

+微信17031115530拉测试微信群交流

持续分享

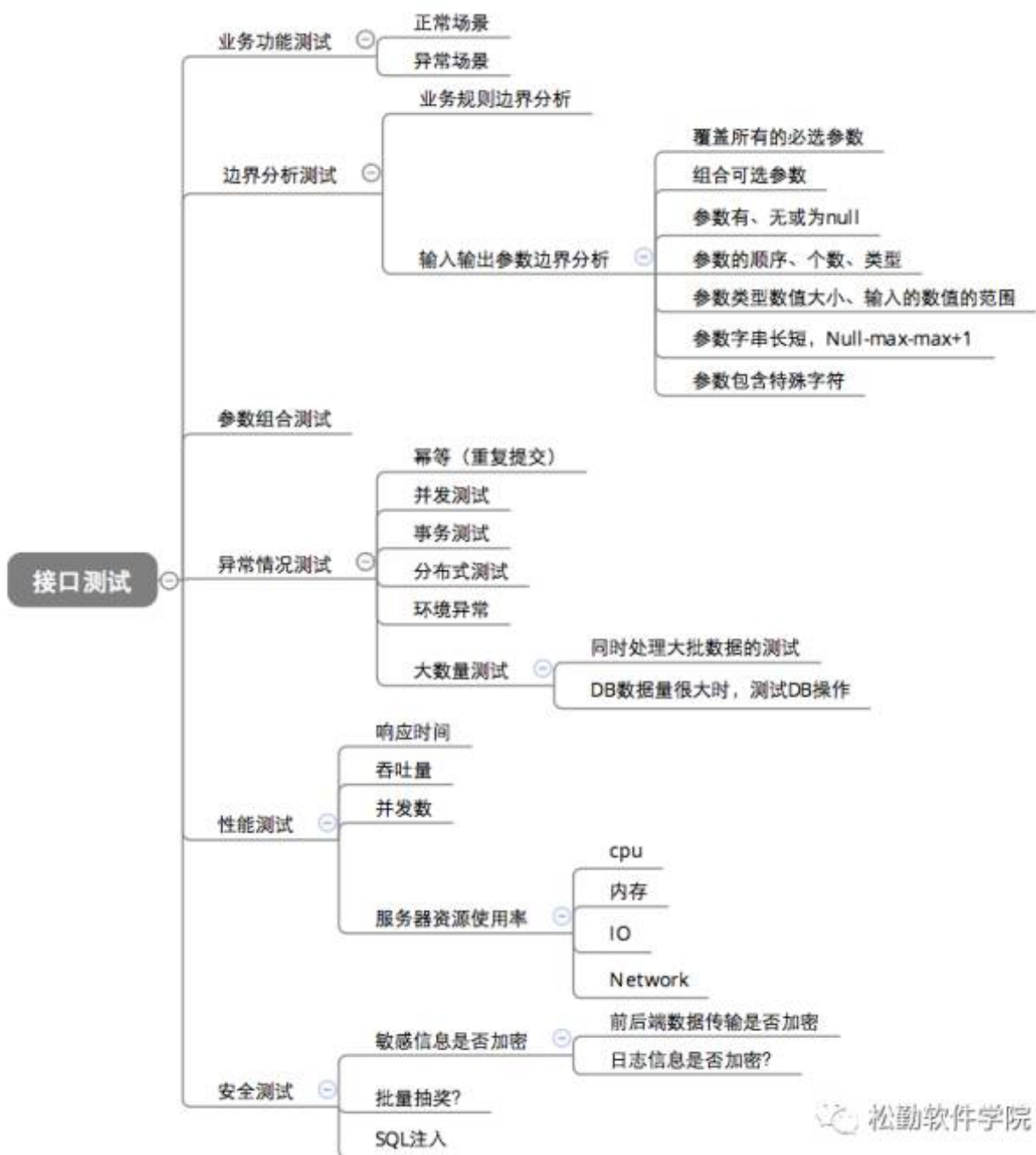
1. 什么是接口测试？为什么要做接口测试？

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。

由于如今的系统复杂度不断上升，传统的测试方法成本急剧增加且测试效率大幅下降，所以就要做接口测试。同时，接口测试相对容易实现自动化持续集成，且相对UI自动化也比较稳定，可以减少人工回归测试人力成本与时间，缩短测试周期，支持后端快速发版需求。接口持续集成是为什么能低成本高收益的根源。现在很多系统前后端架构是分离的，从安全层面来说，只依赖前端进行限制已经完全不能满足系统的安全要求（绕过前面实在太容易），需要后端同样进行控制，在这种情况下就需要从接口层面进行验证。前后端传输、日志打印等信息是否加密传输也是需要验证的，特别是涉及到用户的隐私信息，如身份证，银行卡等。

2. 后端接口测试测什么？

从下面这张图可以看出后端接口测试主要测了些什么：



3. 怎么做接口测试?

一般情况下，由于我们项目前后调用主要是基于http协议的接口，所以测试接口时主要是通过工具或代码模拟http请求的发送和接收。所以我们下面整理了一下使用Jmeter工具进行http接口测试。

3.1 开发接口测试案例的整体方案

第一步： 分析出测试需求，并拿到开发提供的接口说明文档；

第二步： 从接口说明文档中整理出接口测试案例，里面要包括详细的入参和出参数据以及明确的格式和检查点；

第三步： 和开发一起对接口测试案例进行评审；

第四步： 结合开发库，准备接口测试案例中的入参和出参数据，并整理成csv格式的文件；

第五步： 结合接口测试案例文档和csv格式的数据文档，做接口测试案例的自动化案例开发。

3.2 接口自动化适用场景

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。

3.3 接口测试环境准备

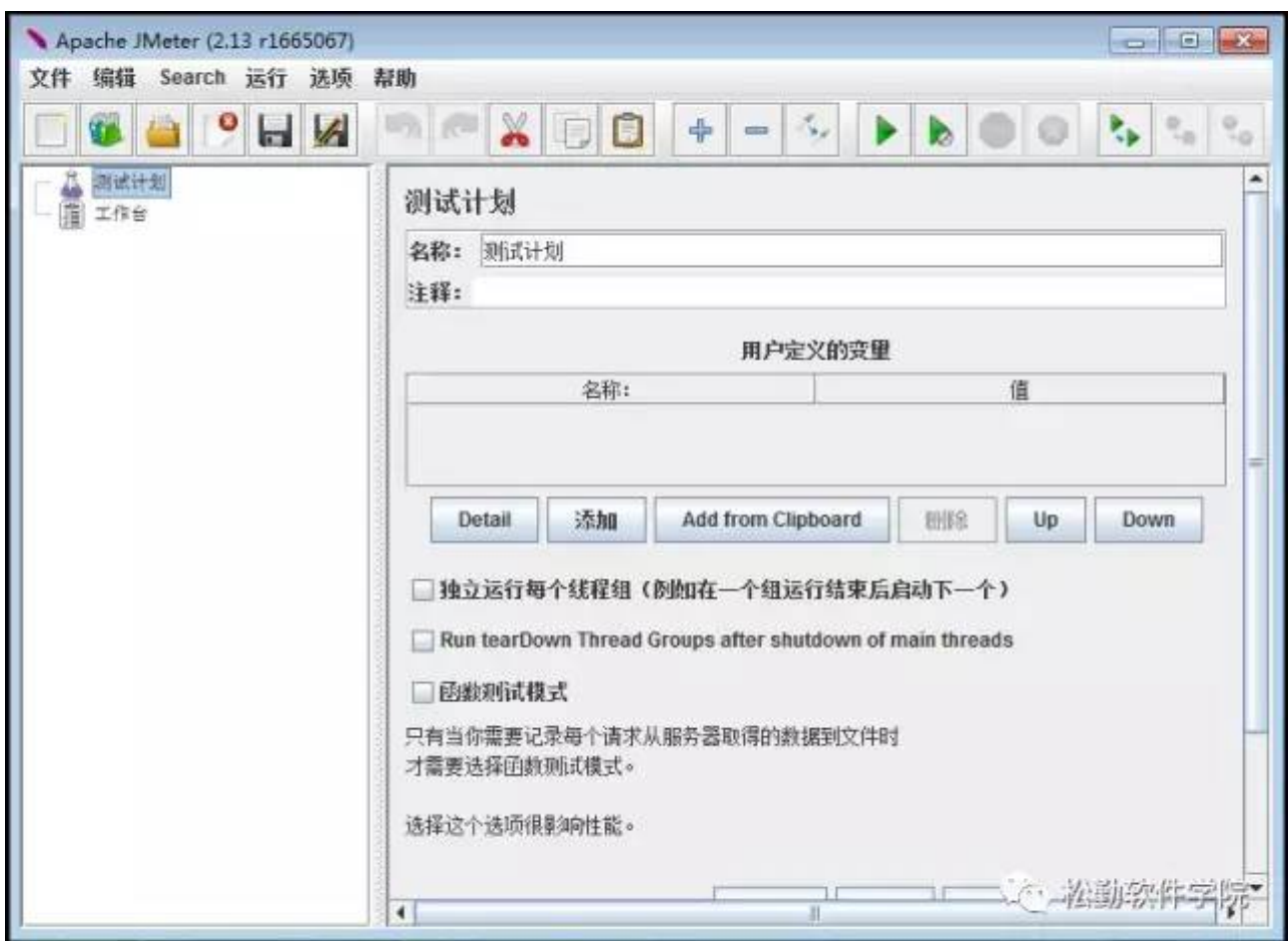
Jdk1.6或以上：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Jmeter, 下载地址: http://jmeter.apache.org/download_jmeter.cgi

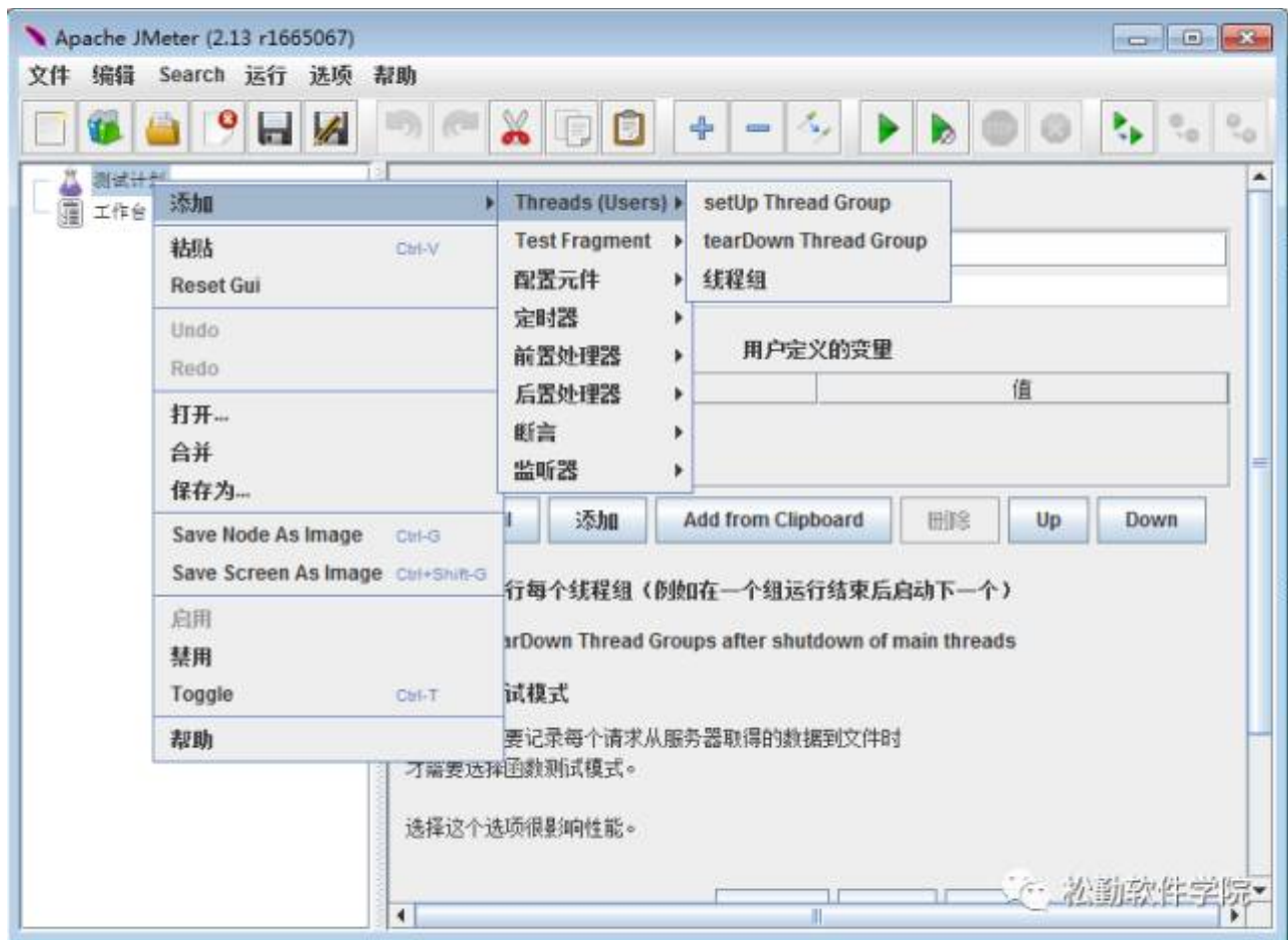
插件的下载安装地址: <http://www.jmeter-plugins.org/>

3.4 创建工程

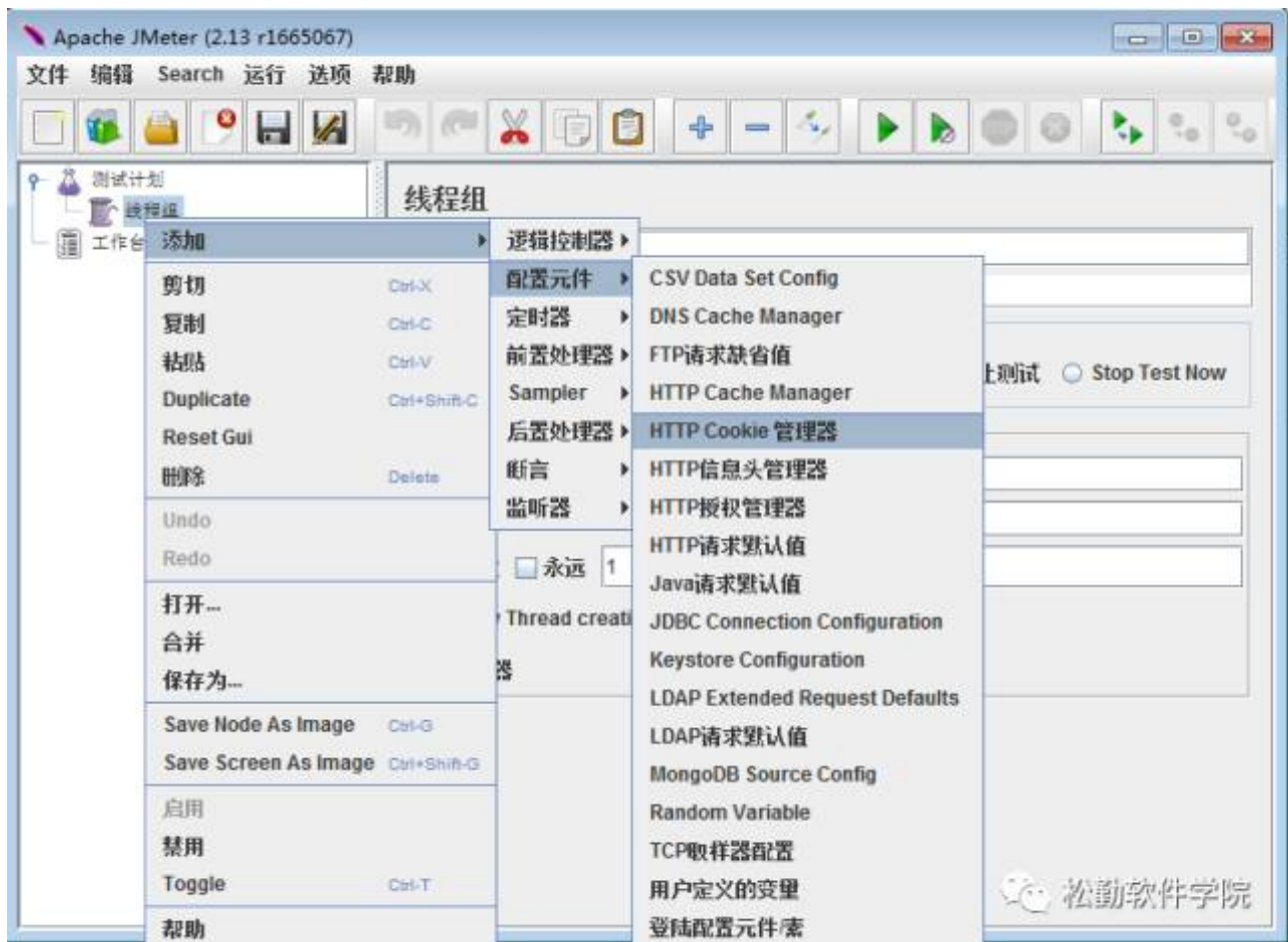
1.打开Jmeter: 下载好Jmeter后, 双击bin目录下的jmeter.bat文件:



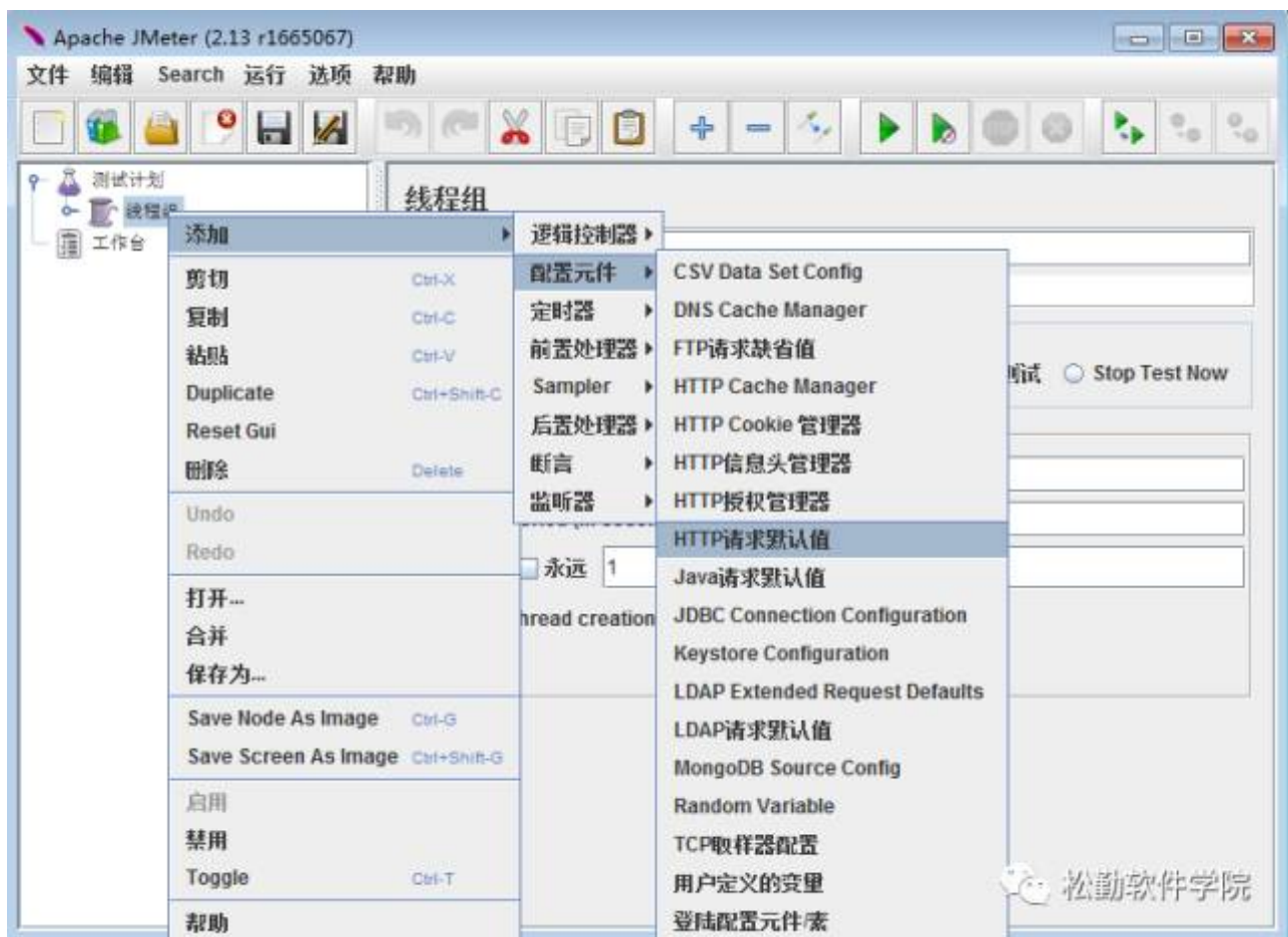
2.添加线程组: 在“测试计划”上点击鼠标右键-->添加-->threads(Users)-->线程组, 添加测试场景设置组件, 接口测试中一般设置为1个“线程数”, 根据测试数据的个数设定“循环次数”。



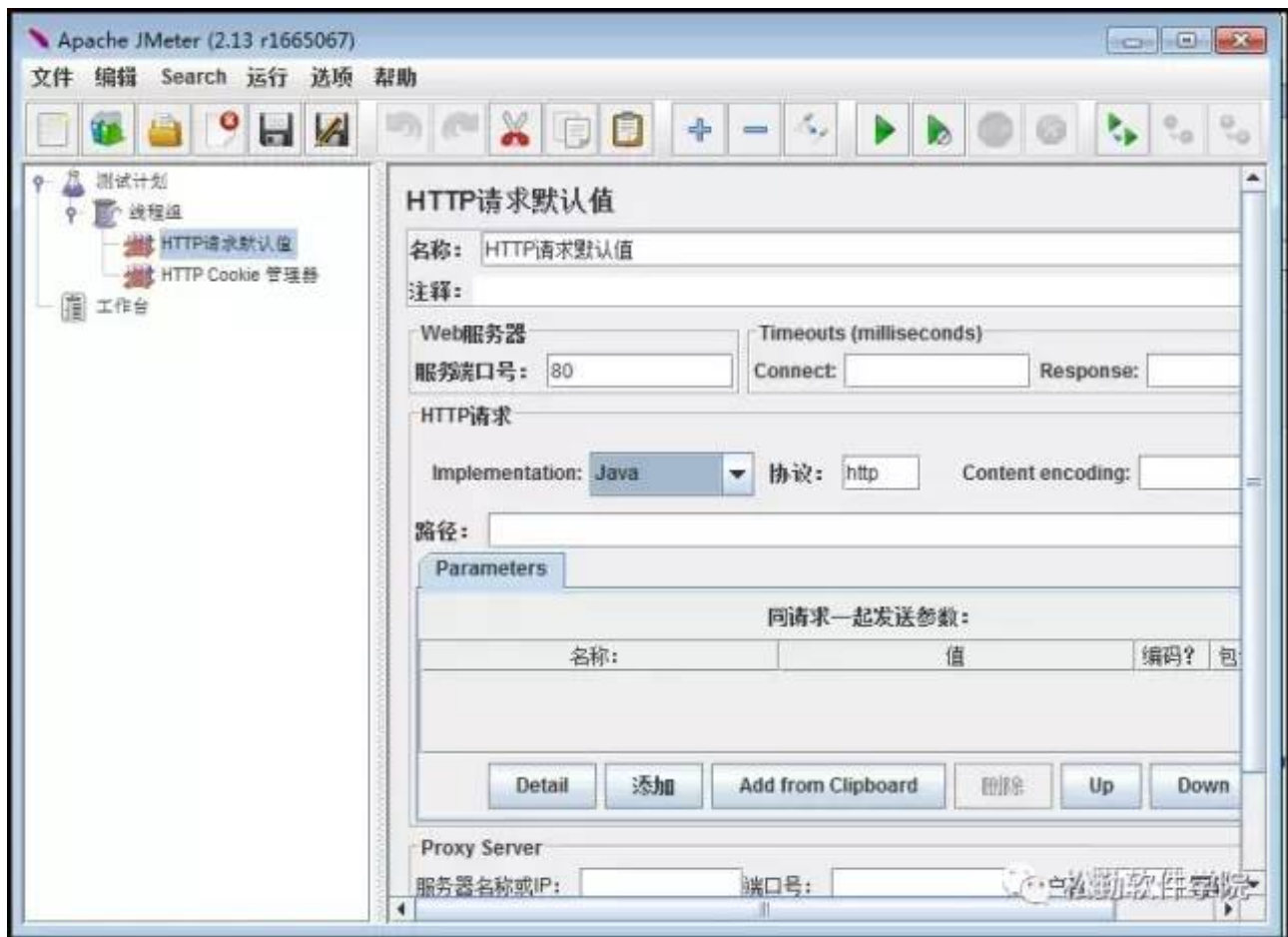
3.添加“HTTP Cookie管理器”：



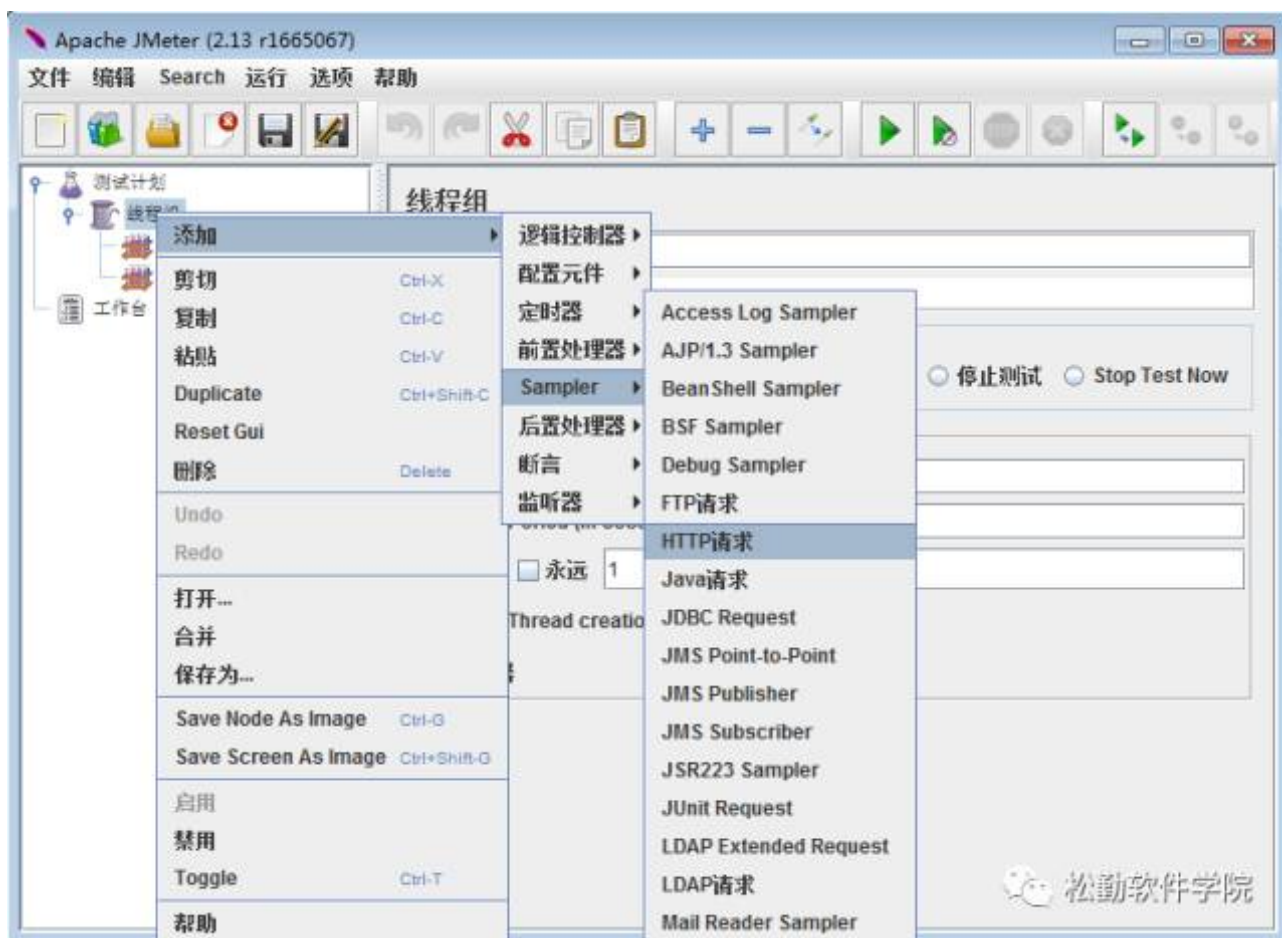
4.添加“Http请求默认值”组件，当被测系统有唯一的访问域名和端口时，这个组件很好用：



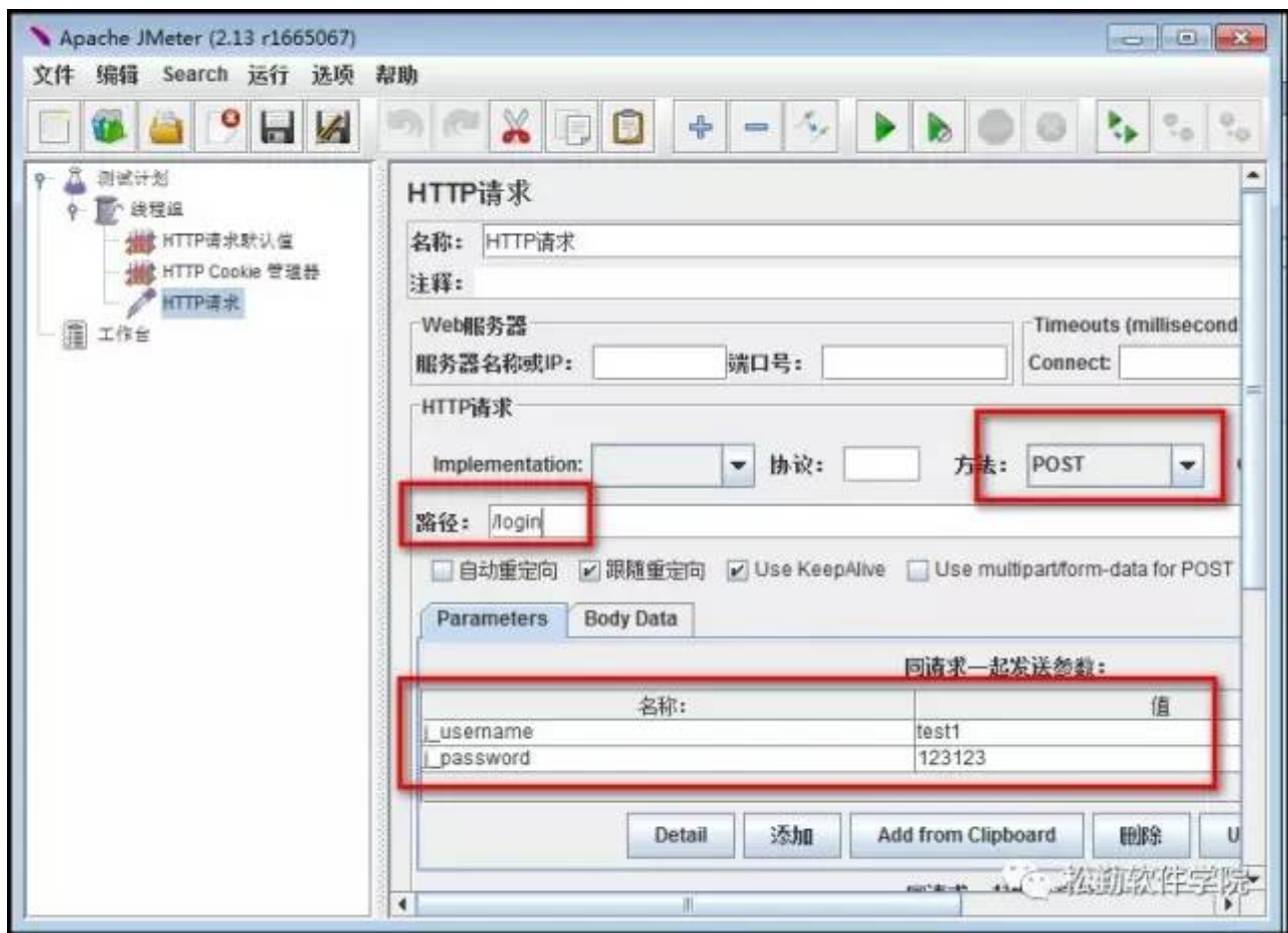
5.在“HTTP 请求默认值”组件配置页面，填写被测系统的域名和端口，http请求的实现包版本以及具体协议类型，线程组里的所有“HTTP Sampler”可默认使用此设置。



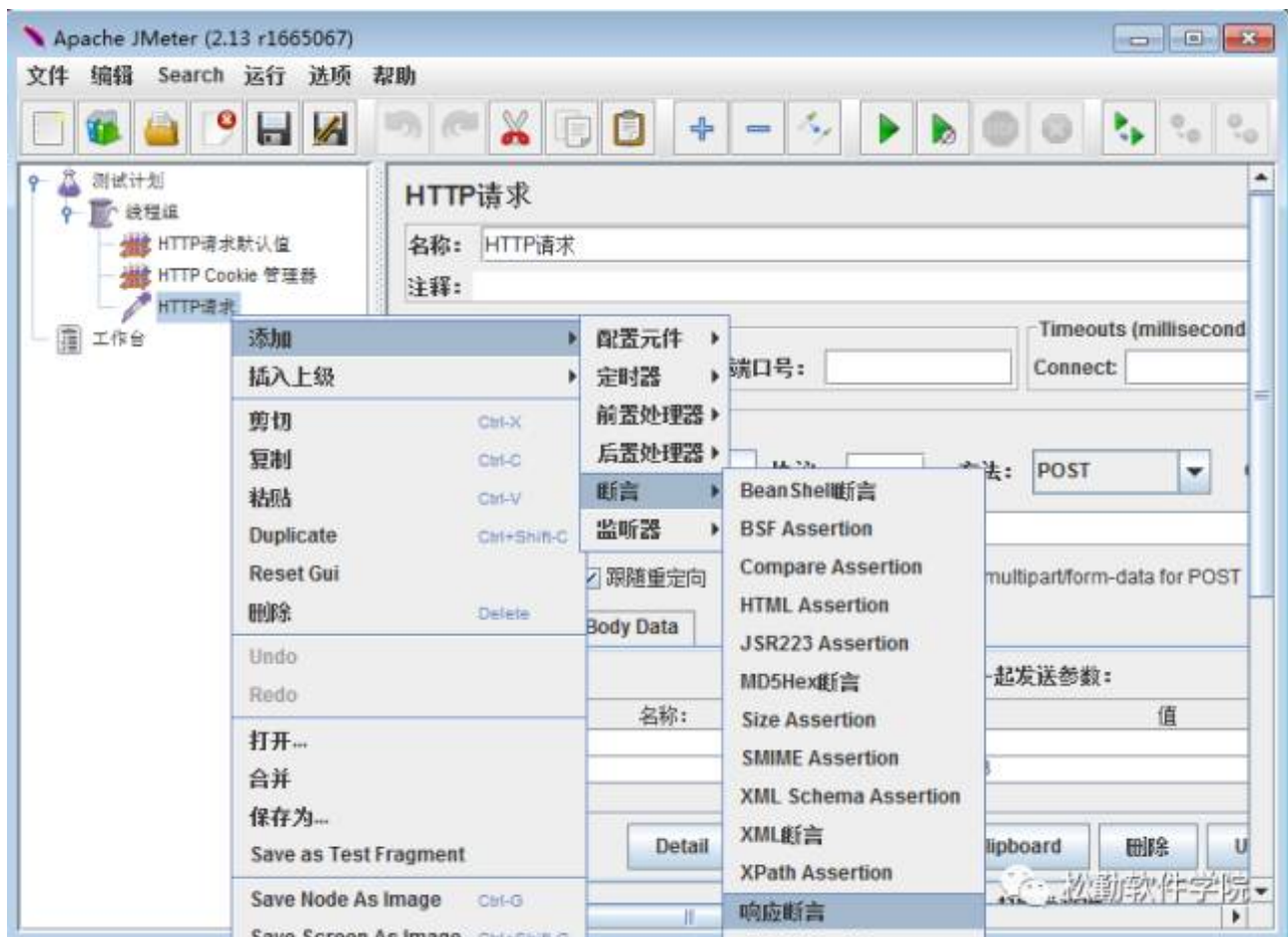
6.在“线程组”里添加“HTTP 请求”的Sampler



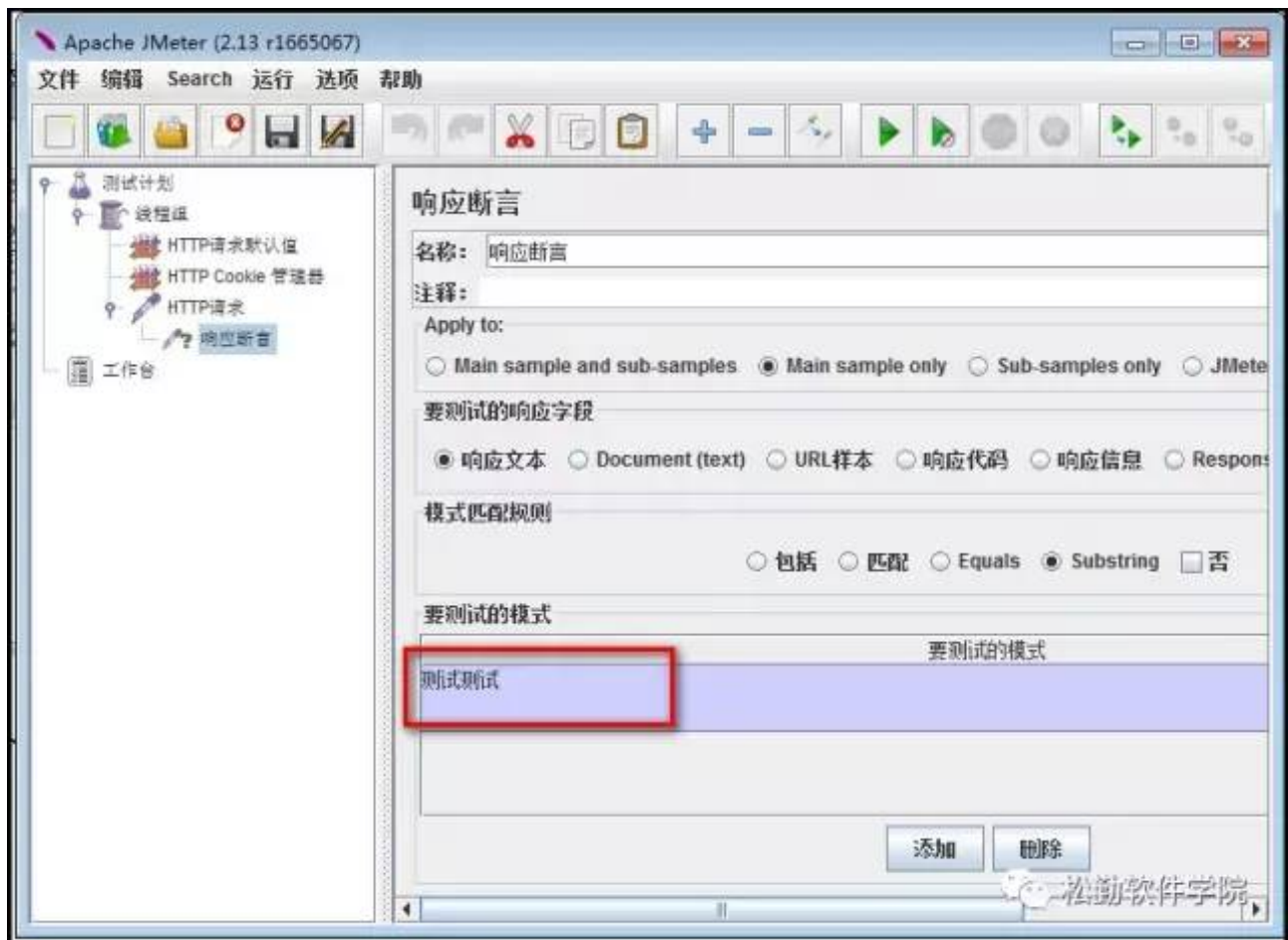
7.在HTTP请求设置页面，录入被测接口的详细信息，包括请求路径，对应的请求方法，以及随请求一起发送的参数列表：



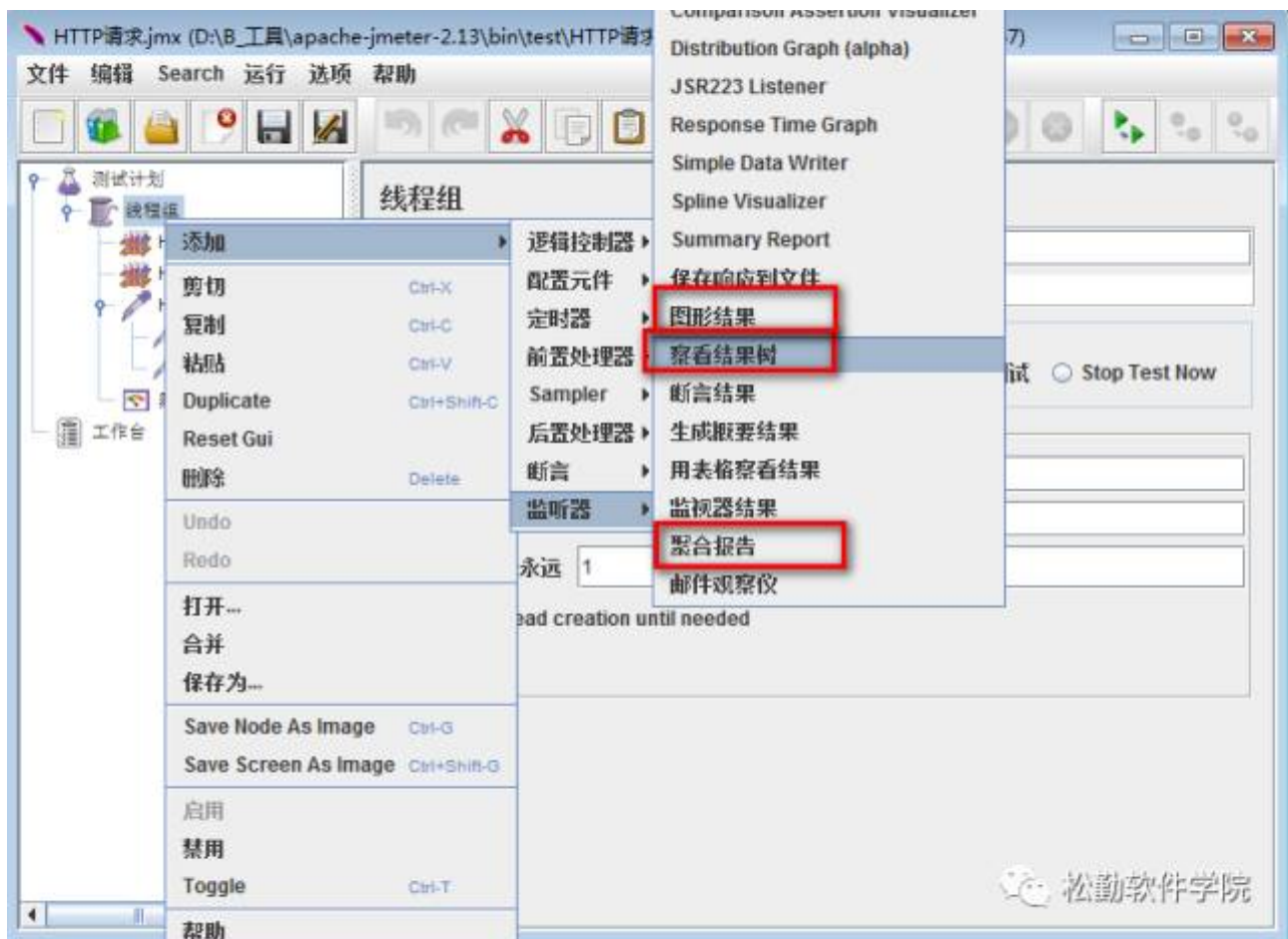
8.设置检查点：在被测接口对应的“HTTP 请求”上，添加“响应断言”：



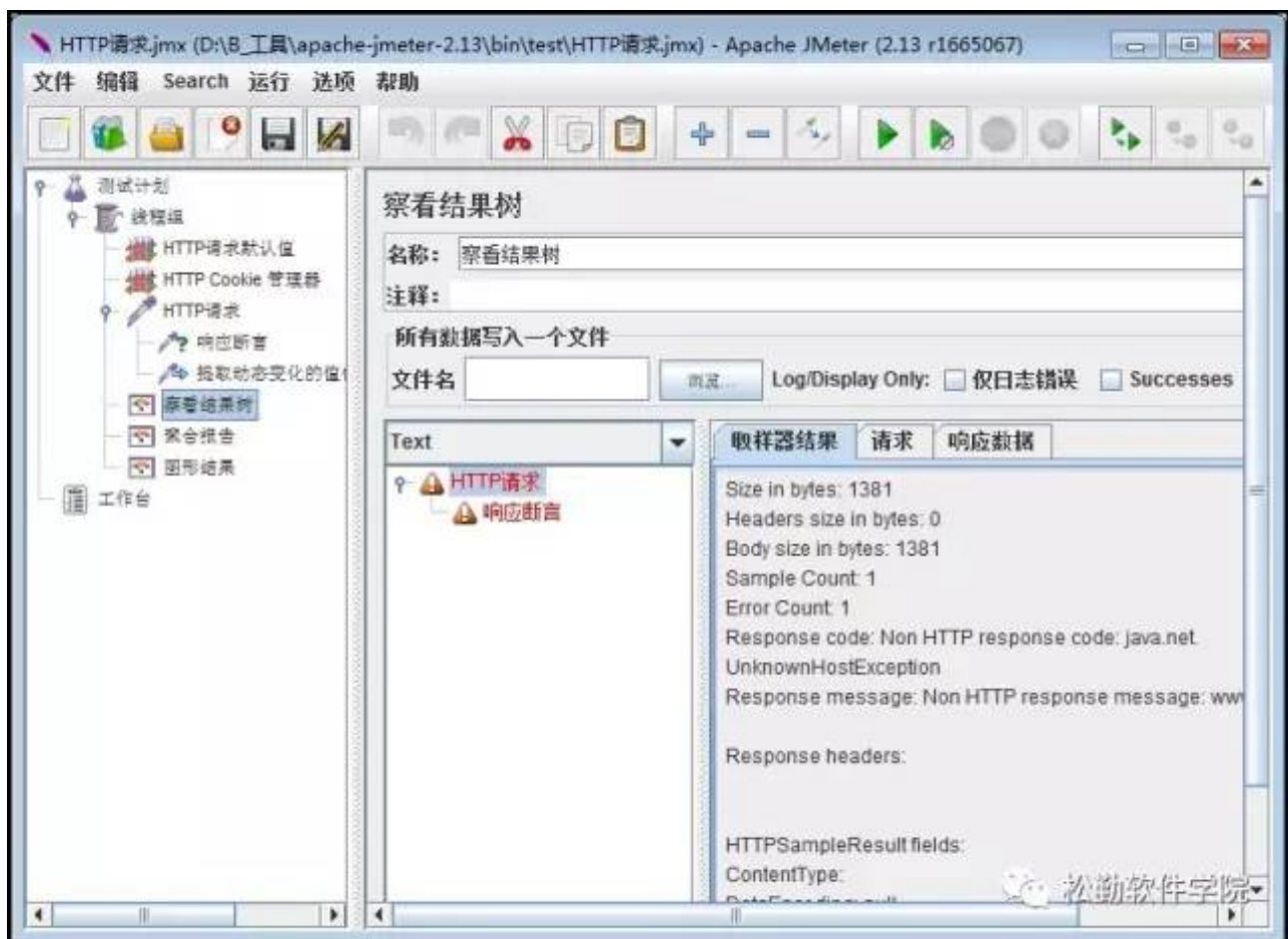
9.在设置页面上添加对相应结果的正则表达式存在性判断即可：



10. 添加监听器：方便查看运行后的结果



运行结果：



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。

4. 接口测试持续集成

对接口测试而言，持续集成自动化是核心内容，通过持续自动化的手段我们才能做到低成本高收益。目前我们已经实现了接口自动化，主要应用于回归阶段，后续还需要加强自动化的程度，包括但不限于下面的内容：

a) 流程方面：在回归阶段加强接口异常场景的覆盖度，并逐步向系统测试，冒烟测试阶段延伸，最终达到全流程自动化。

b) 结果展示：更加丰富的结果展示、趋势分析，质量统计和分析等

c) 问题定位：报错信息、日志更精准，方便问题复现与定位。

d) 结果校验：加强自动化校验能力，如数据库信息校验。

e) 代码覆盖率：不断尝试由目前的黑盒向白盒下探，提高代码覆盖率。

f) 性能需求：完善性能测试体系，通过自动化的手段监控接口性能指标是否正常。

5. 接口测试质量评估标准

a) 业务功能覆盖是否完整

b) 业务规则覆盖是否完整

c) 参数验证是否达到要求（边界、业务规则）

d) 接口异常场景覆盖是否完整

e) 接口覆盖率是否达到要求

f) 代码覆盖率是否达到要求

g) 性能指标是否满足要求

h) 安全指标是否满足要求