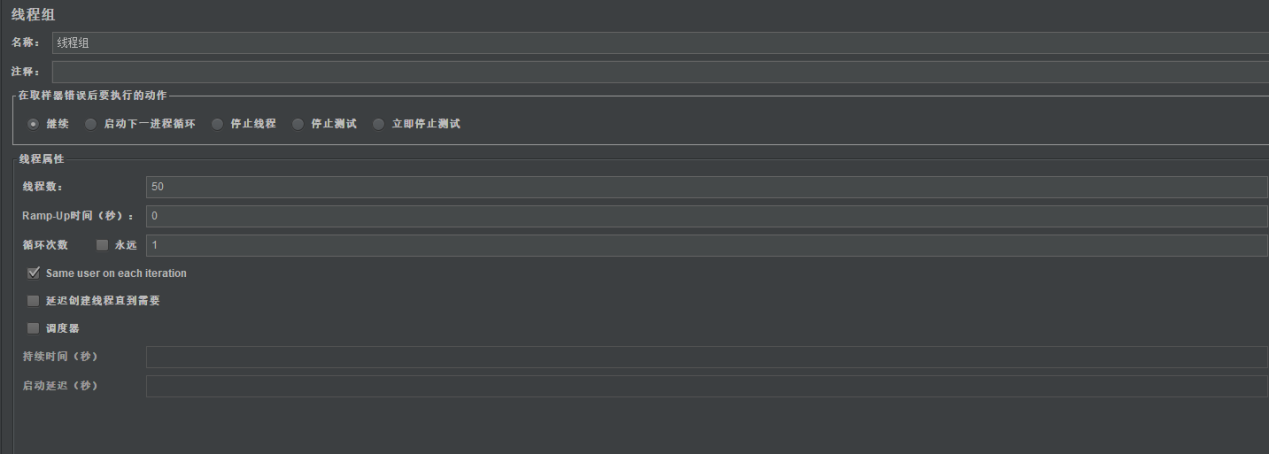
Jmeter并发测试报告

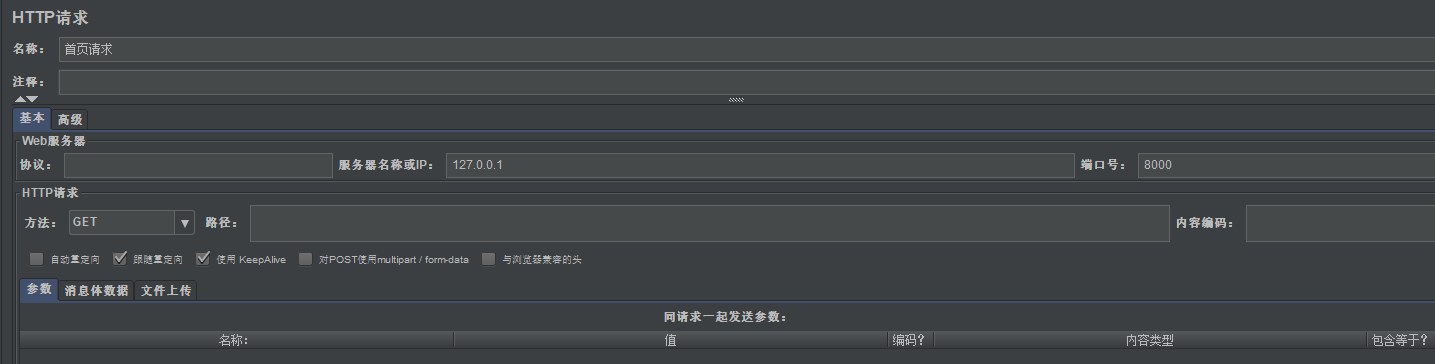
本次测试分为三类：未登陆前页面访问，登录操作，登陆后页面访问，目标是测试他们的并发情况

1. 未登陆前页面访问

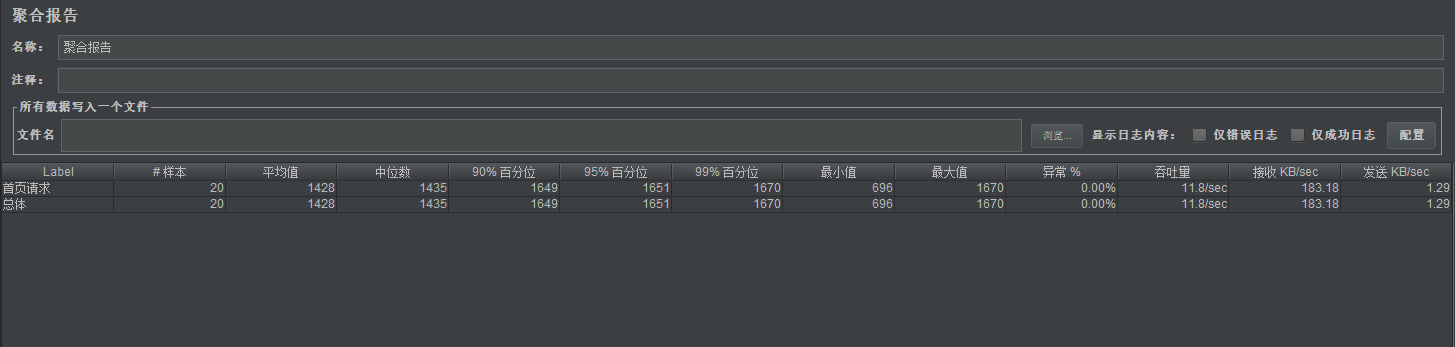
这一部分选取网站首页为测试对象。相关参数如图



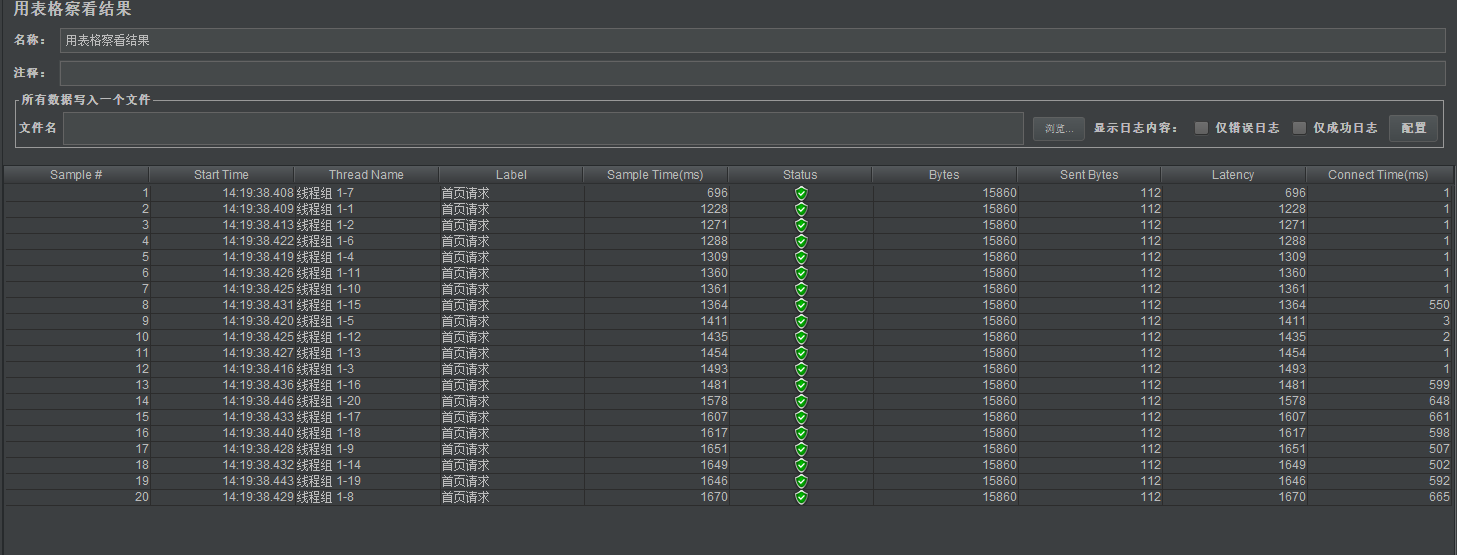
测试并发量为20个线程，设置ramp-up为0表示在0秒内启动20个线程，也就是同时访问，从而模拟并发情况，循环测试设置为默认值为1，其余两个测试的基础参数也是如此，之后不再赘述。



HTTP请求参数如图，ip为本机地址127.0.0.1，端口为项目端口8000，由于默认访问首页，因此无需添加其他路径和传递参数。



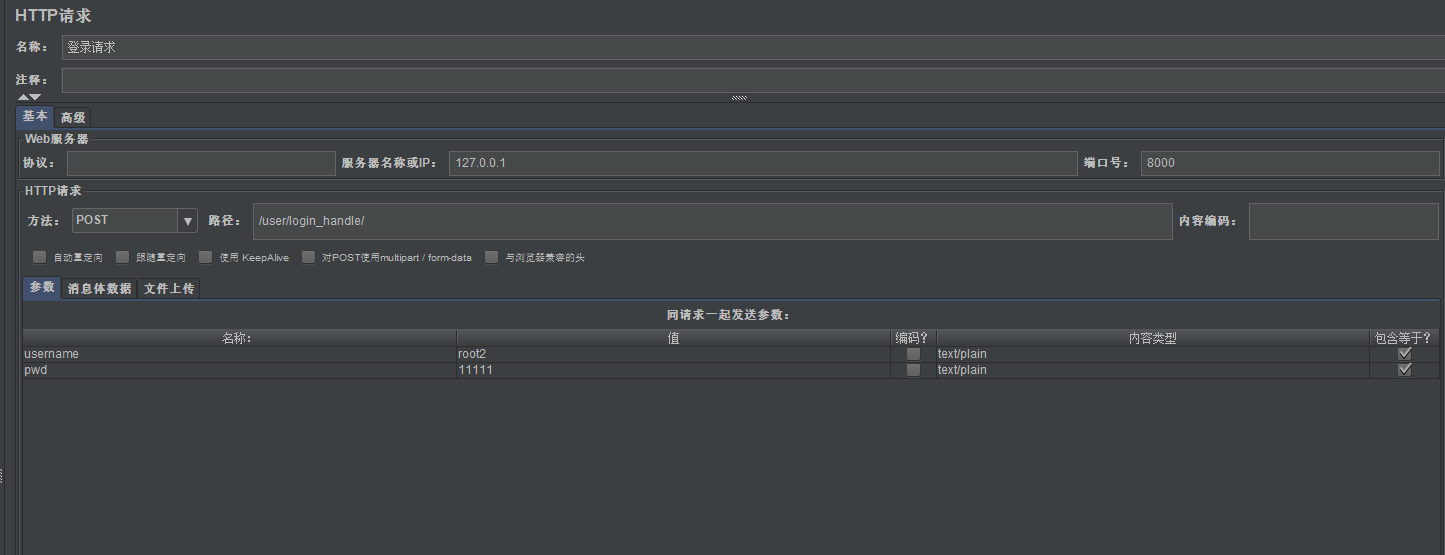
首页并发访问聚合报告如下，可以看到没有出现异常，90%的请求响应时间在1秒左右，是一个用户勉强可以接收的等待时间

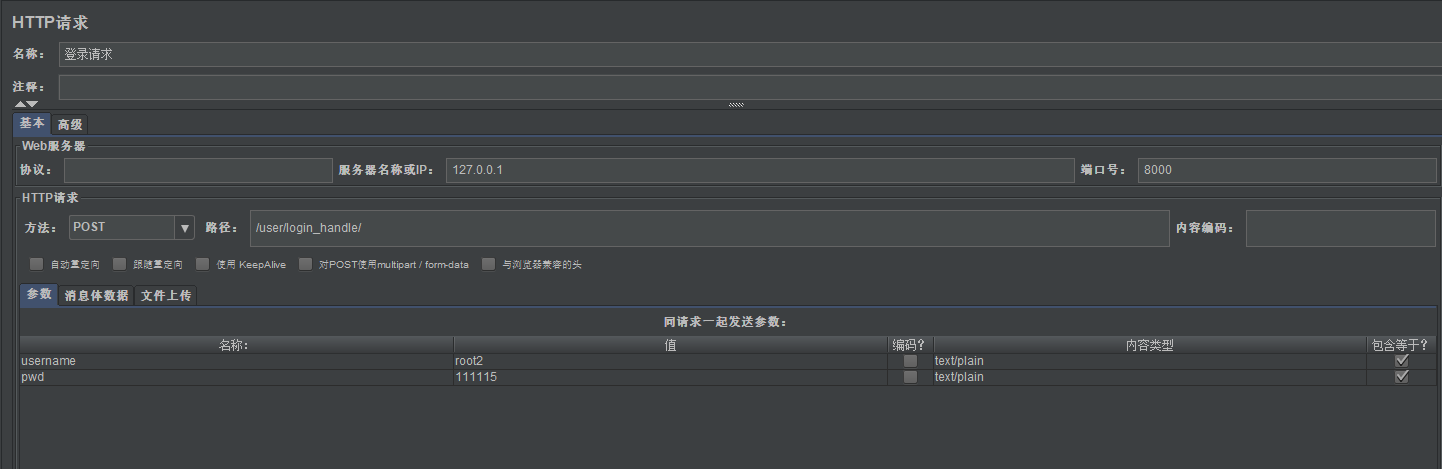


再通过表格可以看到，20个线程的确是几乎同时启动的。

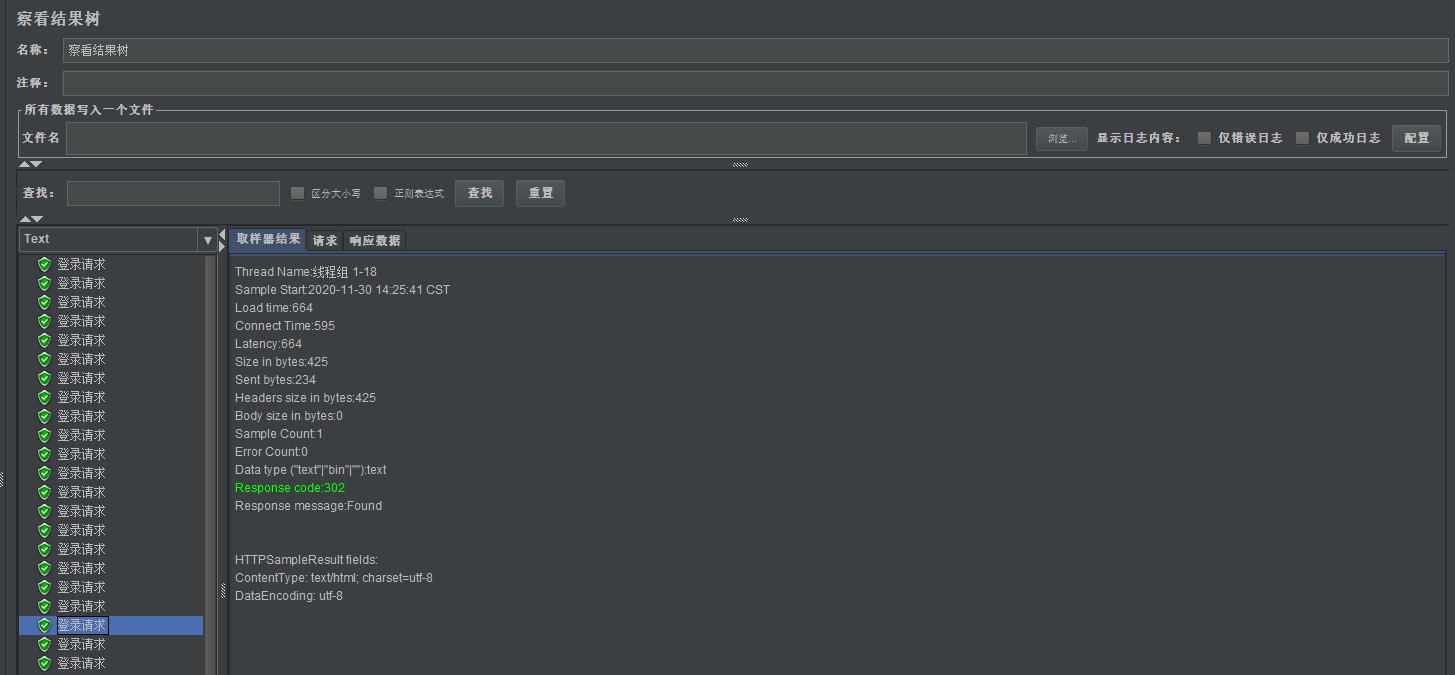
1. 登录请求模拟

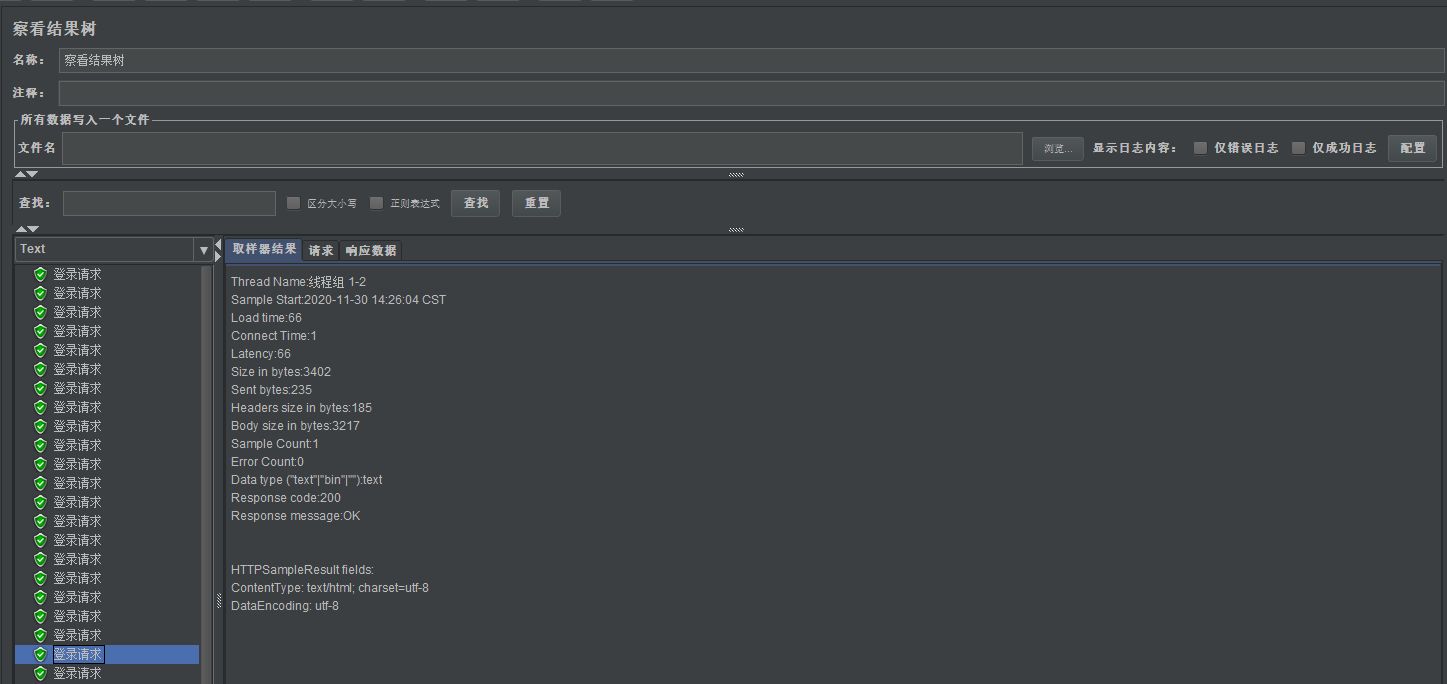
登录请求和第一个的区别主要在于需要传递参数，并且对于是否请求成功不能再只用简单的是否返回了页面来判断，因为正确的登录应该返回首页，而错误的登录则会返回登陆页，具体操作如下





首先是http请求设置，注意这里的路径是登录请求处理的url而不是登录界面的url，原因很容易理解，我们模拟的是登录这个操作，而不是访问登录界面这个操作。然后传参方式和参数名及参数值则是根据具体项目来填的，如图1所示是正确用户名密码，图2则是错误密码，这是为了接下来选择断言器条件。

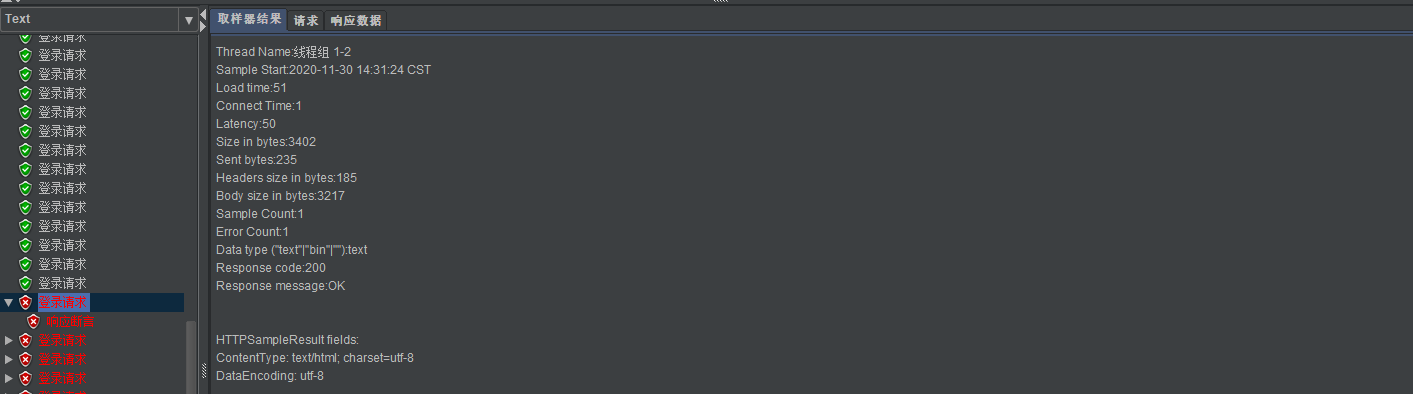




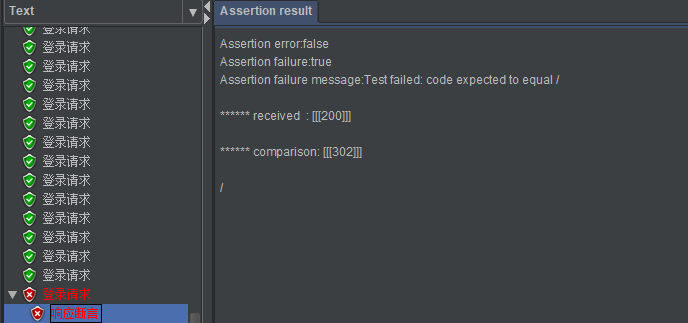
有关断言设置的内容，让我们来看如图两个例子，这是使用默认断言，也就是通过是否成功返回了页面来判断请求的成败与否，图1是使用正确用户名密码的情况，图2是使用错误用户名的情况，正如本部分开头所说，正确登录会跳转回首页，而错误登录则没有跳转，对应的具体区别在于返回状态码，跳转是302，而正常响应是200，显然默认断言无法区分这两种情况，因此我们要添加断言器



如上分析，我们添加的断言器的条件是判断相应代码，只有在相应代码为302时才认为请求成功，

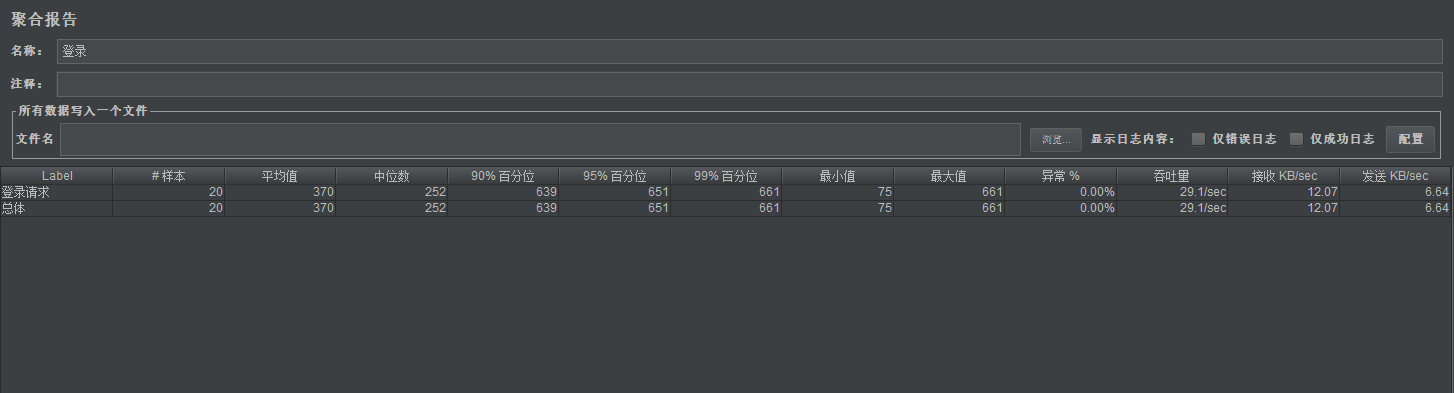


如图是添加断言器后的结果，可以看到，尽管返回码为200，jmeter仍然认为请求失败，这就达到了我们的目的



失败原因写的也很清楚。

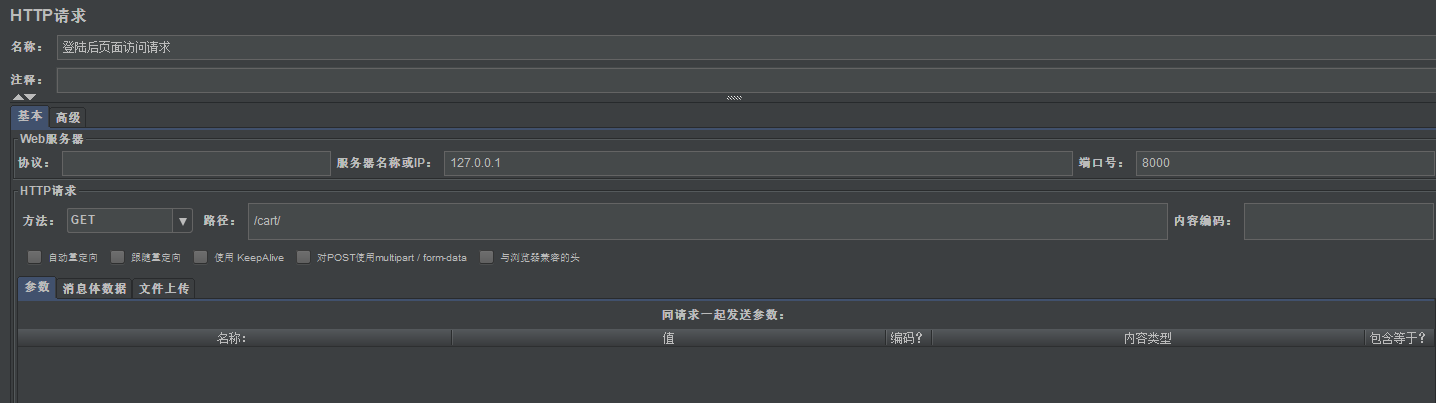
最后，让我们来测试使用正确的用户名密码登录的并发情况。



结果如图，可以看到对于并发登录请求的响应速度明显快过首页访问，这显然是对于用户而言是很友好的。

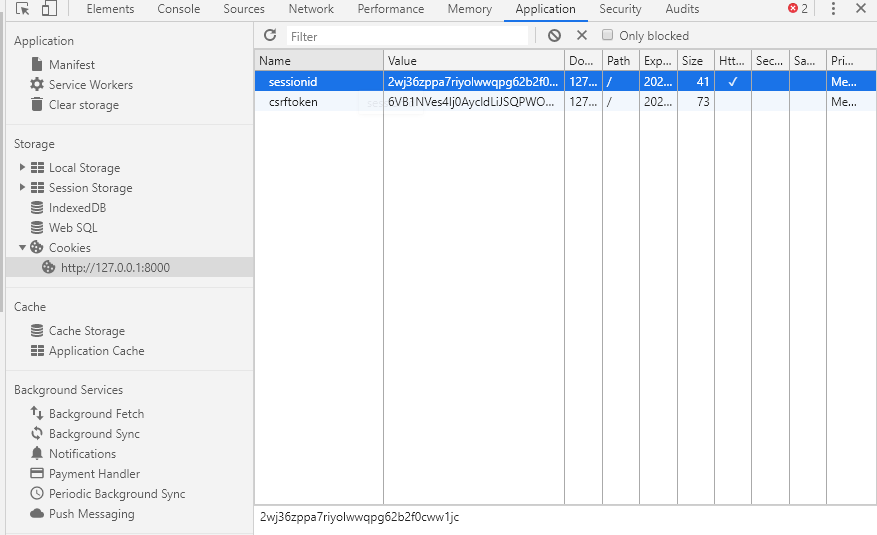
1. 登陆后页面访问

这一部分的要点在于模拟登陆后的请求，因此需要用到的是cookie组件。具体操作如下。

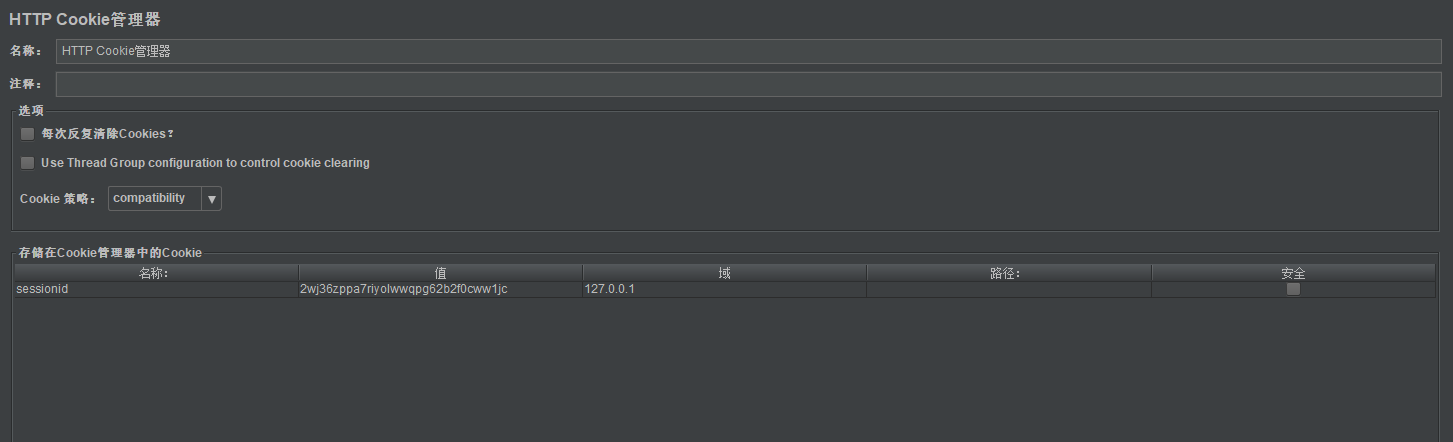


首先是访问路径，cart是用户购物车的情况，在我们的项目中，这个路径只有在登陆后才能正确访问，否则会跳转到登录界面，这也是我们后续判断的一个依据

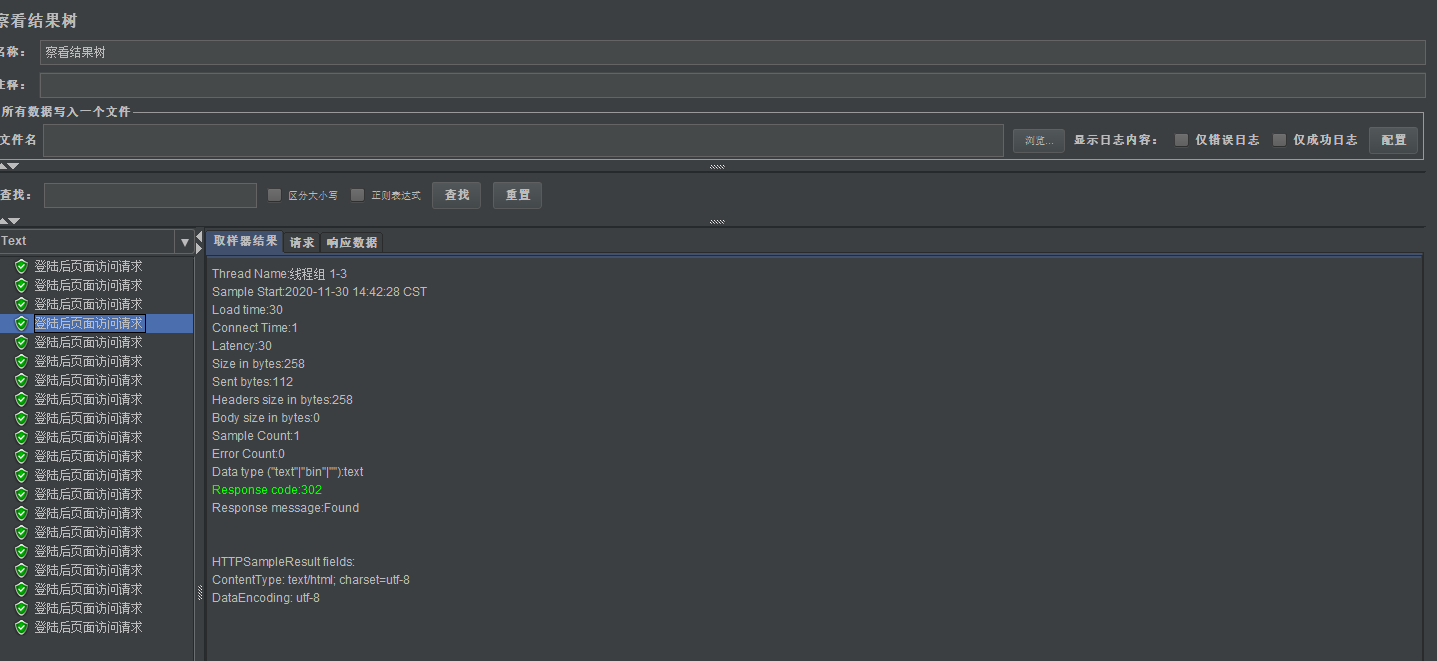
网站如何判断一个请求是否是由一个已经登陆的用户发起的呢，在我们的项目中，是通过cookie来实现的，如图



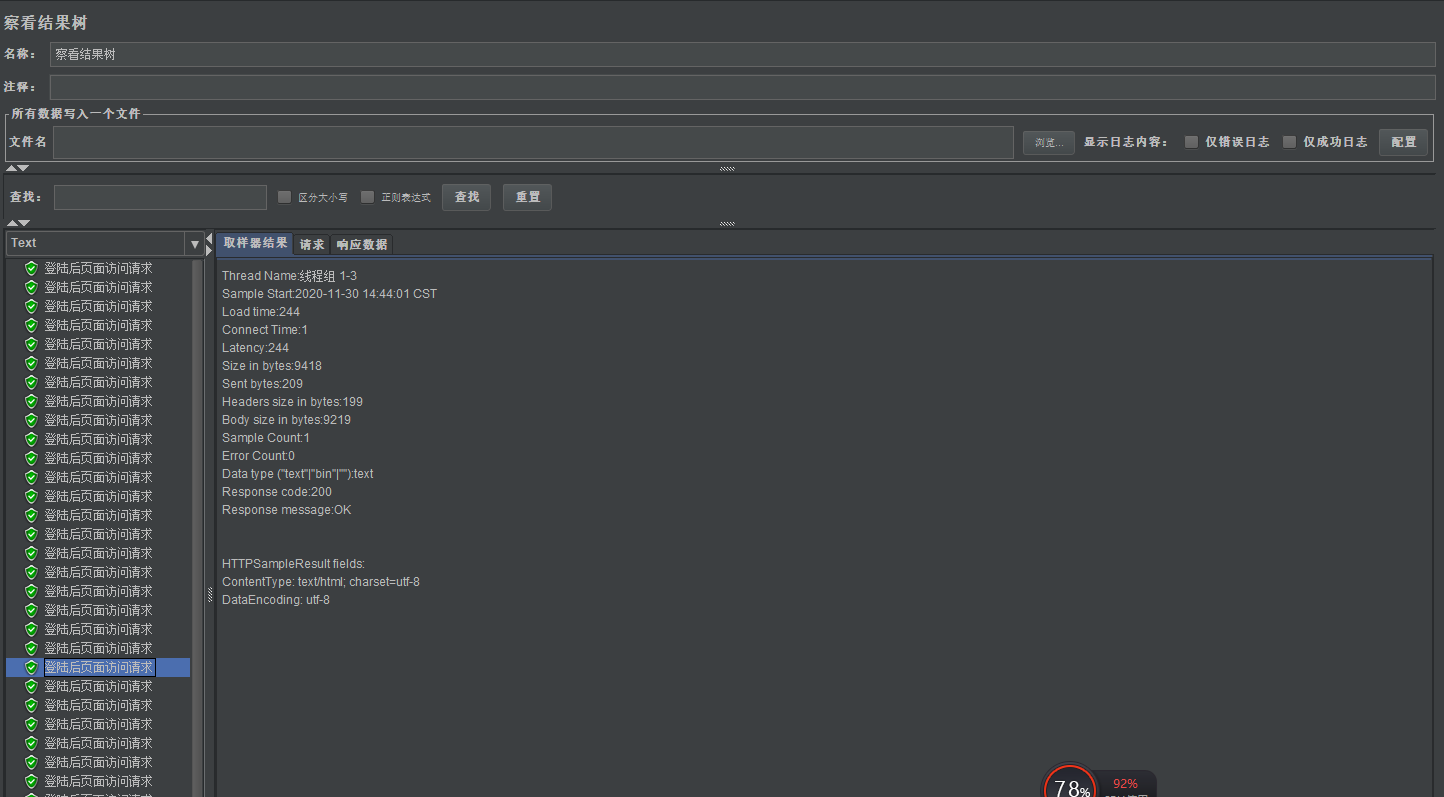
按f12打开开发者工具，可以看到这个值，正是通过它来进行判断的，因此我们要在jmeter请求中添加cookie管理工具



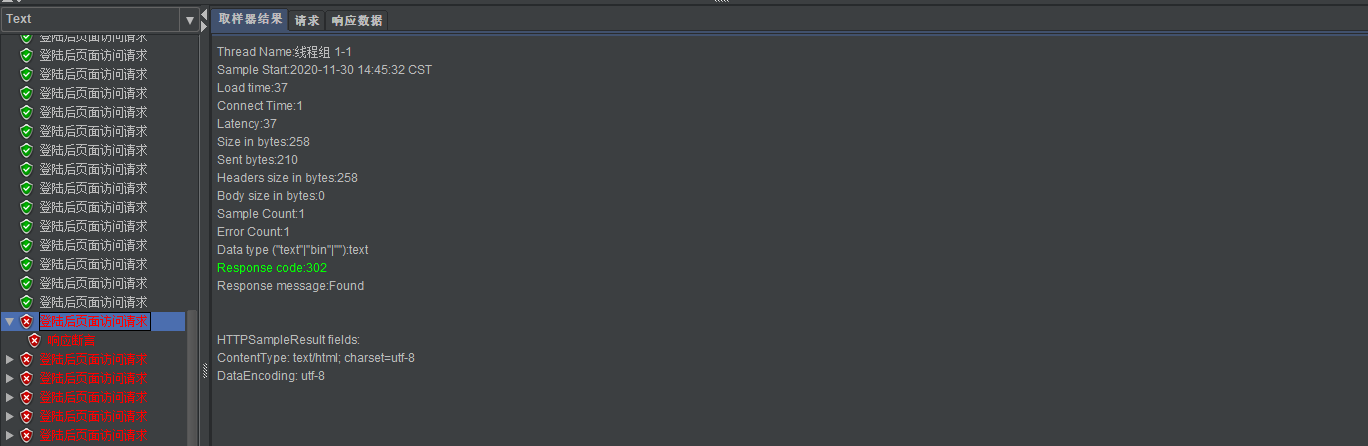
设置如图，要注意的是一定要填写域字段为网站ip，否则无法生效



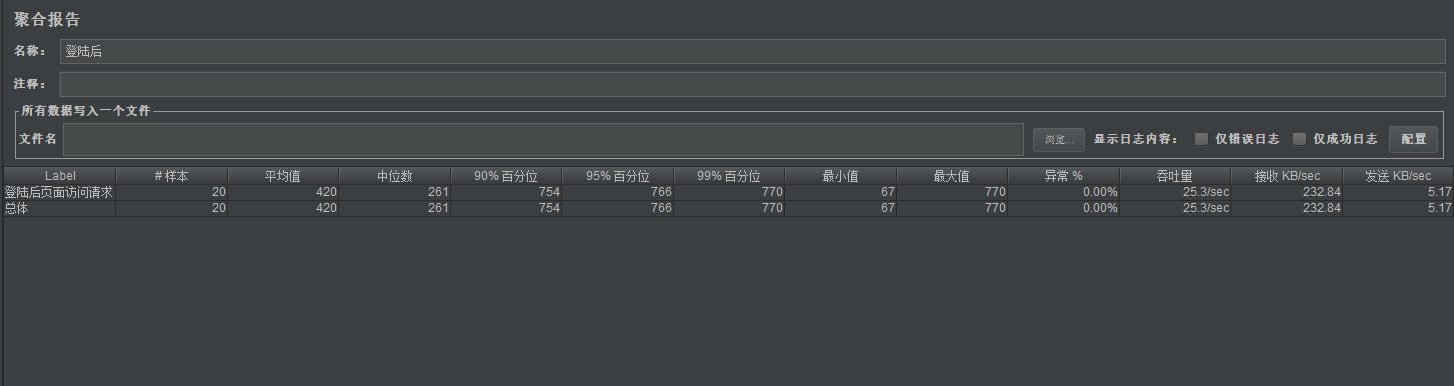
如图是我们不设置cookie直接请求的结果，可以看到返回码为302，说明跳转到了登录界面。



而添加了cookie后的访问请求返回码则是200，因此我们这一部分的断言器和第二部分相反：设置返回码为200时才成功。



设置断言器后不携带cookie访问的结果。



正常的访问登陆后页面的结果如图，可以看到大致和登录请求处理的量级一致。

总结

通过实验结果，我们可以看到对于登陆前首页的访问明显慢于其他两个，推测原因是首页需要加载大量图片信息，从而导致了响应速度慢这一结果。以上便是jmeter测试的全部内容。