

TestTomcat示例

2021年10月4日 11:08

我们用JMeter对TestTomcat进行测试（Tomcat9.0默认使用NIO）

```
12  * @author yang qib  
13  * @date Created in 2021-6-1 22:12  
14  **/  
15  @RestController  
16  @RequestMapping(value = "/testtomcat", produces = "application/json;charset=UTF-8")  
17  public class TestTomcatController {  
18      @GetMapping("/{latency}")  
19      public Object sleepthread(@PathVariable long latency){  
20          try {  
21              TimeUnit.SECONDS.sleep(latency);  
22          } catch (InterruptedException e) {  
23              e.printStackTrace();  
24          }  
25          return ResponseUtil.ok();  
26      }  
27  }  
28  }
```

这个项目只有一个Controller，代码也很简单，是接收到一个get请求后，睡眠指定的时间，这个时长可以通过url传进来

```
application.yaml  
1  management:  
2  endpoints:  
3  web:  
4  exposure:  
5  include: "*"   
6  server:  
7  port: 8080  
8  tomcat:  
9  threads:  
10     #最大工作线程数  
11     max: 200  
12     #最小工作线程数  
13     min-spare: 100  
14
```

我们在application.yaml下对线程数进行了一些配置，最大线程数200，最小线程数100

因为我们要看应用服务器的运行的状况，如线程、内存等，所以我们需要用一个JConsole工具来远程看服务器上JVM的情况。所以我们在跑起项目的时候带一些参数

在runjava.sh中，我们写了一些参数

```
java -Xms1g -Xmx1g -Xss512k \  
-Dcom.sun.management.jmxremote.port=3214 \  
-Dcom.sun.management.jmxremote.ssl=false \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-jar $1
```

-Xms1g: JVM的堆的最大空间为1G

-Xmx1g: JVM的堆的最小空间为1G

-Xss512K: 每个线程的堆栈空间为512K

这三个参数是最基本的参数，决定了JVM运行起来是所占的虚拟空间

后面的参数是JConsole为了远程连接定义的参数

-port=3214：远程连接的端口是3214

-ssl=false：不需要加密

-authenticate=false：不需要用户名和密码

使用./runjava.sh target/xxx.jar，可以使用对应的参数来运行jar包（运行之前记得更改runjava.sh的权限为777）

一些需要注意的部分：

1. 使用jmeter建立测试计划之后，最好在另外一台服务器上去执行测试计划
如：在slave1上建立TestPlan，保存为TestPlan.jmx，然后使用scp将jmx拷贝到slave3上
2. 使用JConsole去看目标服务器的状况：之前在runjava.sh中定义了怎么样去连接，端口为3214
3. slave3上运行测试计划：jmeter -n -t TestPlan.jmx -l result.jtl
-n --nogui：以非gui模式（命令行模式）运行jmeter
-t --testfile：用来运行的jmeter脚本
-l：用来记录结果的文件
（如果遇到权限问题，前面加上sudo，或者用root用户运行）
4. 我们可以使用jmeter直接查看result.jtl，但是看着不方便，所以我们将其使用scp拷回到slave1使用html方式查看
5. 在slave1中，使用jmeter -g result.jtl -o result将其转换为html报告
jmeter -g：固定的
-o：指定输出文件
6. 在生成的result文件夹中，打开index.html即可看到相关报告