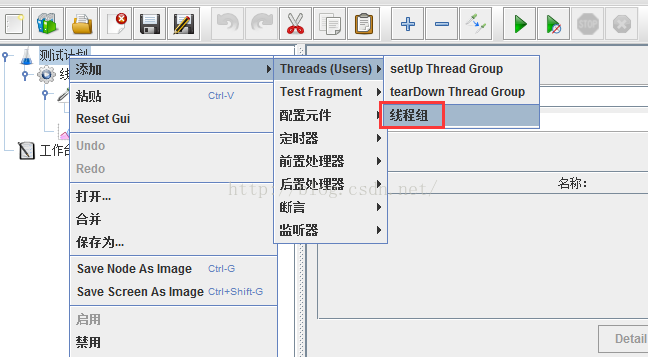
jmeter3进行压力测试

参考：<http://blog.csdn.net/lw4135/article/details/52153723>

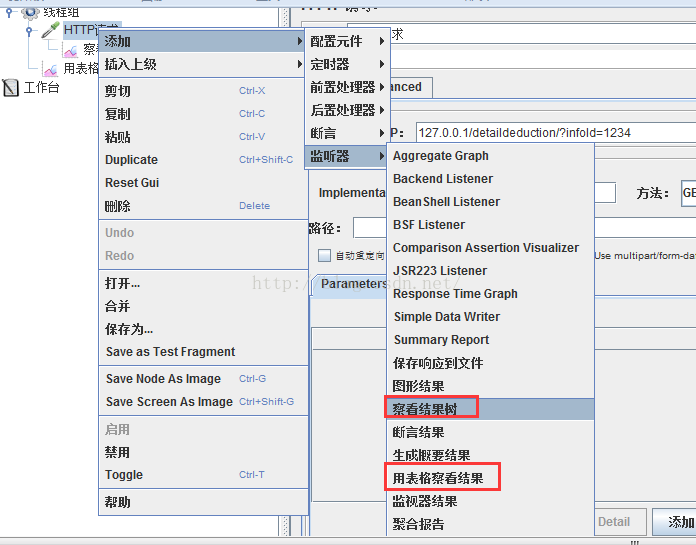
首先需要下载所需的插件-jmeter,解压

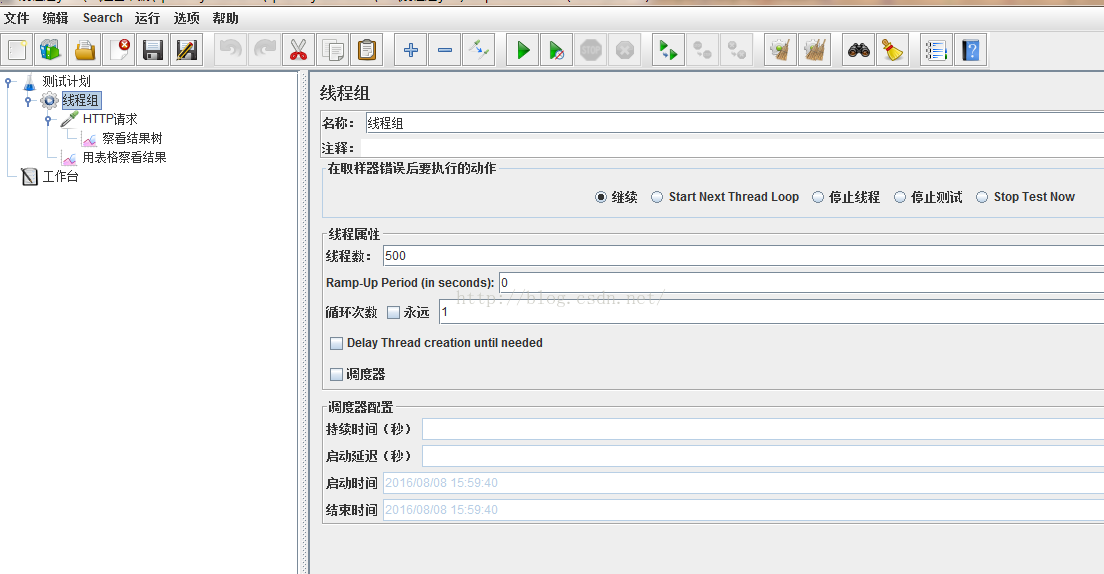
1.bin 目录下双击jmeter.bat启动。🡪一个GUI模式的页面

2.Ramp-Up Period (in seconds)表示线程之间间隔多少时间允许，单位是秒(就是在这个时间之内创建完所有的线程)



3.链接不用写协议头





4.**运行结果图**：

Sample：每个请求的序号

Start Time：每个请求开始时间

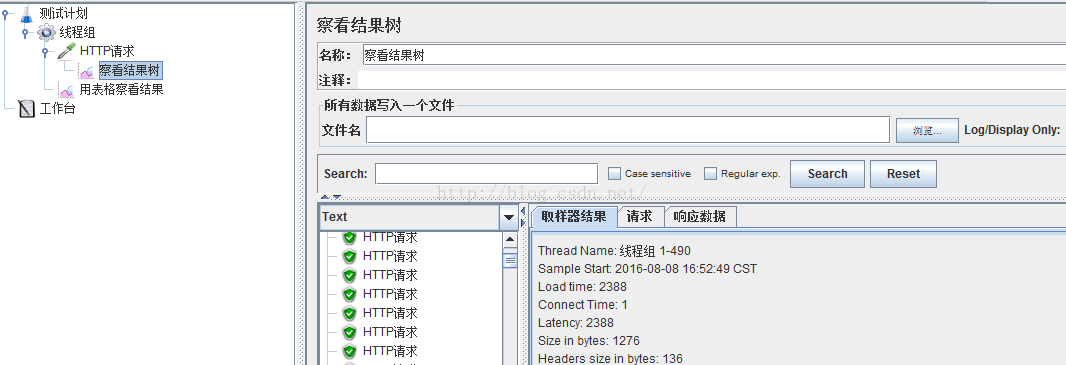
Thread Name：每个线程的名称

Label：Http请求名称

Sample Time：每个请求所花时间，单位毫秒

Status：请求状态，如果为勾则表示成功，如果为叉表示失败。

Bytes：请求的字节数



在下面还有几个参数：

样本数目：也就是上面所说的请求个数，成功的情况下等于你设定的**并发数目乘以循环次数**

平均：每个线程请求的平均时间

最新样本：表示服务器响应最后一个请求的时间

偏离：服务器响应时间变化、离散程度测量值的大小。

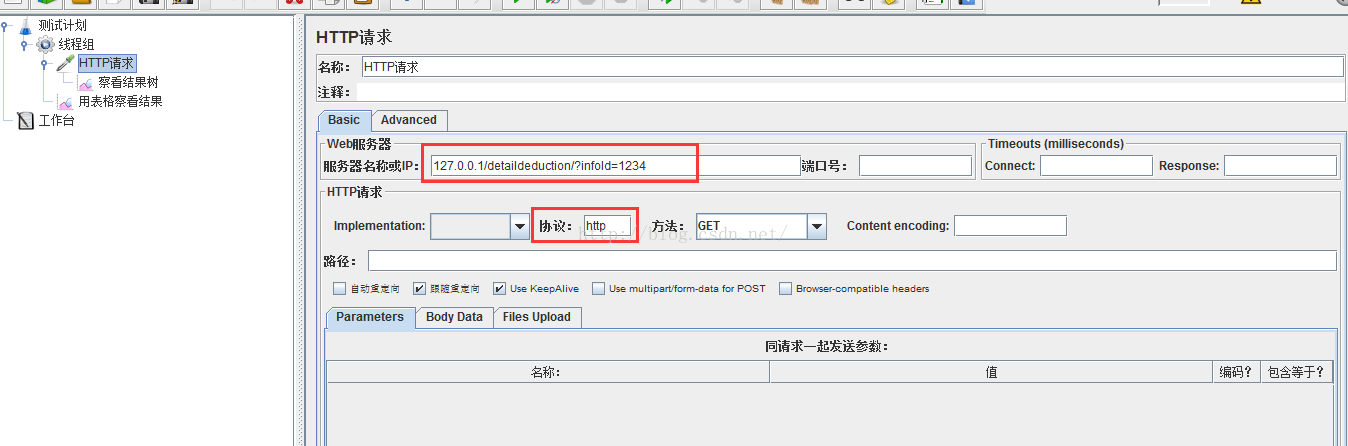
术语：

线程组：测试里每个任务都要线程去处理，所有我们后来的任务必须在线程组下面创建。可以在“Test Plan（鼠标右击） -> 添加  ->Threads(Users) -> 线程组”来建立它，然后在线程组面板里有几个输入栏：线程数、Ramp-Up Period(in seconds)、循环次数，其中Ramp-Up Period(in seconds)表示在这时间内创建完所有的线程。如有8个线程，Ramp-Up = 200秒，那么线程的启动时间间隔为200/8=25秒，这样的好处是：一开始不会对服务器有太大的负载。

取样器（Sampler）：可以认为所有的测试任务都由取样器承担，有很多种，如：HTTP请求。

断言：对取样器返回的请求结果给出判断是否正确。

monitor：它的功能是对取样器的请求结果显示、统计一些数据（吞吐量、KB/S……）等



接口性能测试：

参考：<http://www.ltesting.net/ceshi/open/kyxncsgj/jmeter/2015/0420/207928.html>

jmeter和junit结合测试

参考：<http://blog.163.com/bobile45@126/blog/static/96061992201631214722355/>

Do you need to use JUnit in your testing processes? To answer this question, let’s take a look first at unit testing（单元测试）.

[Unit testing](http://en.wikipedia.org/wiki/Unit_testing) is the lowest resolution of testing （最基本的解决方案）in the software testing lifecycle. When you run unit tests, you need to take the smallest piece of testable functionality in the application, isolate （独立）it from the other code and determine whether its behaviour matches expectations（行为符合预期）. This enables you to verify that small “units” of the application under test functionality work fine - before you start building up larger modules.

[JUnit](http://junit.org/) is by far the most popular unit testing framework for the Java language. According to a  recent survey, more than 30% of GitHub projects use JUnit for unit testing. JMeter developers and contributors also use JUnit to verify that a new feature or bug fix won’t break any existing functionality.

So JUnit clearly has a big value for developers. But how do you use it?

If you’re already using JMeter, you can use its [JUnit Request](http://jmeter.apache.org/usermanual/component_reference.html#JUnit_Request) sampler. This sampler can  execute individual（独立执行） JUnit test cases, enabling you to:

Run JUnit test cases in multiple threads（在多线程里使用junit测试）

Apply JMeter features like:（调整jmeter的功能如下）

* Logic Controllers
* Timers
* Assertions
* Listeners

Control the test execution order and conditions to build an advanced test plan

控制测试的执行顺序和条件来创建一个高级的测试计划

In this article, I’m going to highlight the most common use cases, show you how to perform configurations（配置）, reveal（揭示） where you should put your JUnit tests .jar files, and provide tips on extended scenarios（情景）.

Running Existing JUnit Tests

Feeding JUnit .jar Files to Jmeter（向jmeter中添加junit的jar文件）

The best place to put your .jar file(s) containing JUnit test cases is in the JMETER\_HOME/lib/junit folder. Just drop your .jar file into this folder, restart JMeter, and add a JUnit Request Sampler to your Test Plan. JMeter scans this folder and picks up everything that looks like a JUnit test case - so you should be able to see your JUnit Test Class names and associated（关联） methods in the dropdowns（下拉菜单）.

You can also “tell” JMeter to look into additional locations（额外的文件路径） via the “user.classpath” property（通过user.classpath属性）. This property lives in the “user.properties” file under the /bin folder of your JMeter installation. It can take the following values:

* Single jar file: user.classpath=/Projects/junit/test1.jar
* Multiple jar files: user.classpath=/Projects/junit/test1.jar;/Projects/junit/test2.jar
* A folder: user.classpath=/Projects/junit

Any combination of the above. Individual classpath entries（实体） need to be separated by semicolon（分号）.

More information on various JMeter properties and ways to set and override them can be found in the [Apache JMeter Properties Customization Guide](http://blazemeter.com/blog/apache-jmeter-properties-customization)

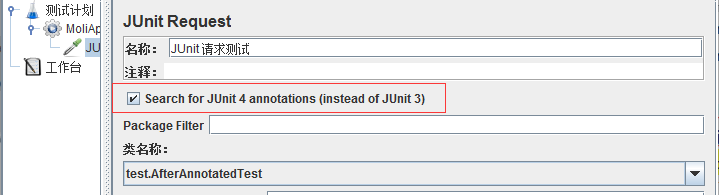
Remember that forward or back slashes in the classpath should correspond to the file separator of the operating system:

For Windows, the file separator is a backslash: \

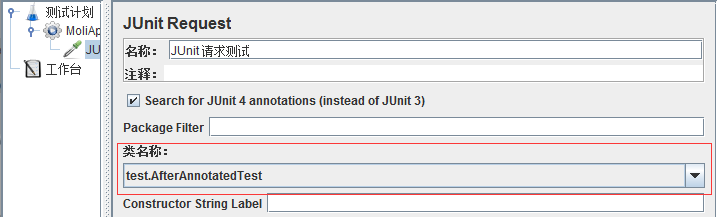
For Linux/Unix/MacOSX, the file separator is a forward slash: /

JMeter’s JUnit Request Sampler recognizes both JUnit3- and JUnit4-style classes and methods（jmeter可以识别junit-3和junit-4的测试类和方法）, but you need to ‘tell’ it which version to look for （需要告知jmeter查找那个版本的junit）via the “Search for JUnit 4 annotations （注释）(instead of JUnit3)” checkbox. When the box is checked, JMeter will look for JUnit4-style annotations and JUnit3 classes won’t be shown (and vice versa(反之同样) if the box isn’t checked).

For instance, I have jars with both JUnit3 and JUnit4 test classes. If I don’t check the box, JMeter will look for JUnit3 tests （默认选择junit-3的测试方法）and JUnit4 won’t be displayed:



And when the box is checked, JMeter is instructed to scan only for JUnit4 tests. As you can see, JUnit3 tests aren’t in the dropdown（junit-3的方法不会出现在下拉菜单里面）



Hopefully you now have all your classes and methods listed in JMeter’s JUnit Sampler. So what’s next?

The JUnit Sampler’s Control Panel Parameters （参数）- Explained!

 Before you can go much further, you’ll need to understand all the parameters of the JUnit Sampler’s control panel. Here’s a quick explanation of each:

Name—这个junit采样测试的名称

This is for the name of the JUnit Sampler, just like any other Sampler or Config Element or Logic Controller. It’s better to give the sampler a proper, unique name to ensure you’ll be able to distinguish it from others when running your [performance test results analysis](http://blazemeter.com/blog/how-analyze-results-load-test-using-blazemeter-0)（运行分析测试结果）.

Comment—注释，对这个采样测试的说明

Here you can enter arbitrary （随意）comments - either for yourself or for colleagues if you’re working in collaboration. This normally involves （包含进）a short summary of what the sampler does.

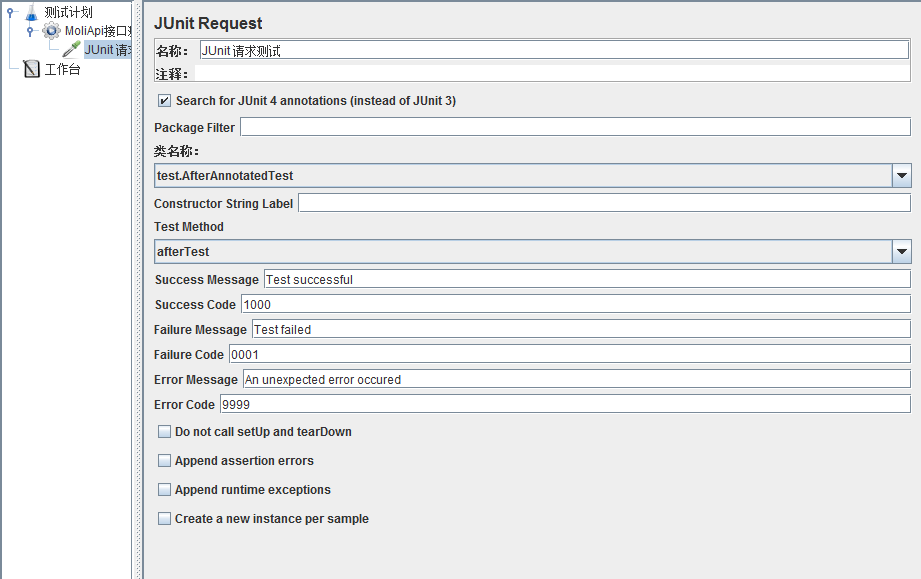
Search for JUnit4 Annotations—是否只查找/扫描junit-4测试

I’ve already covered this above. It tells JMeter whether to look for JUnit3 or JUnit4 tests.

Package Filter—包筛选

By using the Package Filter, you can limit JMeter’s output of JUnit test classes to just one（能够限制jmeter对junit的测试输出到仅仅一个文件）. This will contain the pattern provided through the input（包含提供的输入模式）.

By default JMeter lists all the classes-默认的jmeter会列示所有的类



Remember that the Package Filter only works for packages, not for class names（包筛选器只对包起作用，但是对类不起作用）. So “woolfel” will work but “Dummy” won’t.

Classname

This parameter is pretty self-explanatory. Just select the class where the test case you want to run from lives.

Constructor String Label

If your JUnit test class has a constructor which accepts a single String, you can set its value with this parameter. If your JUnit test class doesn’t declare such a constructor, the JUnit Sample will try to look for an empty constructor.

Test Method

This is the most important part of the JUnit  sampler. Based on the “Classname” selection, the Test Method drop-down shows all the available methods. Only one JUnit test method can be run by the JUnit Sampler at a time.