

# 性能测试

--性能测试工具—JMeter的使用

# 内容回顾

---

## ■ 性能分析与调优

### — 前端分析与调优

- 使用抓包工具

### — 后端分析与调优

- 每个服务器（Web、APP、数据库服务器）分析是哪个慢
- 根据轻负载、重负载、超负载区性能测试指标值分析，由此分析系统性能

# 内容回顾

---

## ■ 书写性能测试报告

- 定性型报告
- 分析型报告
- 比较型报告

# 目录

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- JMeter目录结构
- JMeter体系结构分析
- JMeter运行原理
- JMeter初次使用

# JMeter简介

- Apache JMeter最早是由Apache组织开发的开发人员Stefano Mazzocchi设计编写出来用于测试Apache JServ（后来被Apache Tomcat项目替代的项目）的性能。后来Apache组织重新设计并增强了JMeter的功能，除了用于测试Web应用程序外，被扩展到了可以测试的其他应用程序

# JMeter简介

- 免费、开源、纯Java开发的性能测试工具，可以用于测试静态和动态的资源，例如静态文件、Java小程序、CGI脚本、Java对象、数据库、FTP服务器、邮件服务器和PerlScript等
- 能够对应用程序做功能/回归测试，通过创建带有断言的脚本来验证你的程序是否返回了你期望的结果，JMeter允许使用正则表达式来创建断言

# LoadRunner VS JMeter

LoadRunner	JMeter
性能测试领域的王者	后起之秀
费用高昂	完全免费
支持市面上几乎所有的应用(C、Java、JS、C#)	能够完成上述工作，有些需要单独编码实现
强大的录制功能	第三方工具与插件让JMeter变得强大
强大的结果分析功能	在第三方插件的扩展下部分分析功能也已经具备
开始的早，经验、资料多，学习成本低	开始的晚，学习资料少、学习成本高

# 目录

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- JMeter目录结构
- JMeter体系结构分析
- JMeter运行原理
- JMeter初次使用



# 为什么选择JMeter

## ■ 优点:

- 免费、开源，并且能够实现LoadRunner95%以上的功能
- 支持二次开发、能够针对企业产品做调整，更好的满足企业性能测试需求

## ■ 缺点:

- 用户友好性及集成监控不如LoadRunner

# 目录

---

- JMeter简介
- 为什么选择JMeter
- **JMeter安装**
- JMeter目录结构
- JMeter体系结构分析
- JMeter运行原理
- JMeter初次使用

# JMeter安装

1 JMeter是Java应用程序，需要有JDK环境

2 官网下载：<http://JMeter.apache.org>

3 解压即可使用

tgz: Linux系统  
zip: Windows系统

## Apache JMeter 5.1.1 (Requires Java 8+)

### Binaries

二进制文件

[apache-jmeter-5.1.1.tgz](#) sha512 pgp

[apache-jmeter-5.1.1.zip](#) sha512 pgp

### Source

源代码

[apache-jmeter-5.1.1\\_src.tgz](#) sha512 pgp

[apache-jmeter-5.1.1\\_src.zip](#) sha512 pgp

# 目录

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- **JMeter目录结构分析**
- JMeter体系结构
- JMeter运行原理
- JMeter初次使用

# JMeter 目录结构分析

- **Bin:**放置了各项配置文件（如JVM设置、日志设置）、启动文件、示例脚本等

- **JMeter.properties:**JMeter的系统配置文件，可以针对JMeter做各种配置操作，比如：远程负载机等

**remote\_hosts=127.0.0.1**

**remote\_hosts=127.0.0.1:1099,172.168.1.13:1099, 172.168.0.16:1099**

**server\_port=1099**

- **Docs:**放置了JMeter API离线帮助文档

# JMeter 目录结构分析

---

- **Extras:** JMeter辅助功能，提供与Ant、Jenkins集成的可能性，我们利用Ant与Jenkins来构建性能测试自动化构架
- **Lib:** JMeter组件以jar包形式放置在lib/ext目录下，如果要扩展JMeter组件，扩展后的jar包即放在此目录
- **printable\_docs:** JMeter的离线帮助文件放置目录

# JMeter启动

---

■ 开启：

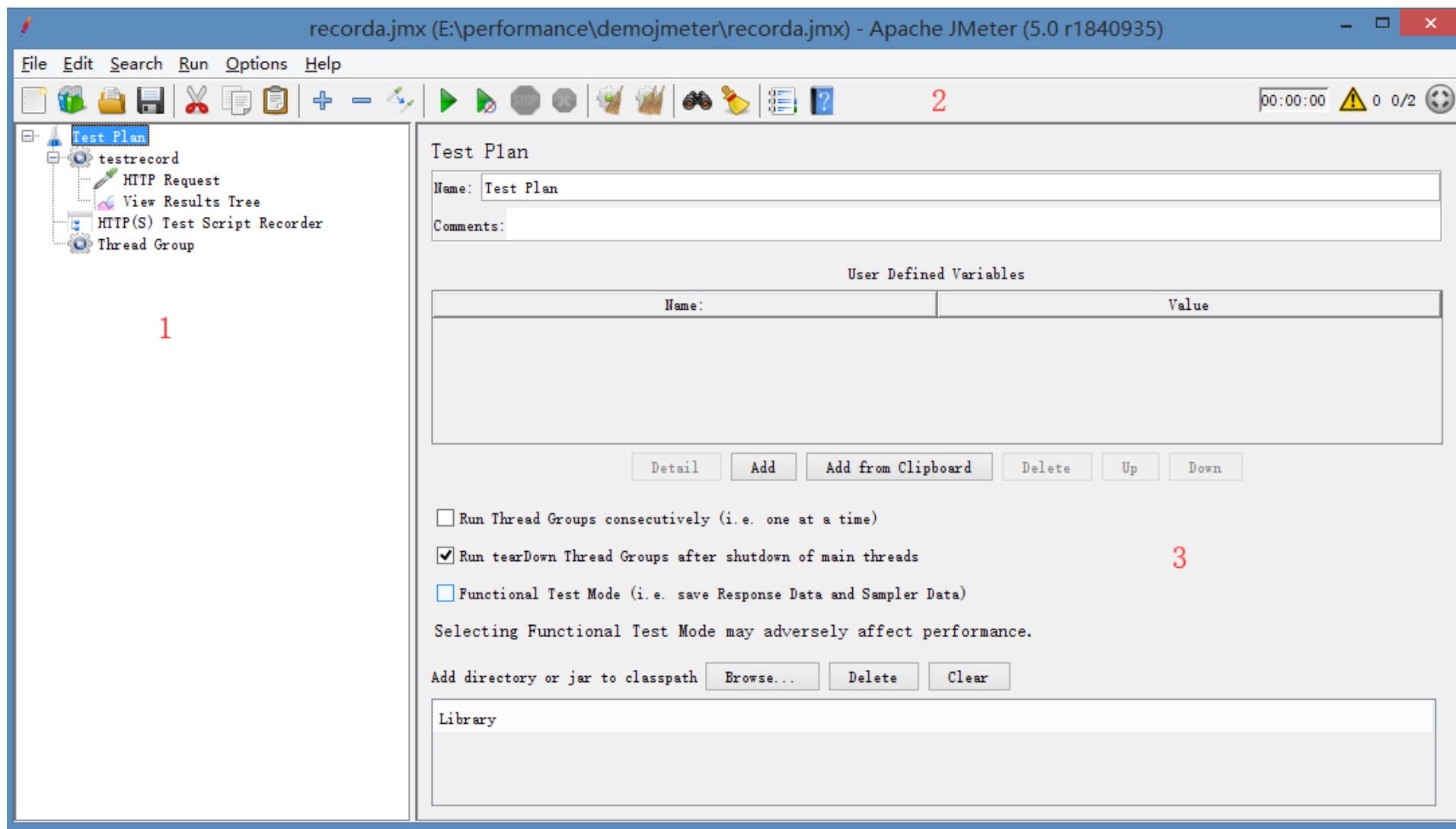


jmeter.bat

■ JMeter-server.bat

— 当前机器作为一台远程**负载机**时，启动此文件

# JMeter工作区介绍





# JMeter工作区介绍

- 区域1：目录树，存放测试设计过程中使用到的元件；执行过程默认从根节点开始顺序遍历树上的**元件**
  - 元件：比如HTTP请求就是一个元件
- 区域2：菜单栏，图标是菜单快捷方式
- 区域3：测试元件编辑区域

# 目录

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- JMeter目录结构分析
- **JMeter体系结构**
- JMeter运行原理
- JMeter初次使用

# JMeter结构体系

---

## 1、组成部分

- 1) 负载发生器：产生负载，多线程模拟用户行为
- 2) 用户运行器：脚本运行引擎，用户运行器附加在线程上，根据脚本模拟指定的用户行为
- 3) 资源生成器：生成测试过程中服务器、负载机的资源数据
- 4) 报表生成器：根据测试中获得的数据生成报表，提供可视化的数据显示方式

# JMeter使用初体验—测试计划

---

- JMeter中一个脚本即是一个测试计划，也是一个管理单元

# JMeter组件类别—线程

## ■ threads (users) 线程

### — Setup thread group:

- 一种特殊类型的线程，可用于执行预测试操作。即执行测试前进行定期线程组的执行

### — Teardown thread group:

- 一种特殊类型的线程，可用于执行测试后动作。即执行测试结束后执行定期的线程组

# JMeter组件类别—线程

- 以上两个线程组，举个例子：LoadRunner的脚本除了action里是真正的脚本核心内容，还有初始化“环境”的初始化脚本和测试完毕后对应的清除信息的脚本块，与其对应

## — Thread group:

- 通常添加使用的线程，一般一个线程组可看做一个虚拟用户组，其中每个线程为一个虚拟用户

## JMeter组件类别—取样器

- **Sampler（取样器或采样器）**：用来模拟用户操作，向服务器（被测系统）发出HTTP请求、WebService或Java请求等，可以把HTTP请求元件看成是一个没有界面的浏览器，它可以发送HTTP请求，接收HTTP响应数据

# JMeter组件类别——逻辑控制器

- **Logic Controller（逻辑控制器）**，包含两类原件：
  - 一类是控制Test Plan中Sampler节点发送请求的逻辑顺序控制器，常用的有：**If Controller、Swith Controller、Loop Controller、Random Controller**等
  - 另一类是用来组织和控制Sampler节点的，如**Transaction Controller、Throughput Controller**等



# JMeter组件类别—监听器

---

- **Listener:** 提供监测统计数据、脚本调试，也可以用来保存响应内容到文件

# JMeter组件类别—配置元件

- 配置元件：提供的就是进行各种配置的组件。可以是测试计划级别，也可以是线程组级别，或者是sampler级别的配置
  - 可以在这里选择对应的组件进行参数化操作，变量定义、登录配置元件、JDBC连接配置、HTTP相关配置等

# JMeter组件类别—定时器

- **Timer(定时器)**: 用于操作之间设置等待时间，等待时间使性能测试中常用的控制客户端QPS的手段，JMeter定义了Constant Times、Constant Throughput Times、Guass Ramdon Times等不同类型的Times
  - 每秒查询率QPS是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准，在因特网上，作为域名系统服务器的机器的性能经常用每秒查询率来衡量

# JMeter组件类别—前置处理器

## ■ 前置处理器（Pre Processors）

- 用于在实际请求发出之前对即将发出的请求进行特殊处理
- 例如：**Count**处理器可以实现自增操作，自增后生成的数据可以被将要发出的请求使用，而**HTTP URL Re—Writing Modifier**处理器则可以实现URL重写，当URL中有sessionID一类的session信息时，可以通过该处理器填充发出请求实际的sessionID

# JMeter组件类别—后置处理器

## ■ 后置处理器（Post Processors）

- 用于对Sampler发出请求后得到的服务器响应进行处理。一般用来提取响应中的特定数据（类似LoadRunner中的关联）
- 例如：Regular Expression Extractor用于提取响应数据中匹配某正则表达式的数据段，并将其填充在参数中

# JMeter组件类别—断言

- 断言（Assertions）：用于检查测试中得到的响应数据等是否符合预期，Assertions一般用来设置检查点，用以保证性能测试过程中的数据交互与预期一致

# JMeter组件类别—测试片段

---

- 测试片段（Test Fragment）：是一种特殊的线程组，在测试树上与线程组一个层级，但是它不被执行，除非它是一个模块控制器或者被控制器所引用时才会被执行

# JMeter组件类别—非测试元件

---

- 非测试元件（Non-Test Elements）：设置配置信息



# 目录

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- JMeter目录结构分析
- JMeter体系结构
- **JMeter运行原理**
- JMeter使用初体验

# JMeter运行原理

---

- JMeter以线程方式运行，通过线程组来驱动多个线程（类似LoadRunner中的虚拟用户）运行测试脚本对被测试服务器发起负载，每个负载机上都可以运行多个线程组

# 目录

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- JMeter目录结构分析
- JMeter体系结构
- JMeter运行原理
- **JMeter初次使用**

# JMeter使用初体验

---

- 创建测试计划
- 新建线程组
- 开发脚本
  - 手工书写、Badboy录制和JMeter使用代理方式录制
- 运行场景
- 查看监控

# JMeter使用初体验—测试计划

- 测试计划：是JMeter测试的起点，是存放脚本的容器，JMeter中一个脚本即是一个测试计划
- 测试计划四要素：
  - 脚本中计划只能有一个
  - 至少要有有一个线程组
  - 至少有一个采样器
  - 至少有一个监听器

# JMeter使用初体验—测试计划

- 在测试计划里面可以配置用户的一些全局变量
- 独立运行每个线程组 (**Run Thread Groups consecutively**) : 一个测试计划下面可能会包含多个线程组 (类似于Loadrunner中group的概念), 勾选此项的话, 则会顺序执行每个线程组, 而不是同时启动所有的线程组

# JMeter使用初体验——线程组

## ■ 线程组：

- 相当于有多个用户，同时去执行相同的一批次任务。每个线程之间都是隔离的，互不影响的。一个线程的执行过程中，操作的变量，不会影响其他线程的变量值

## ■ 启动线程组的方法：

- **Test Plan 右键——Adds——Threads——Thread Group**

# JMeter使用初体验—开发脚本

## ■ 添加HTTP请求

- 添加完成线程组后，在线程组上右键菜单（添加--->Sampler--->HTTP请求）选择HTTP请求。对于JMeter来说，取样器（Sampler）是与服务器进行交互的单元。一个取样器通常进行三部分的工作：
  - 向服务器发送请求
  - 记录服务器的响应数据
  - 记录响应时间信息





# HTTP Request 参数解释

---

- **Name**: 该请求的名称
- **protocol**: 协议名称
- **ServerName or IP**: 被请求的服务器名称或IP
- **Port Number**: 端口号
- **方法**: HTTP请求的方法;
- **Path**: 路径

# HTTP Request 参数解释

---

- **Use keep Alive:** JMeter 和目标服务器之间使用 Keep-Alive方式进行HTTP通信（默认选中）
- **Use multipart/form-data for HTTP POST :** 当发送HTTP POST 请求时，使用

# HTTP Request 参数解释

---

- **Content encoding:** 内容的编码方式 (**Content-Type=application/json;charset=utf-8**)
- **路径:** 目标的**URL**路径 (不包括服务器地址和端口)
- **自动重定向:** 如果选中该项, 发出的**HTTP**请求得到响应是**301/302**时, **JMeter**会重定向到新的界面

# JMeter使用初体验——设置线程组

- **线程数(Number of Threads)**: 虚拟用户数。一个虚拟用户占用一个线程。设置多少虚拟用户数在这里也就是设置多少个线程数
- **准备时长 (Ramp-Up Period (in seconds))**: 设置的虚拟用户数需要多长时间全部启动。如果线程数为20, 准备时长为10, 那么需要10秒钟启动20个线程。也就是每秒钟启动2个线程
- **循环次数 (Loop Count)**: 每个线程发送请求的次数。如果线程数为20, 循环次数为100, 那么每个线程发送100次请求。总请求数为 $20 \times 100 = 2000$ 。如果勾选了“永远”, 那么所有线程会一直发送请求, 直到选择停止运行脚本

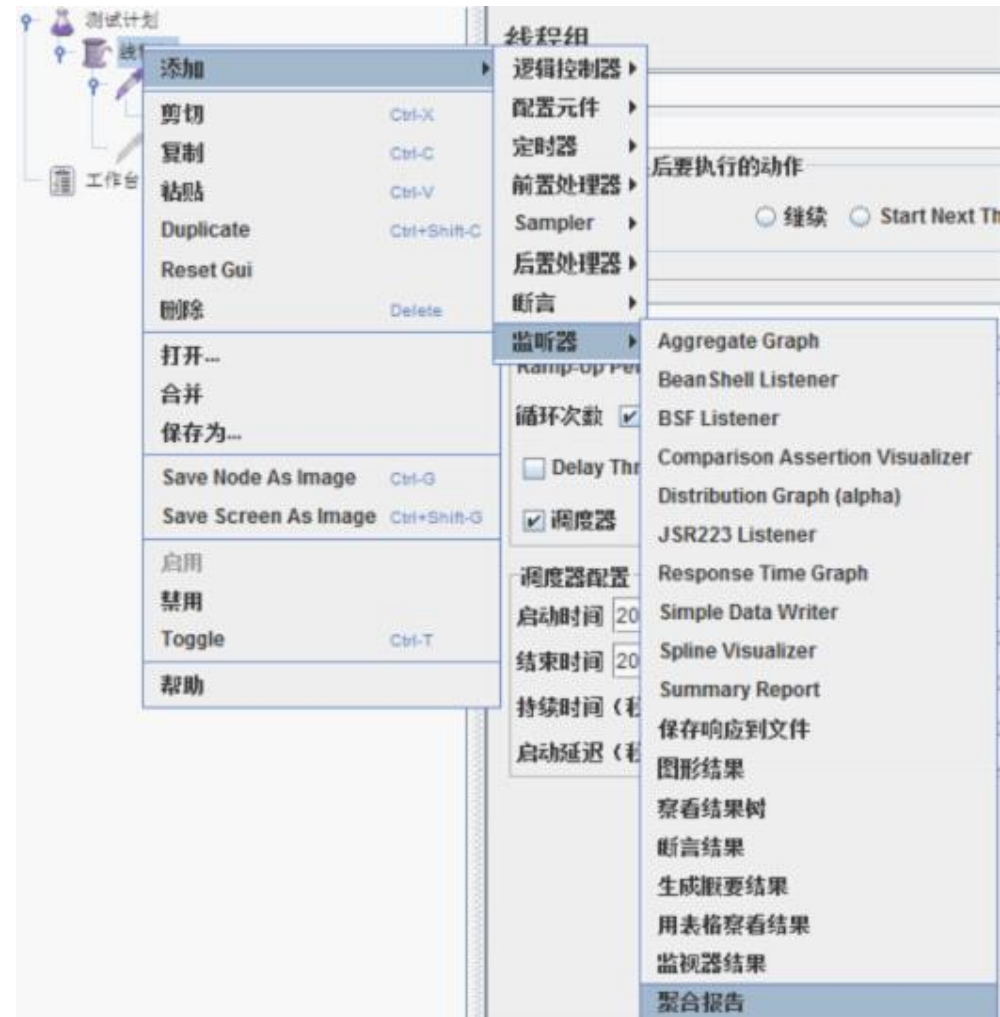
# JMeter使用初体验——设置线程组

## ■ Scheduler: 调度器

- **Duration (seconds)** : 持续时间, 测试计划持续多长时间
- **Startup delay(seconds)**:启动延时。点击启动按钮后, 仅初始化场景, 不运行线程, 等待延时时间到才运行

# JMeter使用初体验—添加监听器

- JMeter 中使用监听器元件收集取样器记录的数据并以可视化的方式来呈现。JMeter有各种不同的监听器类型，这里添加聚合报告来查看结果



# 聚合报告结果分析

Term	Definition
Label	每个 JMeter 的 element(例如 HTTP Request) 都有一个 Name 属性, 这里显示的就是 Name 属性的值
#Samples	表示你这次测试中一共发出了多少个请求, 如果模拟 10 个用户, 每个用户迭代 10 次, 那么这里显示 100
Average	平均响应时间——默认情况下是单个 Request 的平均响应时间, 当使用了 Transaction Controller 时, 也可以以 Transaction 为单位显示平均响应时间。
Median	中位数, 也就是 50% 用户的响应时间。
90%Line	90% 用户的响应时间。
Min	最小响应时间。
Max	最大响应时间。
Error%	本次测试中出现错误的请求的数量/请求的总数。
Throughput	吞吐量——默认情况下表示每秒完成的请求数 (Request per Second), 当使用了 Transaction Controller 时, 也可以表示类似 LoadRunner 的 Transaction per Second 数
KB/sec	每秒从服务器端接收到的数据量, 相当于 LoadRunner 中的 Throughput/Sec

# JMeter使用初体验—添加监听器

- **View Results Tree:** 如果我们的请求成功发送给服务器，那么结果树里面的模拟请求会显示为绿色，可以通过取样器结果里面的响应状态码信息来判断

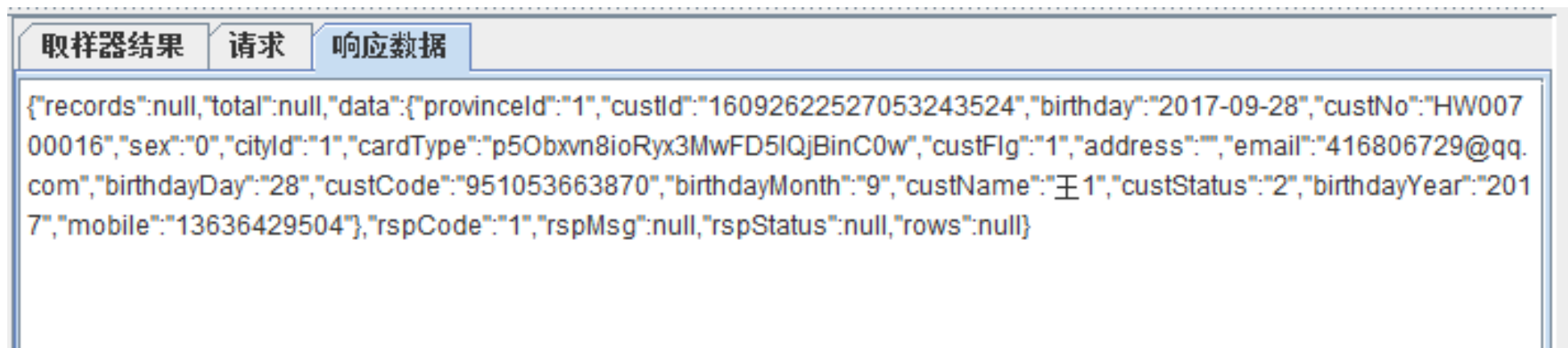


- 里面有我们发送的请求的方法、协议、地址以及实体主体数据，以及数据类型，大小，发送时间，客户端版本等信息



# JMeter使用初体验—添加监听器

- 响应数据：里面包含服务器返回给我们的响应数据实体



## 录制方式产生脚本—使用Badboy录制

- Badboy是用C++开发的，被用于测试和开发复杂的动态应用。它提供了强大的屏幕录制和回放功能，同时也提供了丰富的图形结果分析功能
- 下载Badboy:<http://www.badboy.com.au/>
- 使用Badboy录制脚本，然后将录制的脚本导出为JMeter格式的脚本，最后将该脚本导入到JMeter，借助于JMeter强大的测试功能模拟大量的虚拟用户，进行复杂的性能测试
- 在Badboy中，step就类似于Loadrunner中事务的概念，我们可以通过添加step的方式来定义事务

# 录制的原理

- LoadRunner/JMeter录制是针对网络通讯协议层面的，它只关心客户端与服务器端的通讯包
- LoadRunner/JMeter的并发测试实际上就是并发客户端与服务器端的通讯过程
- 压力是通过多进程/多线程方式实现的，目前流行OS都是支持线程。每个进程/线程向服务器后台发送1个请求脚本数据包

# 录制的原理

- LoadRunner/JMeter录制脚本采用proxy的方式，客户端请求的数据会先到proxy服务器(录制器)，录制器将数据转发给服务器端；服务器端的响应数据先发给录制器，然后由录制器响应给客户端

# 内容总结

---

- JMeter简介
- 为什么选择JMeter
- JMeter安装
- JMeter目录结构分析
- JMeter体系结构
- JMeter运行原理
- JMeter初次使用



# Question

---