

测试RestfulDemo

2021年10月4日 16:34

在这个例子中，我们使用JMeter的一些元件来定义一个比较复杂的测试计划

性能测试的类型

基准测试

当我们制定好测试计划之后，我们会给系统施加一个较低的压力，主要用来查看我们定义的测试计划是否正确、有效

负载测试

当基准测试完成之后，我们会进行负载测试，即从低向高逐步增加压力，通过这样的一个过程来逐步找到系统的瓶颈在哪里，然后就可以修改参数、修改代码等来提高系统的负载能力

压力测试

在调优完成之后，我们会对系统进行压力测试，来测试系统的容量，看系统的负载到底有多大

并发测试

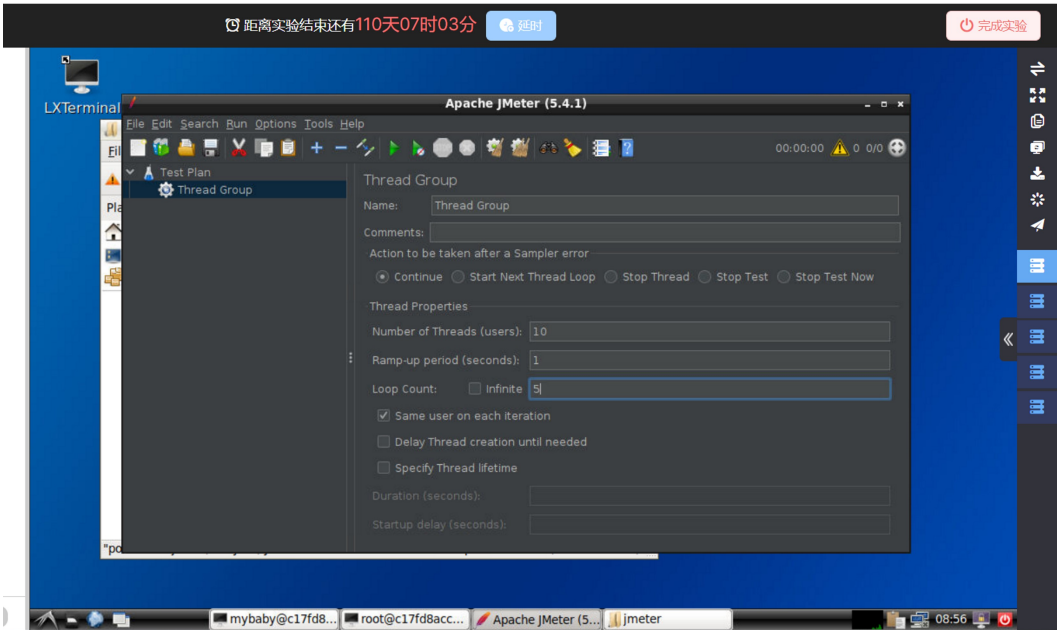
针对某些特定功能，我们可以进行并发的测试

我们用RestfulDemo来进行一个基准测试，我们选择其中的创建商品的API来进行测试，这是一个post请求。我们知道实际过程中我们是不能重复添加商品的，所以我们借助元件来演示如何约束关键字，在测试时向系统增加不同的数据

具体操作

我们新建测试计划，添加Thread Group，定义10个线程，一秒钟之内跑完，循环5次

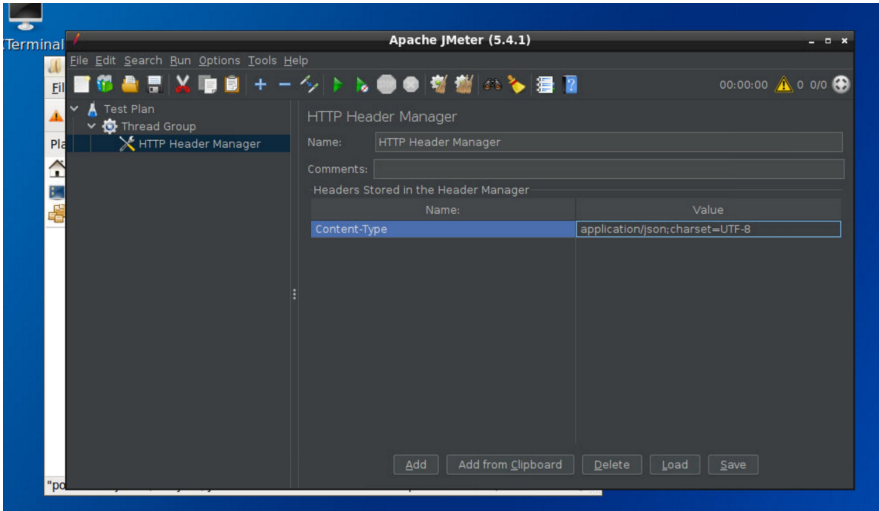
这样大概有5秒钟的测试，50次的请求



添加一个HTTP Header Manager元件，来定义content-type

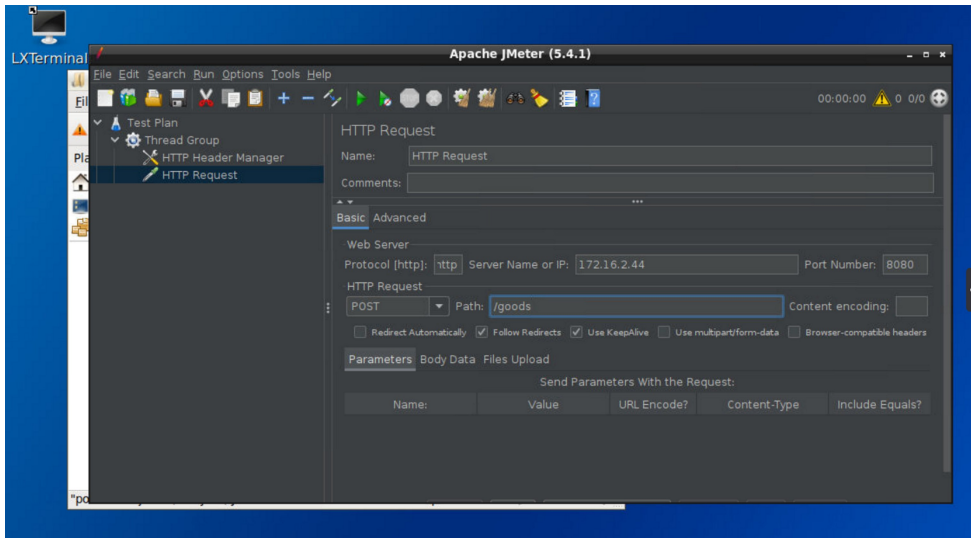
因为这是一个post请求，所以我们需要告诉服务器我们post请求的body格式是json，字符集是utf-8

如果不加这个头的话，会返回400的错误



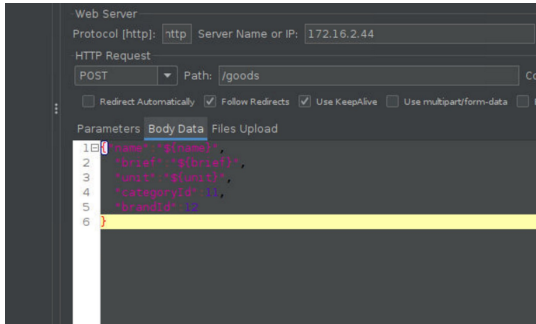
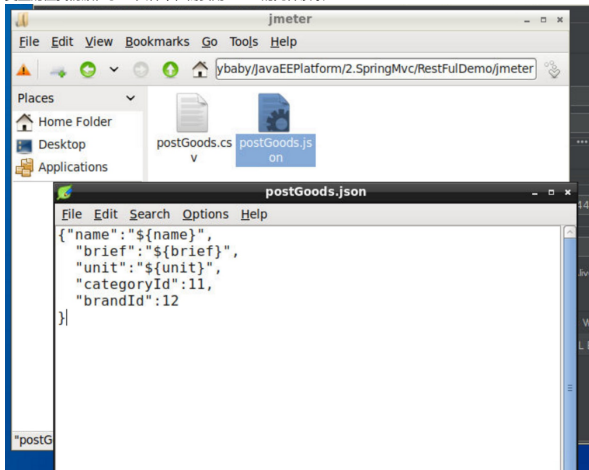
然后添加HTTPRequest的Sampler，和之前的TestTomcat中的差不多

指定协议、服务器ip、端口、请求类型、url



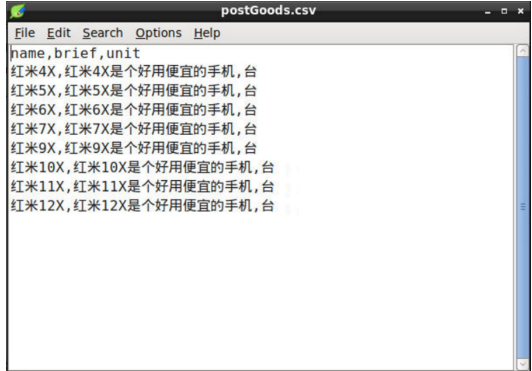
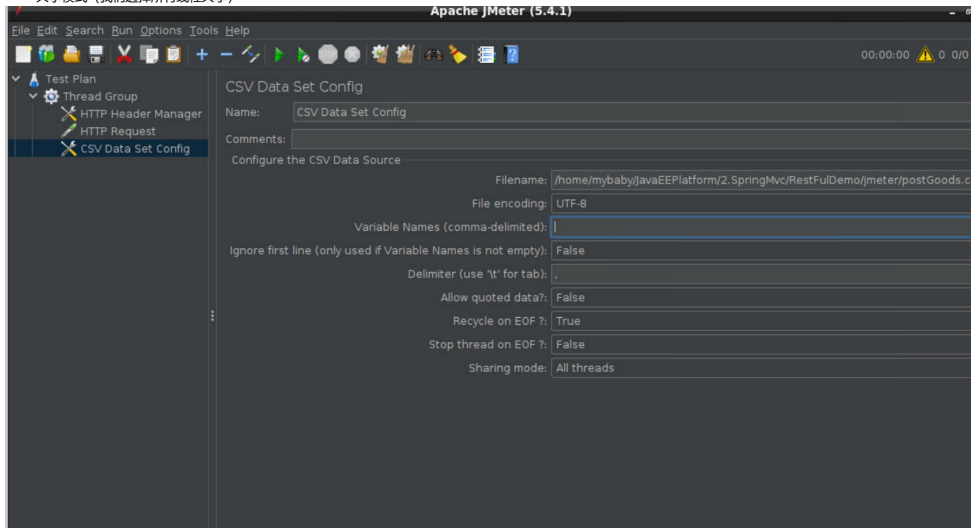
因为post请求，所以我们需要定义在body中传过去的数据，这里我们用JMeter的元件来从文件中读取测试数据
在项目的根目录的jmeter文件夹内有测试数据，.json文件是一个模板，复制粘贴到Body Data中
模板解析：

有三个json字段是\${开头的值，表示这个值是变量，需要从其他地方获取
在这里表示每个请求中的这些变量需要从模板中取值
变量的值我们放在了csv文件中，需要用JMeter的元件读取



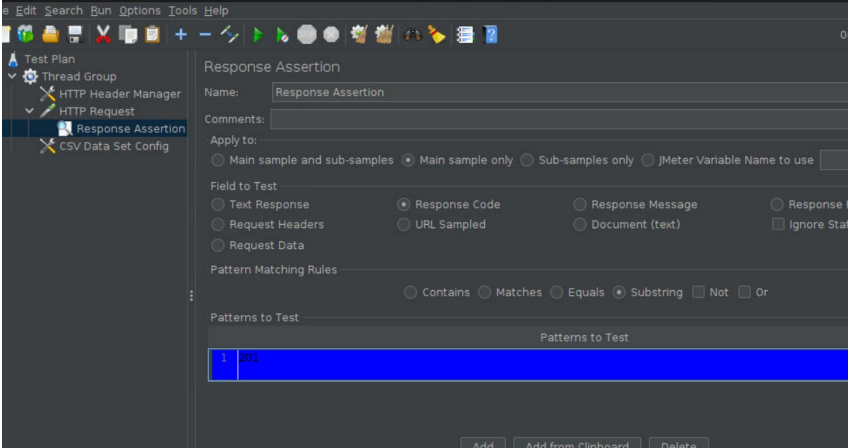
为ThreadGroup添加一个CSV Data Set Config，用来从CSV中读取数据
CSV文件的格式：第一行是变量的名称，后面每一行是三个变量的值
只需要为CSV Data Set Config指定文件名、格式即可，其余的使用默认设置
其余几个设置分别是：

- 变量名（csv中已经有了，就不需要指定）
- 是否忽略第一行（我们的csv中第一行是变量名，不用忽略）
- 使用什么符号作为分隔符（默认为半角逗号）
- 是否允许引号
- 是否在结束后循环
- 是否在结束后结束线程
- 共享模式（我们选择所有线程共享）



为Sampler定义Assertion

首先添加一个Response Assertion，规定返回码必须是201（post成功的返回码是201而非200）



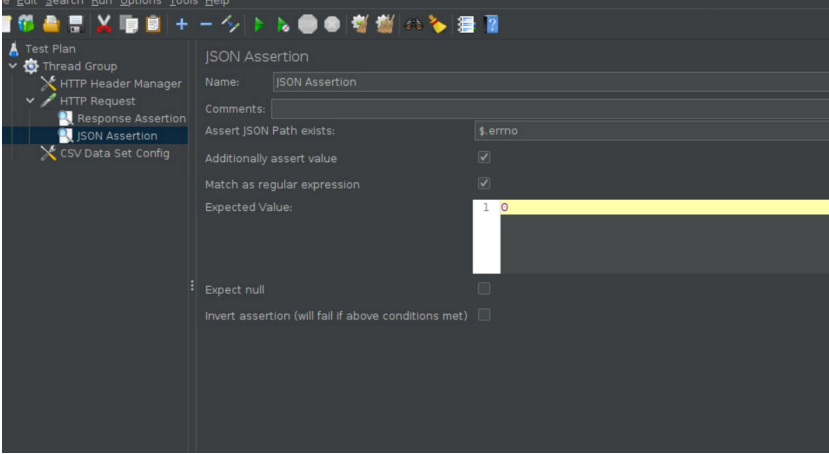
因为返回的是json格式的数据，所以我们不仅要查看返回码，还要看返回的body中的json数据

所以我们添加一个JSON Assertion

这里我们使用json path的定义，可以帮助我们坚持json是否符合我们定义的格式

勾选additionally assert value，表示我们可以指定断言值

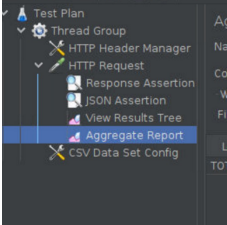
这里我们定义，返回的body的json中应该有error字段，并且其值应该为0



为Sampler添加Listener

第一个是View Result Tree，察看结果树，比较常用，用来收集结果

第二个是Aggregate Report，用来查看汇总结果

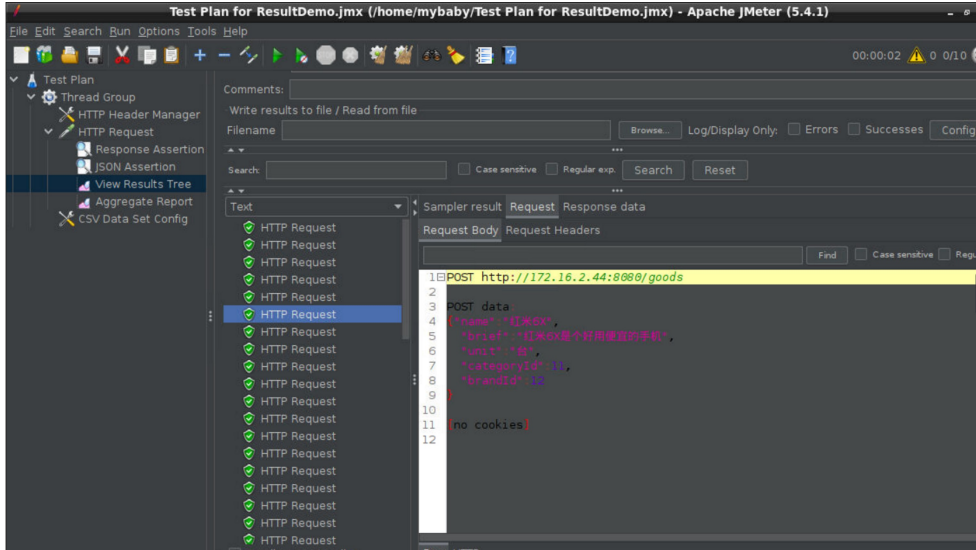


定义完成之后，我们将其保存为jmx文件。因为这只只是一个基准测试，所以我们用jmeter直接运行测试计划

测试之前，我们在目标服务器上用上一步的runjava.sh将打包好的jar包运行起来

./runjava.sh target/restfuldemo-0.0.1-SNAPSHOT.jar

之后再JMeter中点击Start，就可以在Listener中看到汇总结果



对于get请求，如果想拿到不同的值，可以使用随机值函数
/orders/\${__Random(1,100,id)}