

Jmeter 性能测试利器（入门指南）



摘要：本文简单介绍Jmeter性能测试工具的使用，包括组件的使用和常见的业务场景的接口测试，后端接口使用的springboot做服务端。

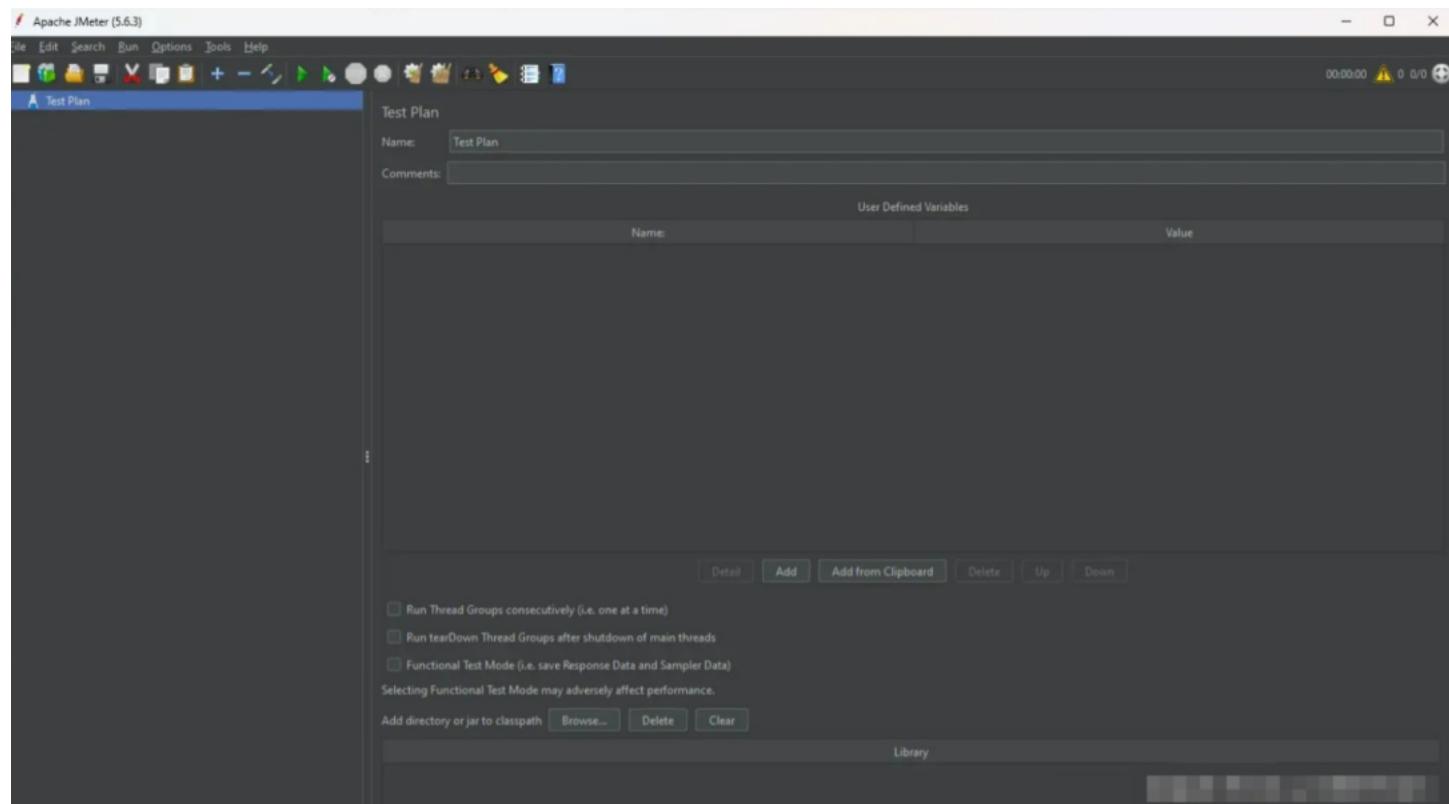


我下载的是<https://dlcdn.apache.org//jmeter/binaries/apache-jmeter-5.6.3.zip>

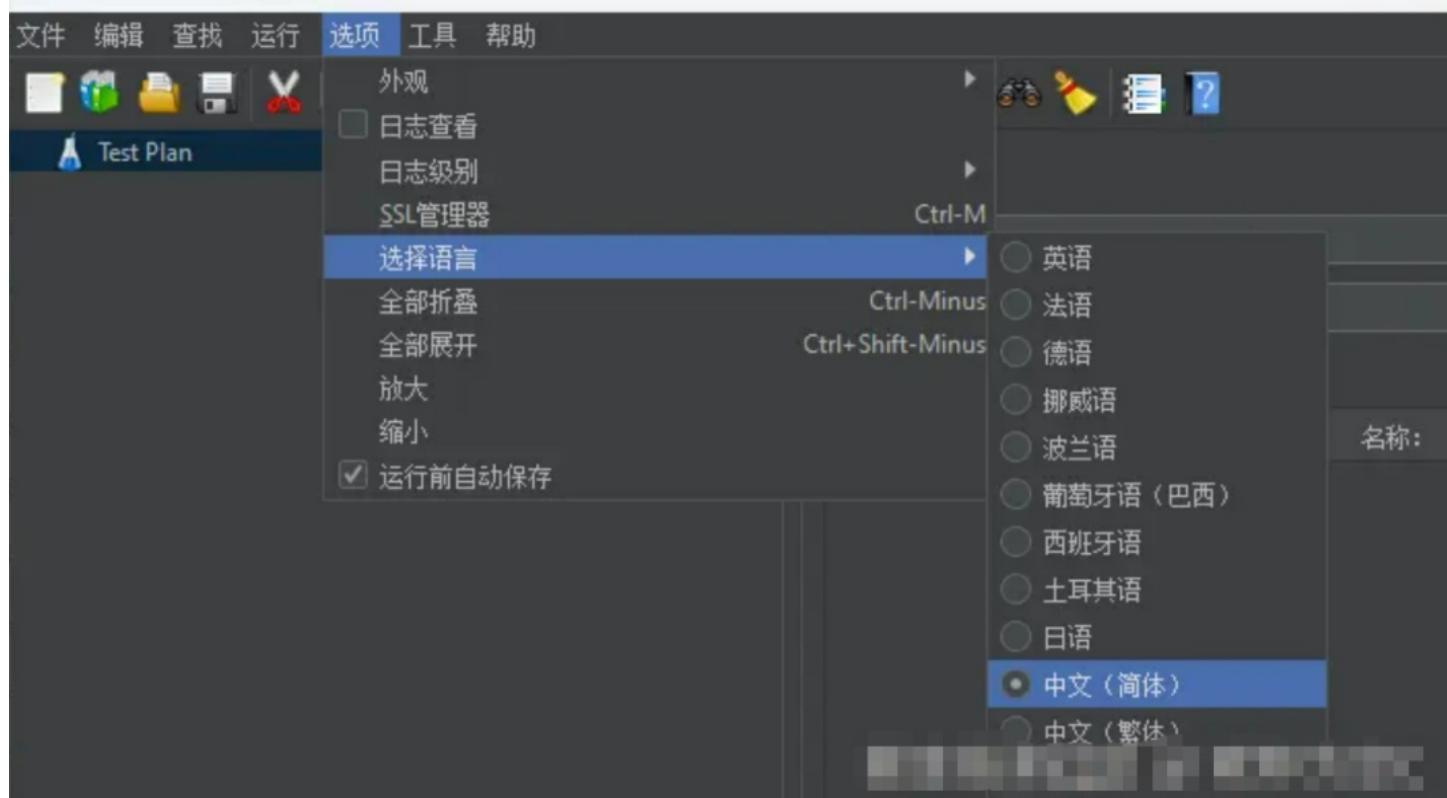
下载地址：jmeter.apache.org/download_jm...



先解压，然后切换到bin目录双击jmeter.bat



设置中文语言



永久设置方案，配置jmeter.properties，在39行左右。

```

1 #Preferred GUI language. Comment out to use the JVM default locale's language.
2 #language=en
3 language=zh_CN

```

▶ 创建一个测试计划 “”

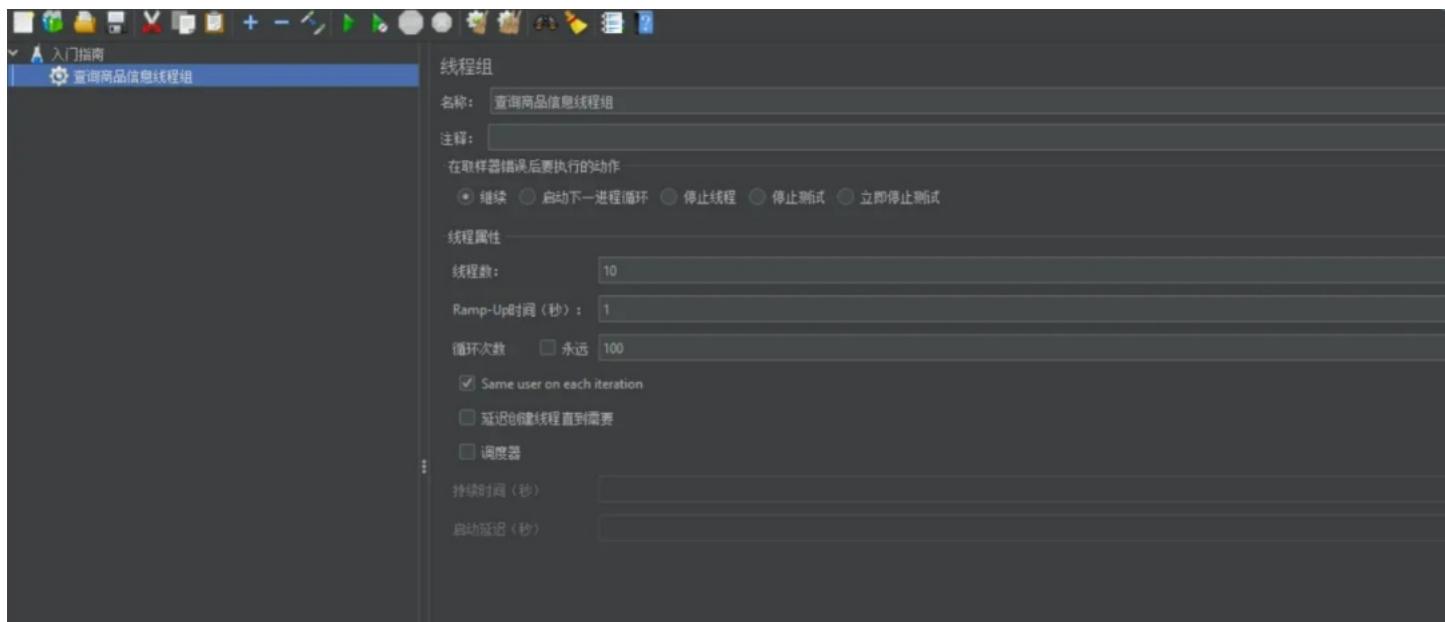
创建一个计划并保存



▶ 第一个测试案例 “”

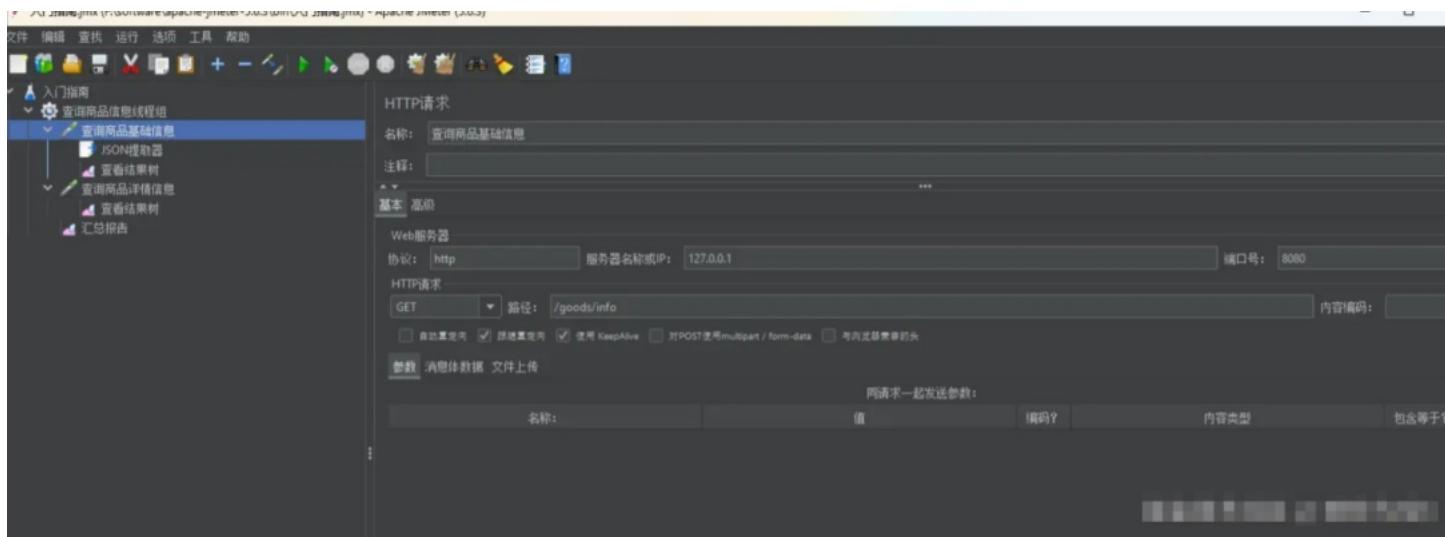
创建一个线程组

线程组就理解为用户，多少个用户同时访问，线程组下面设置的变量，只要没有设置为全局的，都是当前线程共享的，不会跨线程。



添加取样器 HTTP 请求

就是配置你调用的接口，如果有请求头一类的，你再添加【配置元件/HTTP请求头管理器】



添加JSON提取器

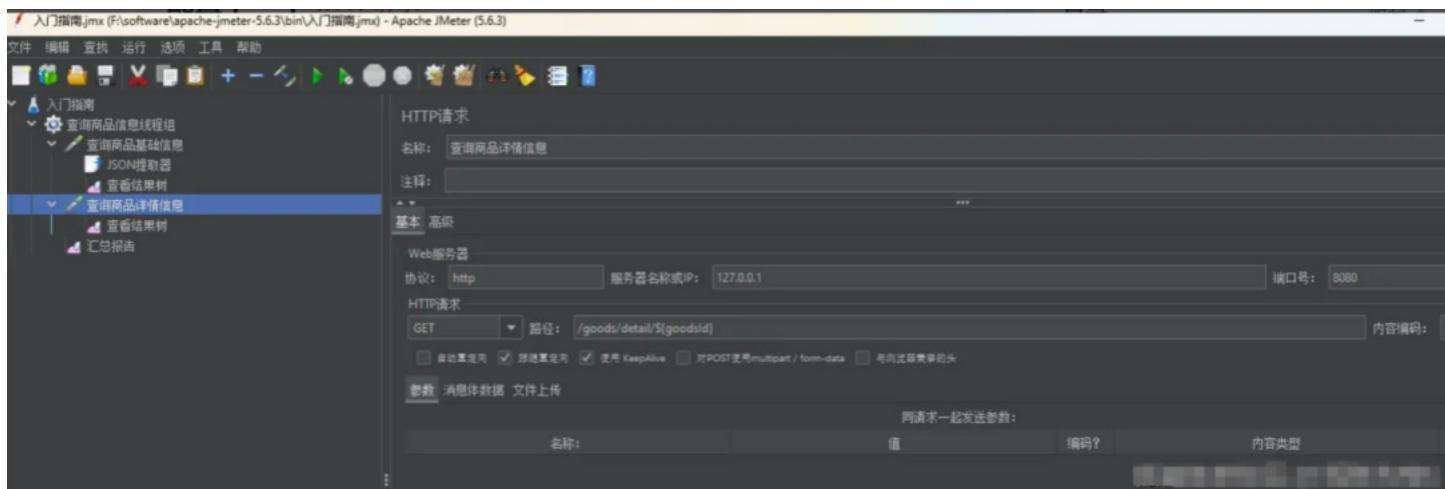
因为我们要获取数据给下一个接口，提取了下面的data属性。

```
1
2 {
3     "code": 200,
4     "data": 1755763130398,
5     "message": "success"
}
```



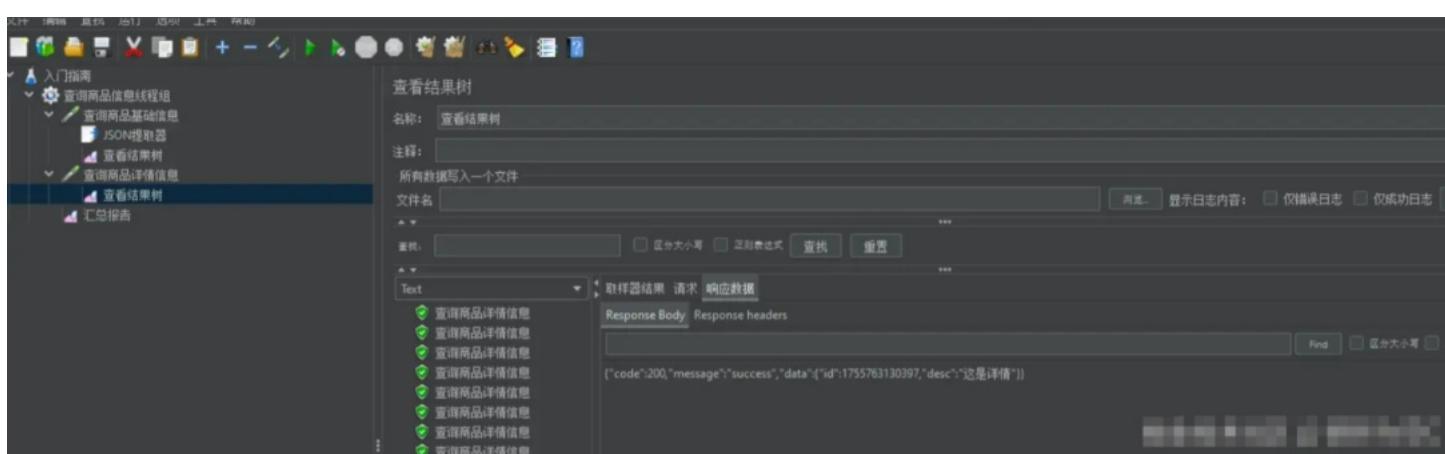
配置下一个HTTP请求

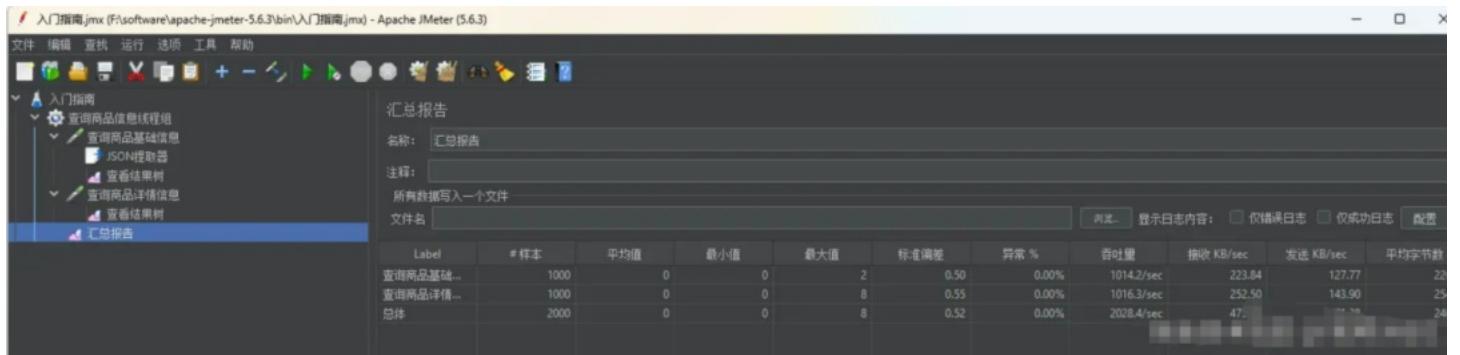
可以使用上面提取的参数



添加监听器

- **查看结果树** 添加到取样器下面的；
- **汇总报告** 可以添加到取样器下也可以添加到线程组下面查询总体情况；





◆ 常用处理器使用说明 ◆

BeanShell

它功能非常强大，通过脚本转换出来新的值，传递给全局或者当前线程。

vars：当前线程变量；

通过 **vars** 对象的方法访问：

- **vars.get("key")**：获取字符串值（若键不存在返回 null）。
- **vars.put("key", "value")**：设置字符串值。
- **vars.getObject("key")**：获取任意类型的对象（需手动转换）。

props：全局共享

- **props.get("key")**：获取字符串值（若键不存在返回 null）。
- **props.setProperty("key", "value")**：设置字符串值（等价于 `props.put("key", "value")`）。
- **JMeter 函数**：`_${_P(key)}`（等价于 `props.get("key")`）、`_${_property(key, defaultValue)}`（若键不存在返回默认值）。

场景	推荐使用 vars 还是 props?	原因
线程内临时数据传递 (如同一个请求的多个后置处理器)	vars	线程局部，无需全局共享，避免污染全局空间。
跨线程组传递动态数据 (如登录 Token)	props	全局共享，其他线程组可直接读取。
分布式测试中统一配置 (如环境标识)	props	所有节点共享同一配置，无需手动同步。
存储复杂对象 (如自定义 Java 对象)	vars	支持任意类型，props 仅支持字符串。
测试生命周期内的全局状态 (如测试开始时间)	props	随 JMeter 实例存在，覆盖整个测试周期。

现场变量转换为全局变量

```
1 props.setProperty("total_requests", vars.get("goodsId"));
```

编写 MD5 加密逻辑，将原始密码替换为加密后的值

```
1 import java.security.MessageDigest;
2 // 原始密码 (可从变量获取，如 ${raw_pwd})
3 String rawPwd = vars.get("raw_pwd");
4 // MD5 加密方法
5 MessageDigest md = MessageDigest.getInstance("MD5");
6 byte[] digest = md.digest(rawPwd.getBytes());
7 StringBuilder sb = new StringBuilder();
8 for (byte b : digest) {
9     sb.append(String.format("%02x", b & 0xFF));
10 }
11 String encryptedPwd = sb.toString();
12 // 替换请求参数中的密码 (假设请求体中有 password 字段)
13 // 方法 1：修改变量 (后续请求体用 ${encrypted_pwd} 调用)
14 vars.put("encrypted_pwd", encryptedPwd);
15 // 方法 2：直接修改请求体 (需结合请求体的变量名，如请求体为 JSON)
16 // prev.setBodyData("{\"username\":\"test\", \"password\":\"\" + encryptedPwd + \"\"}");
```

51Testing 征稿啦！



征稿方向

AI测试	游戏测试
车载测试	测试技术及工具应用
嵌入式测试	职场经历、简历面试技巧

投稿方式

1. 稿件发送邮箱: editor@51testing.com
2. 扫描右方二维码添加微信, 备注“投稿”



公众号 - 51Testing软件测试网

链接 :

<https://juejin.cn/post/7560539198449598479>

本文为51Testing经授权转载, 转载文章所包含的文字来源于作者。如因内容或版权等问题, 请联系51Testing进行删除

