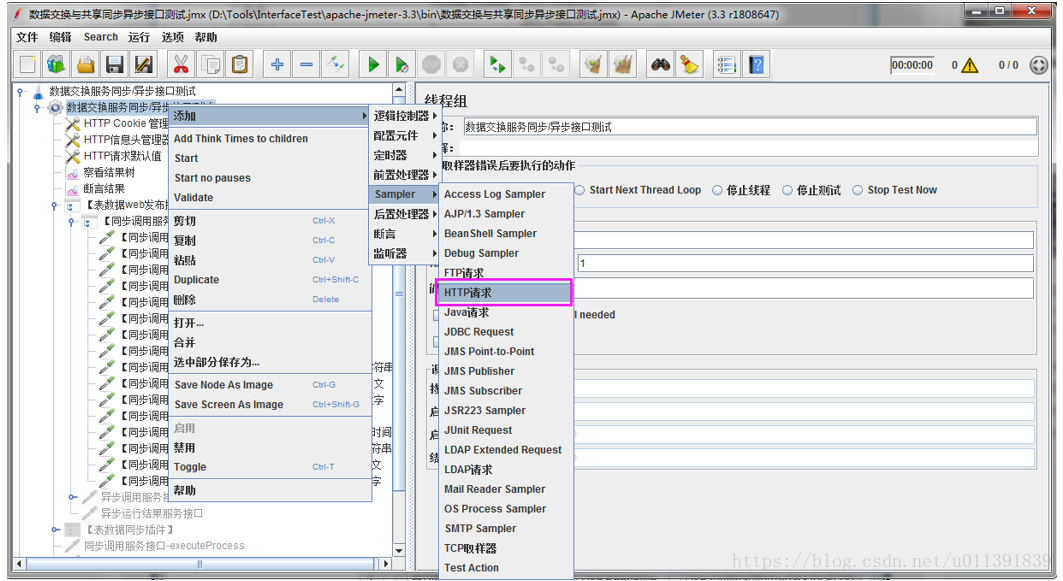


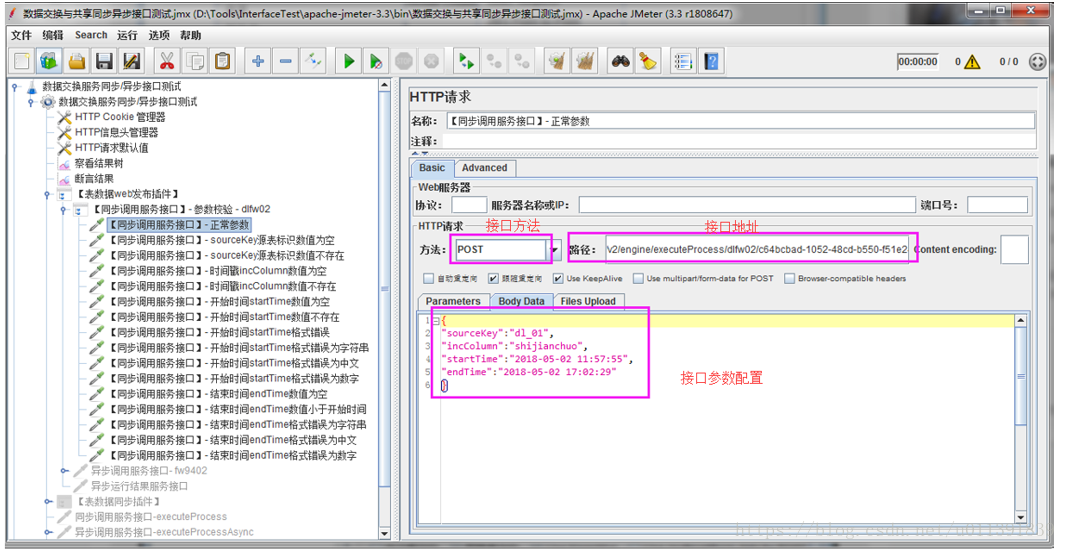
当所有的接口测试的访问域名和端口都一样时，可以使用该元件，一旦服务器地址变更，只需要修改请求默认值即可。

4) 在“线程组”里添加“HTTP 请求”的Sampler

在“线程组”右键-->添加-->samlper-->“HTTP 请求”

在HTTP请求设置页面，录入被测接口的详细信息，包括请求路径，对应的请求方法，以及随请求一起发送的参数列表，配置如下：

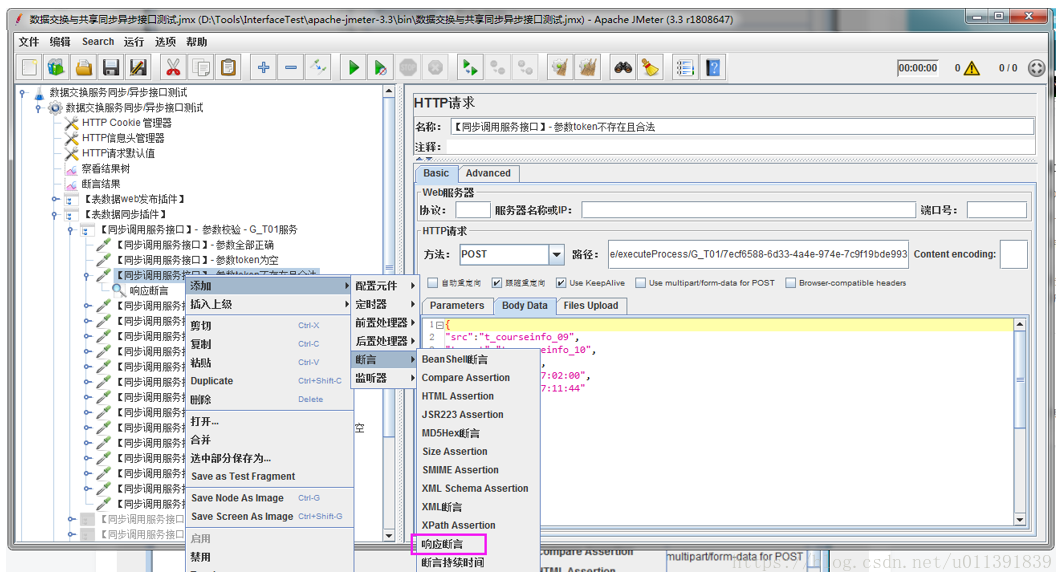




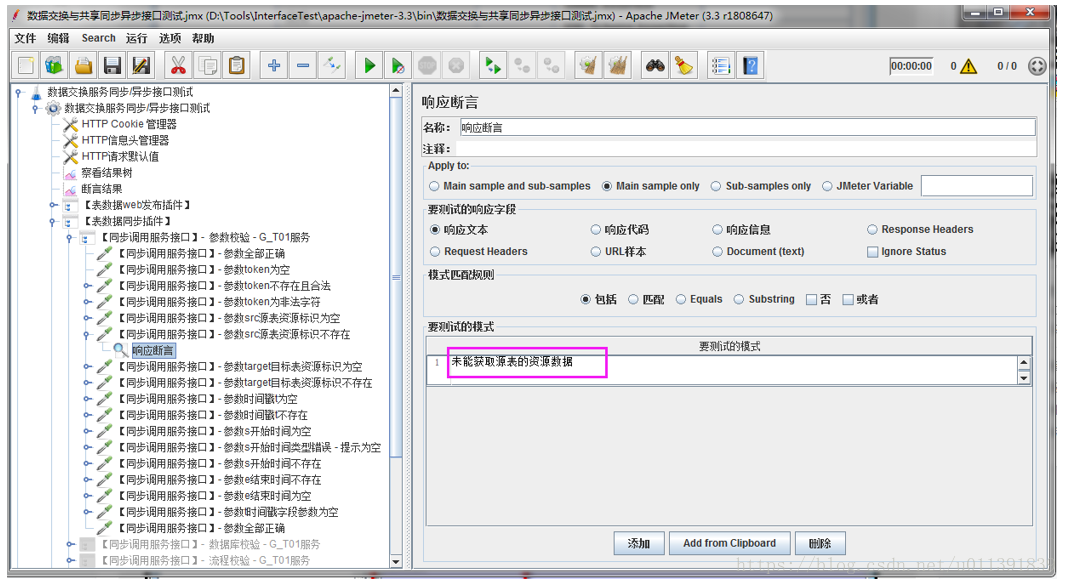
注：由于Jmeter请求线程组内的请求时从第一个开始执行，所以我们将需要最先执行的请求放在前面

5) 添加断言

在被测接口对应的“HTTP 请求”上，添加“响应断言”。右键点击HTTP请求“添加”–>“断言”–>“响应断言”



在设置页面上添加对相应结果的正则表达式存在性判断即可：



　 要测试的响应字段：响应文本、Document(text)、URL样本、响应信息、Response Headers、Lgnore Staus等选项。虽然接口返回的是Json格式的数据，但对于Jmeter来说返回数据为文本，所以，这里可以勾选“响应文本”

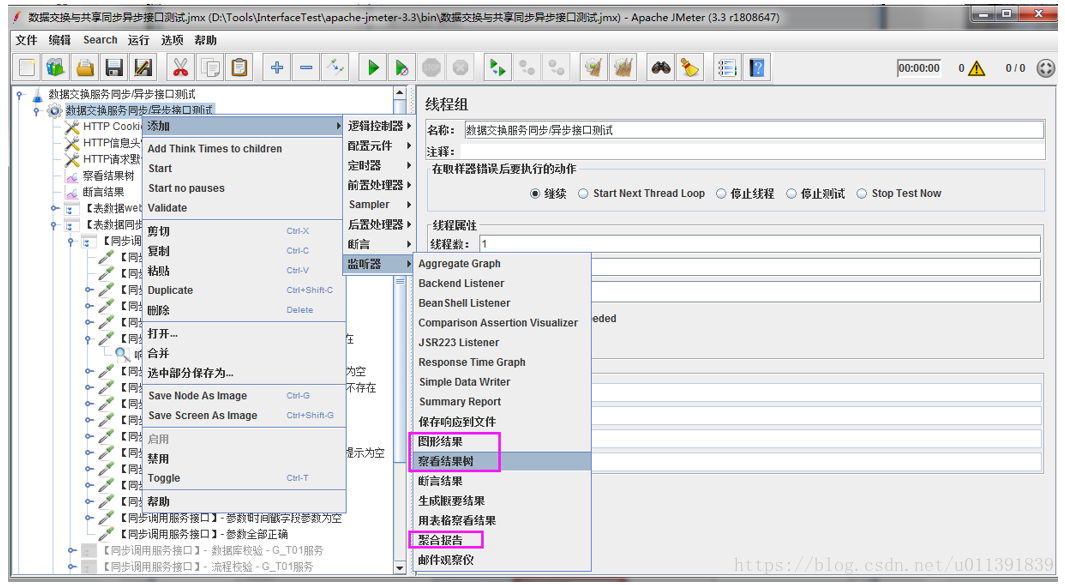
　　模式匹配规则：包括、匹配、Equals、Substring。这里只需要验证返回数据中是否包含主要的关键字，所以，这里勾选“包括”。

要测试的模式：其实就是断言的数据。点击“添加”按钮，输入要断言的数据。

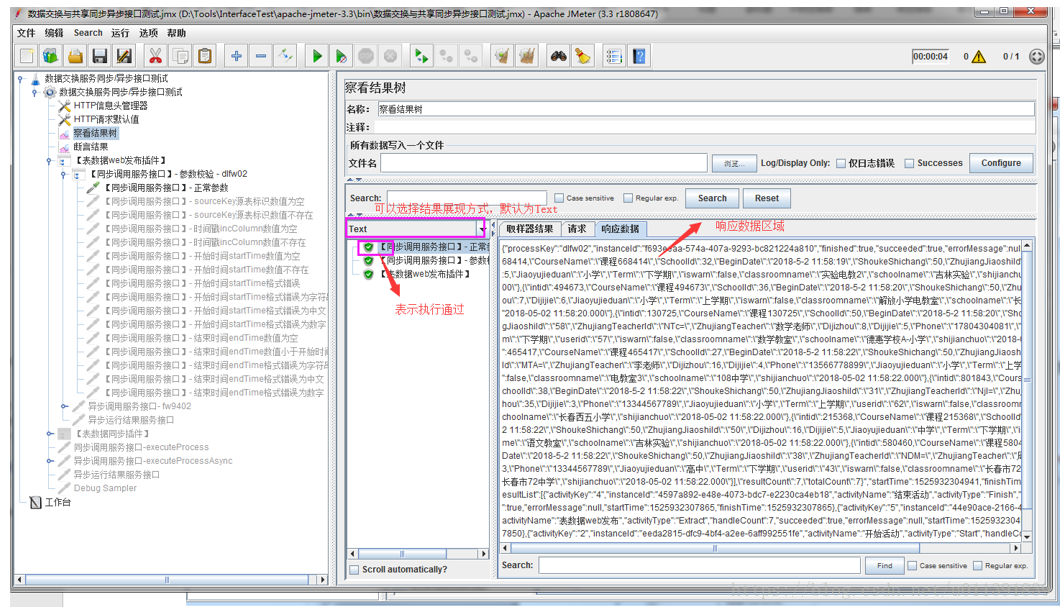
6) 添加监听器（测试报告配置）

在“线程组”右键-->添加-->监听器->查看结果树、用表格查看结果、聚合报告三种结果的报告展示

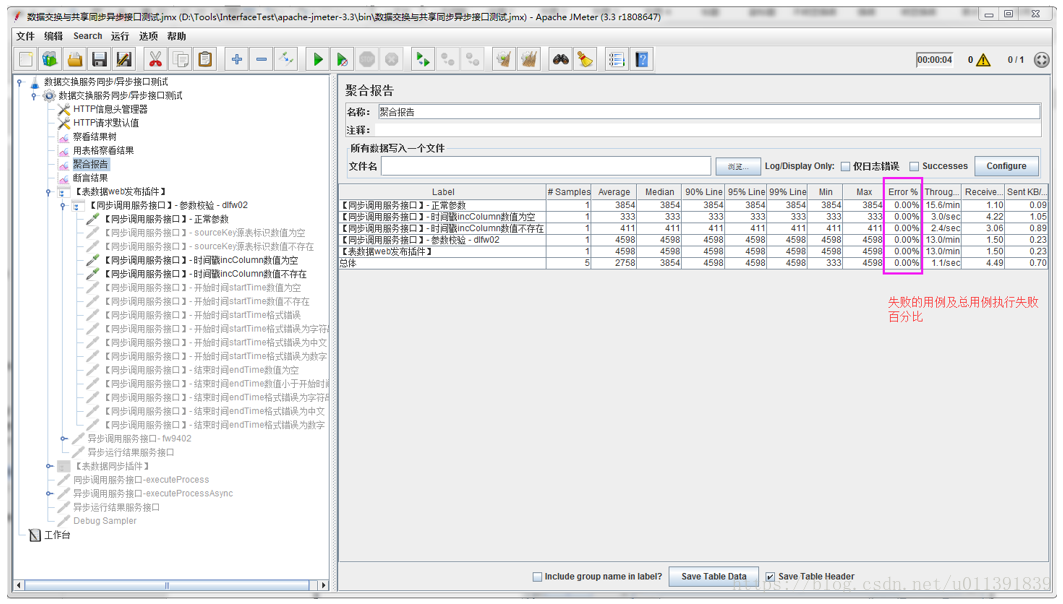
点击运行后，即可看到运行结果，结果如下：



运行结果查看：



使用聚合报告查看：



上述步骤完成了一个简单测试案例的创建，复杂测试案例均在此基础上扩展完成。使用Jmeter工具开发的接口测试案例，一个子系统建议放在同一个 “测试计划”中，流程测试可以通过“线程组”来区分，这样也便于设定不同的测试数据个数。比较独立的接口，可以统一放在一个线程组内，顺序完成测试。

　流程性接口的测试：如果要测试的接口可以组成一个流程，只需要顺序添加多个“HTTP 请求”的Sampler，各请求之间可以提取需要在上下文传递的数据作为参数，以保证流程中数据的一致性。

12. 接口测试结果报告（待定）

13. 接口测试中常见问题

接口测试经常遇到如下的bug和问题：

（1） 传入参数处理不当，导致程序crash;

（2） 类型溢出，导致数据读出和写入不一致；

（3） 因对象权限为进行校验，可以访问其他用户敏感信息；

（4） 状态处理不当，导致逻辑出现错乱；

（5） 逻辑校验不完善，可利用漏洞获取非正当利益等；

14. 接口自动化适用场景及持续集成

接口自动化框架：Jmeter+Jenkins

目前设计的自动化接口测试案例有两个运行场景：

1) 测试前置、开发自测：一个新的自动化接口测试案例开发完成后，直接发给接口对应的开发，安排在开发本地环境执行，一旦开发确认完成接口开发，就开始执行接口测试案例，基本上可以实时拿到测试结果，方便开发快速做出判断。【开发本地运行的方式就是打开JMeter工具，导入JMX文件，开始执行可。】

2) 回归测试：开发本地测试通过后，或整个需求手工测试通过后，把自动化的接口测试案例做分类整理，挑选出需要纳入到回归测试中的案例，在持续集成环境重新准备测试数据，并把案例纳入到持续集成的job中来，这些用于回归的接口测试案例需要配置到持续集成平台自动运行。

对接口测试而言，持续集成自动化是核心内容，通过自动化的手段才能有效降低成本，提高接口测试的价值。如果使用LR、JMeter、SoapUI工具做自动化测试，工具本身支持命令行模式运行，可以接合Jenkins 等自动化平台，实现项目版本更新后的自动化回归测试

关于持续自动化回归测试的建议：

1、接口脚本开发时要注意参数的取值的可用性，不因为时间或数据状态的变化引起脚本不能正常运行，降低脚本维护成本.

2、接口回归功能的覆盖度控制，需要根据脚本的实际功能和重要性判断自动化回归覆盖度，回归内容越多脚本维护成本越高，一般应用接口不建议全功能覆盖（毕竟接口有变化会做详细测试，如果没修改其它变更可能对其产生的影响一般不会影响其逻辑判断）

3、接口脚本需要一定的自动化校验能力，除请求http状态的判断外，还需要对核心内容的正常性做判断（判断内容可与数据库内容匹配等方式，不建议用写死的内容）。

4、持续性能测试，还需要做好相关的监控、性能指标的分析自动化，减少人工操作。