

Jmeter方法

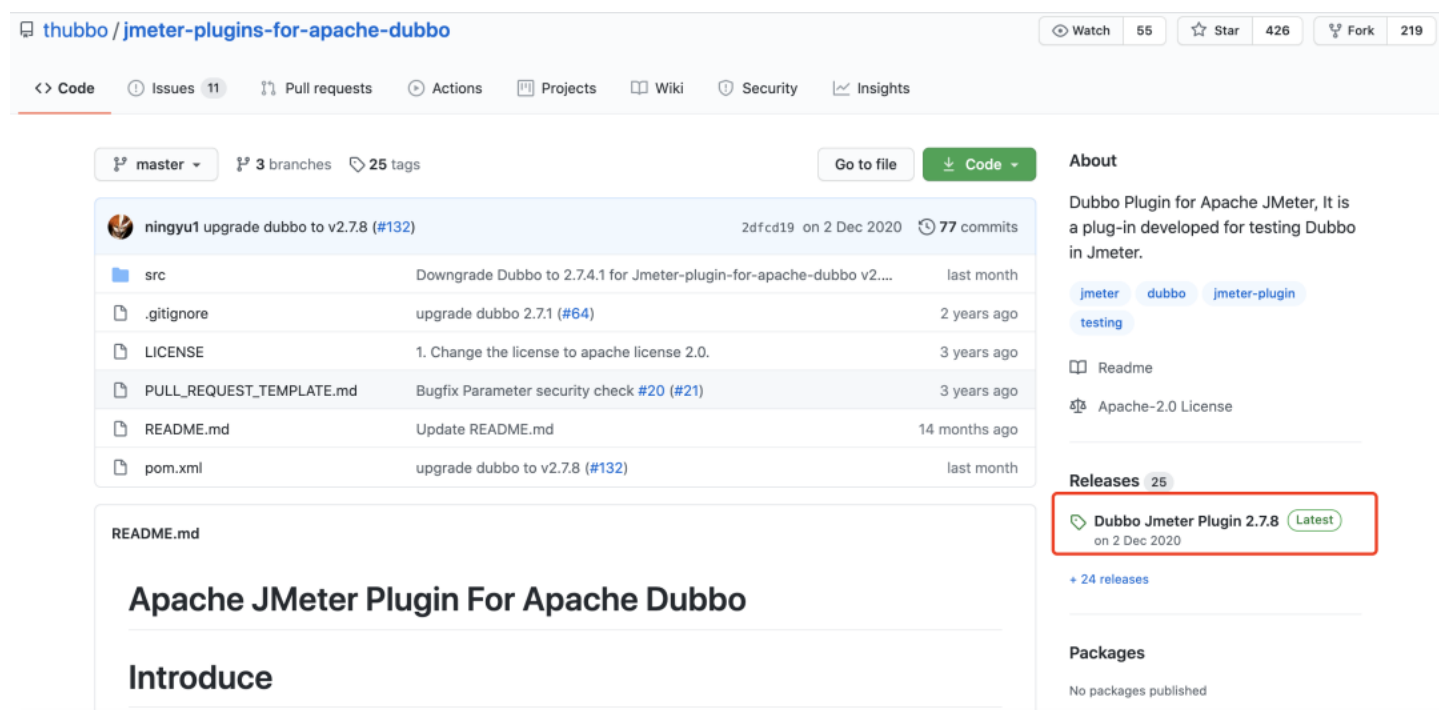
JMeter 默认是不支持 Dubbo 接口测试的，但是我们可以通过拓展的插件或 jar 包实现此功能。

JMeter 插件拓展

1.1 插件下载

测试 Dubbo，我们需要下载 Dubbo 的插件，在 Apache 的 Dubbo 插件 GitHub 中可以找到：

<https://github.com/thubbo/jmeter-plugins-for-apache-dubbo>



thubbo / jmeter-plugins-for-apache-dubbo

Watch 55 Star 426 Fork 219

<> Code Issues 11 Pull requests Actions Projects Wiki Security Insights

master 3 branches 25 tags

Go to file Code

ningyu1 upgrade dubbo to v2.7.8 (#132) 2dfcd19 on 2 Dec 2020 77 commits

File	Commit Message	Time Ago
src	Downgrade Dubbo to 2.7.4.1 for Jmeter-plugin-for-apache-dubbo v2....	last month
.gitignore	upgrade dubbo 2.7.1 (#64)	2 years ago
LICENSE	1. Change the license to apache license 2.0.	3 years ago
PULL_REQUEST_TEMPLATE.md	Bugfix Parameter security check #20 (#21)	3 years ago
README.md	Update README.md	14 months ago
pom.xml	upgrade dubbo to v2.7.8 (#132)	last month

README.md

Apache JMeter Plugin For Apache Dubbo

Introduce

About

Dubbo Plugin for Apache JMeter, It is a plug-in developed for testing Dubbo in Jmeter.

Readme

Apache-2.0 License

Releases 25

Dubbo Jmeter Plugin 2.7.8 Latest on 2 Dec 2020

+ 24 releases

Packages

No packages published

点击进入，这里我们选择当前的最新版本 2.7.8 进行下载。

新版改进：

1. Upgrade Dubbo version v2.7.8
2. 支持类型：Locale #124、LocalDateTime、LocalDate、LocalTime #126
3. 修复压测sample串接口问题 #122

下载

[jmeter-plugins-dubbo-2.7.8-jar-with-dependencies.jar](#)

Assets 2

Source code (zip)

Source code (tar.gz)

1.2 插件安装

插件下载好后只需要将 jar 包放入到 `jmeter` 的 `lib` 下的 `ext` 文件夹即可。

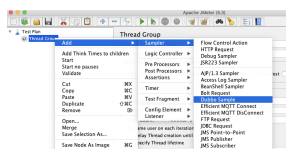
由于我的 JMeter 是使用 Mac 的 Homebrew 安装的，所以我的路径

为 `/usr/local/Cellar/jmeter/5.3_1/libexec/lib/ext`，具体的根据自己的安装路径寻找即可，如下可以看到我的 jar 包已放入。

```
1 % pwd
2 /usr/local/Cellar/jmeter/5.3_1/libexec/lib/ext
3 % ls | grep jmeter-plugins-dubbo-2.7.8-jar-with-dependencies.jar
4 jmeter-plugins-dubbo-2.7.8-jar-with-dependencies.jar
5
```

1.3 重启 JMeter

重启 `Jmeter` 后再采样器可，以看到 `Dubbo Sample` 的存在。



测试接口准备

这里在本地准备好了一个 Dubbo 的测试 demo 供演示使用，`provider`的方法如下，直接调用方法，返回一个用户信息列表：

```
1 public class UserServiceImpl implements UserService {
2
3     @Override
4     public List<UserAddress> getUserAddressList(String userId) {
5         UserAddress userAddress1 = new UserAddress(1, "杭州市西湖区", "1", "Tester1", "12345678", "Y");
6         UserAddress userAddress2 = new UserAddress(2, "杭州市余杭区", "2", "Tester2", "12345678", "N");
7         return Arrays.asList(userAddress1, userAddress2);
8     }
9 }
10
```

通过 `dubbo-admin` 可以查到当前的服务已经提供完成:



测试 Dubbo 接口

这里选择以直连的方式进行接口的测试，也就是利用了 Dubbo 的泛化调用的方式，不需要知道注册中心，只需要知道服务接口，方法名以及服务提供者的地址即可。关于 Dubbo 的泛化调用，后续再单独介绍。

3.1 修改配置

Dubbo Sample

Name:

Comments:

Config Center

Protocol: ? Group: ? Namespace: ? Username: ? Password: ?

Address: ? Timeout: ?

Registry Center

Protocol: ? Group: ? Username: ? Password: ?

Address: ? Timeout: ?

RPC Protocol

Protocol: ?

Consumer & Service

Timeout: ? Version: ? Retries: ? Cluster: ? Group: ? Connections: ?

Async: ? Loadbalance: ?

Interface

Get Provider List Interfaces: Methods:

Interface: ?

Method: ?

Args Attachment Args

paramType	paramValue

打开 [Dubbo Sample](#) 后我们会看到如上图的配置(可能根据版本的不同会有所变化)，如果遇到了相同的情况，将 [Version](#) 中的"1.0"删除，否则会影响我们的传参，然后再填入接口的信息即可，如下：

Dubbo Sample

Name:

Comments:

Config Center
Protocol:

Registry Center
Protocol:

RPC Protocol
Protocol:

Consumer & Service
Timeout:

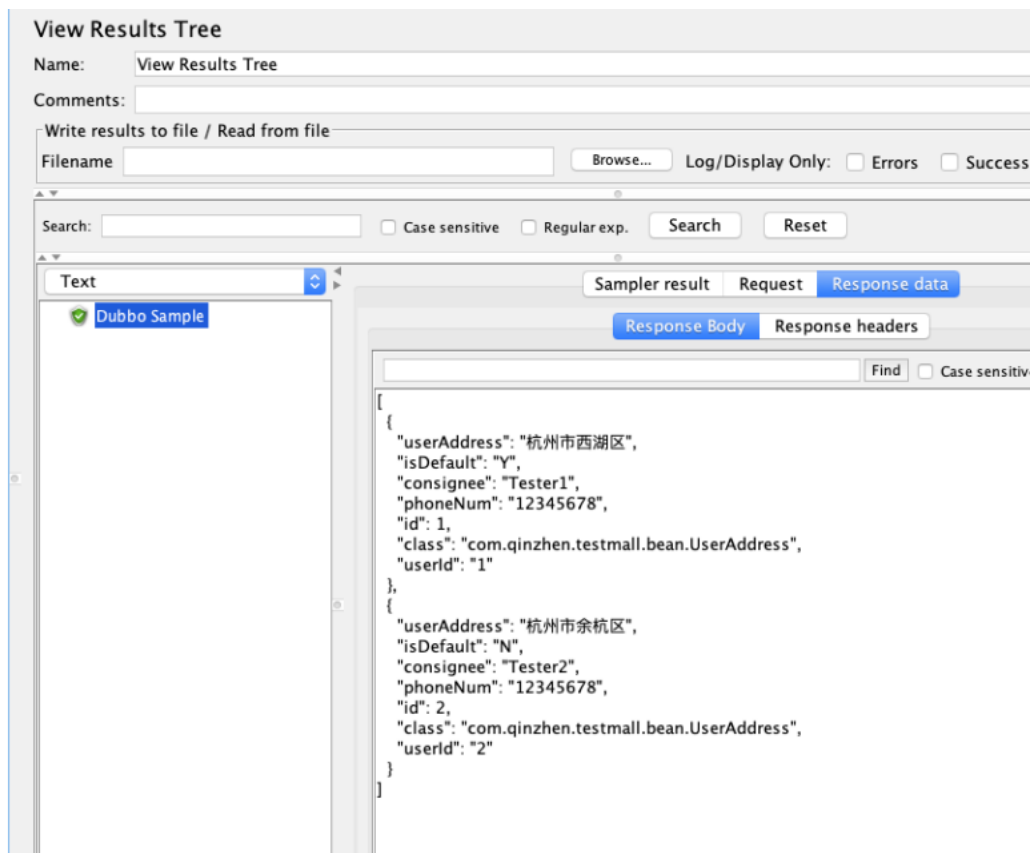
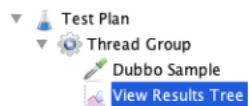
Interface
 Interfaces:
Interface:
Method:

paramType	paramValue
java.lang.String	1

- 注意：这里要注意入参的类型要按照 Java 的数据类型填完整的类型格式。

3.2 结果展示

执行测试，查看测试结果如下：



可以看到接口成功响应，结果成功返回，测试 Dubbo 接口成功。