# Jmeter 介绍

## 1. Jmeter是什么？

### 1.1 概念

Jmeter：是Apche公司使用Java平台开发的一款测试工具。

作用：

1. 接口测试

2. 性能测试

3. 压力测试

。。。。。。

## 2. 了解：配置JMeter运行环境（JDK、JRE、JVM）

### 2.1 JDK

JDK概念：java开发工具包，程序员使用。包含JRE和JVM。

下载方式：

1. 官网下载地址：<http://www.oracle.com/>

2. 百度搜索“JDK”关键字

我们要知道，如果使用JMeter，必须要安装JDK或者JRE。

**Jmeter**工具安装

目标

**-** 了解**Jmeter**安装目录结构

**2. Jmeter**下载与安装

**2.1** 官网下载地址**:**

http://jmeter.apache.org/download\_jmeter.cgi

下载示意图：

注意：

下载后，解压文件到任意目录，避免在一个有空格或者有汉字的路径安装Jmeter，这将导致问题。

需要配置环境变量

**2.2** 启动**JMeter**的两种方式：

进入bin目录

1. 双击 ApacheJMeter.jar文件;

2. 双击 Jmeter.bat文件;

i. 出现Jmeter不是内部或外部命令在环境变量PATH中添加Jmeter路径bin目录，

比如（E:\lisheng-workspace\software\apache-jmeter-4.0\bin）

## 3 Jmeter常用目录文件介绍

**3.1 Bin**目录

存放可执行文件和配置文件

Jmeter.bat：windows系统中JMeter的启动文件

ApacheJMeter.jar Java环境下的JMeter启动文件

Jmeter.log：日志文件

Jmeter.sh：linux系统中JMeter的启动文件

Jmeter.properties：系统配置文件

Jmeter-server.bat：windows分布式测试要用到的服务器配置

Jmeter-serve：linux分布式测试要用到的服务器配置

**3.2 docs**目录

docs：是JMeter的java Doc，可打开api\index.html页面来查看;

**3.3 printable\_docs**目录

printable\_docs的usermanual子目录下的内容是JMeter的用户手册文档，其中usermanual

下**component\_reference.html**是最常用到的核心元件帮助文档。

## 4 Jmeter 工具功能界面布局

目标

了解Jmeter功能界面布局

熟悉测试计划面板

**1.** 主界面布局

JMeter的主界面布局分为标题栏、菜单栏、工具栏、树形标签栏和内容栏

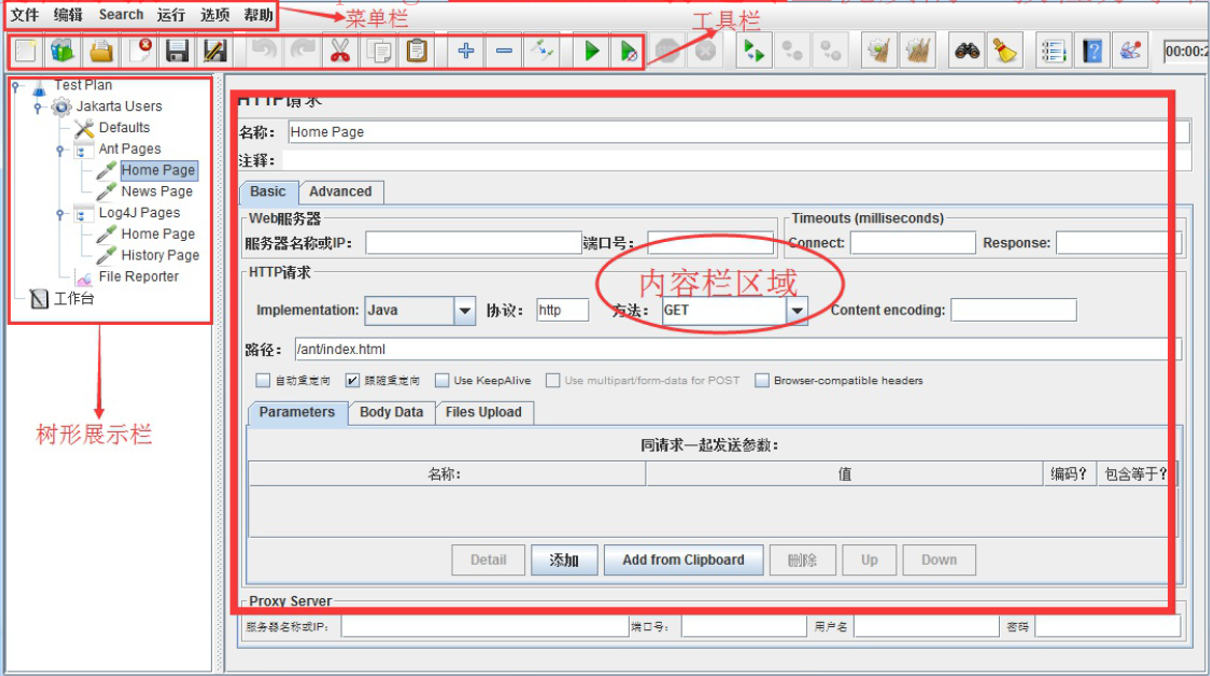
标题栏：主要显示计划信息及JMeter版本。

菜单栏：全部的功能的都包含在菜单栏中。

工具栏：工具栏中的按钮在菜单栏都可以找到，工具栏就相当于菜单栏常用功能的快捷按钮

树形标签栏：树形标签栏通常用来显示测试用例（计划）相关的标签。

内容栏：配合树形标签栏显示，树形标签中点击哪个标签，内容栏中就显示相应的内容和操作。



## 5.使用JMeter进行接口测试 重要

### 使用JMeter的解决方案

1. 添加【\*\*测试计划\*\*】

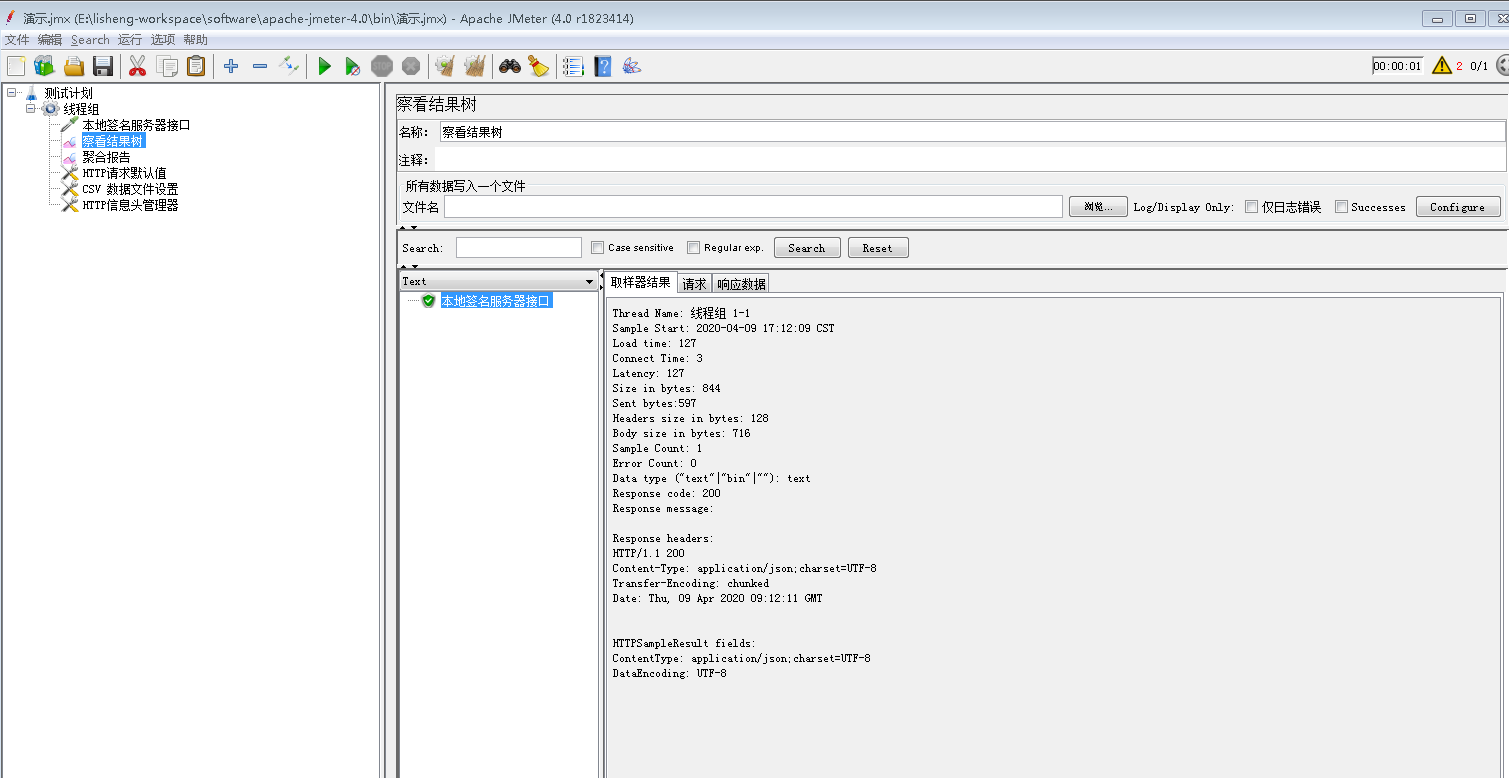
2. 基于添加的测试计划添加【\*\*线程组\*\*】，循环次数设置为100次

3. 在【sampler】中基于线程组添加\*\*HTTP请求\*\*

4. 在【监听器】基于线程组添加【察看结果树】

5. 在监听器基于线程组添加【聚合报告】

**Test Plan(**测试计划**)**



作用：

1. 本次测试所需要的【组件】都是基于测试计划添加；

2. 本次测试所有组件的设置与执行都基于测试计划；

组件：完成指定功能代码段的封装；

选项**(**在这里我们只介绍我们会使用到的选项**)**

独立运行每个线程组：

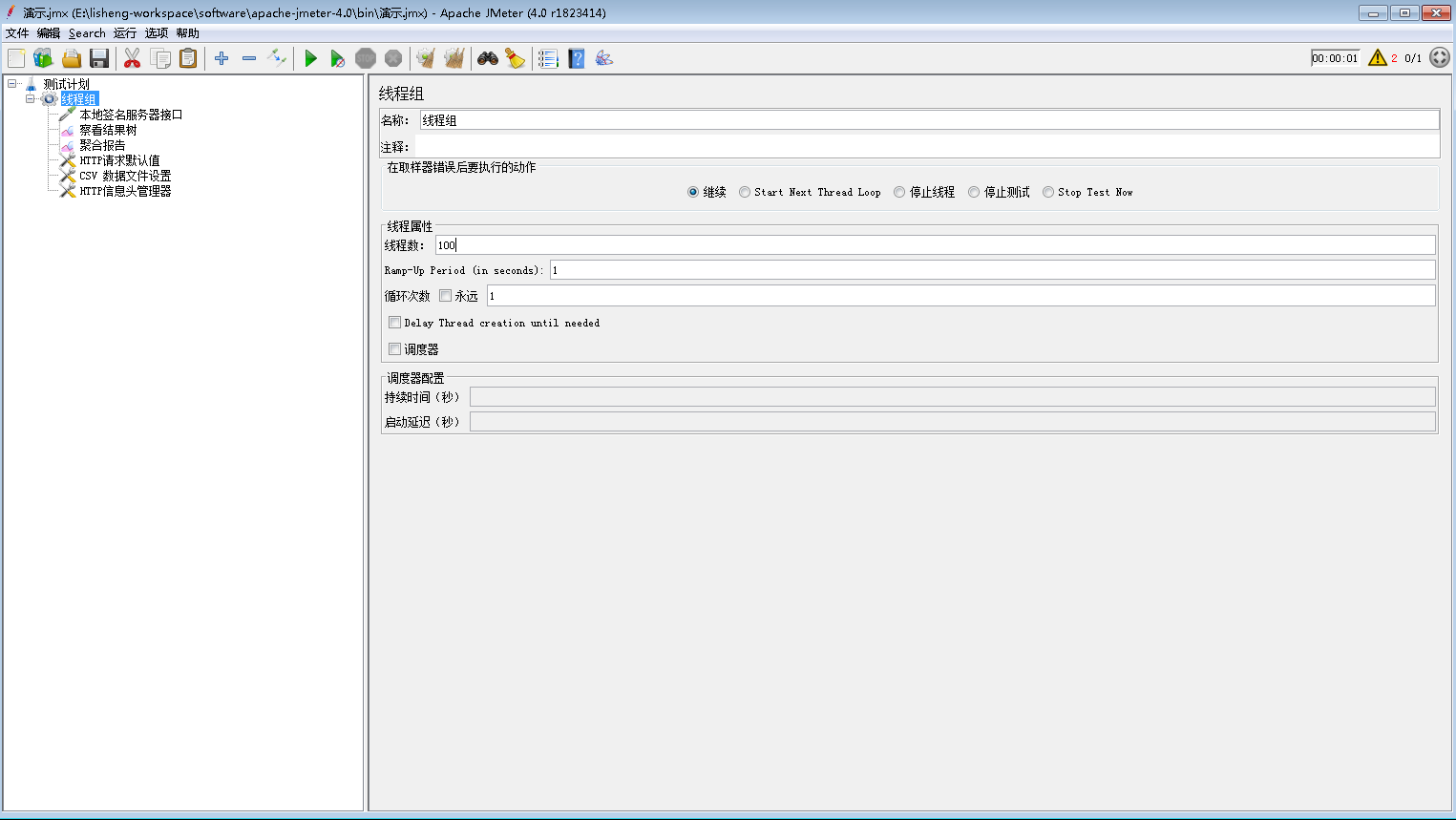
进程：是每个正在运行的应用程序。

线程：按照进程的指令去执行指定的代码。

线程组（多线程）：多个线程的组合。

线程组（多线程）的执行顺序是并行的。

### Threads(User)线程组 【重点】



1) thread group(线程组)

2) setup thread group【特殊线程组】

3) teardown thread group【特殊线程组】

**4.1 thread group(**线程组**)**

作用**:**

1. 添加测试中使用的大多数组件

线程属性

31

- 线程数：虚拟用户数

- Ramp-Up Period(in serconds)：启动虚拟全部用户数所需要的时间

- 循环次数 ：指定次数或勾线永远

- 调度器：勾选后，调度器配置才能使用；

调度器配置

- 持续时间（秒）：设置脚本压测持续时间

- 启动延迟（秒）：启动延迟时间

### 组件详解

1. HTTP请求

2. 察看结果树

**4.3 - HTTP**请求

作用**:**

1. 模拟前端或第三方软件向服务器发送请求;

2. 设置请求时的方法和参数数据;

参数详解：

1. 名称：本属性用于标识一个取样器，建议使用一个有意义的名称。

2. 服务器名称或IP ：HTTP请求发送的目标服务器名称或IP地址。

3. 端口号：目标服务器的端口号，默认值为80 。

4. 协议：向目标服务器发送HTTP请求时的协议,可以是http或者是https ,默认值为http 。

5. 方法：发送HTTP请求的方法，可用方法包括GET、POST、PUT、DELETE。

6. Content encoding ：内容的编码方式，默认值为iso8859；一般设置【UTF-8】

7. 路径：目标URL路径（不包括服务器地址和端口）

8. 同请求一起发送参数:请求时需要传递参数，如：学院资源list查询

http://127.0.0.1:8000/api/departments/?$dep\_id\_list=T01,T02,T03

参数名称：【$dep\_id\_list】

参数值：T01,T02,T03

**Body Data**选项作用：

1.新增或更新时需要传递JSON报文；如学院新增是的JSON报文填写位置：

{

"data": [

{

"dep\_id": "T01",

"dep\_name": "Test学院",

"master\_name": "Test-Master",

"slogan": "Here is Slogan"

}

]

}

2. 【注意】：新增和更新时传入报文也需要设置Content-Type:application/json

告诉服务器我传的数据格式为JSON格式；

设置地点：配置元件-->HTTP信息头管理器

**HTTP**请求总结：

1. 接口完整请求地址

2. JSON报文存放地址

3. 设置默认请求数据格式

**4.3 -** 察看结果树

作用：

1.查看请求服务器时的请求信息;

2.查看服务器响应数据;

3.记录信息到指定文件;

说明：

1. 文件名：存放服务器响应后的状态信息； 如：e:\查询所有response.txt

2. 取样结果：服务器响应的信息头信息；比如：响应代码，响应数据大小

3. 请求：查看向服务器请求时的信息；比如：请求地址、方法、数据等

4. 响应数据：查看服务器响应的数据；比如：获取资源时，返回的JSON数据

察看结果树总结：

1. 查看请求

2. 查看响应

3. 存储请求状态信息

**4.4** 线程组总结：

setup thread group：一种特殊线程组，测试计划运行之前首先执行，一般做初始化操作

teardown thread group：一种特殊线程组，测试计划运行结束时运行，一般做收尾工作

thread group(线程组) ：线程组，我们测试计划中场景创建和实现都是基于此线程组

元件

概念：相同类似功能组件的集合称之为元件

1. 逻辑控制器

2. 配置元件

3. 定时器

4. 前置处理器

5. Sampler

6. 后置处理器

7. 断言

8. 监听器

元件结论：

只学重要的、常用的

**Jmeter** 各元件中需要掌握元件

整理出各大元件常用的重点组件

**1.** 配置元件（**config Element**）

1) CSV Data Set Config

2) HTTP请求默认值

3) HTTP信息头管理器

**6.** 断言（**Assertions**）

1) 响应断言

**6.** 监听器（**Listener**）

1) 察看结果树

2) 聚合报告

3) 断言结果

总结：

正常来说，应该开始按照顺序一个组件一个组件的进行讲解。

问题：每个组件都不能独立执行。都需要多个组件进行配合，才能够解决实际问题。

解决方案：按照JMeter主要解决的问题点来讲解组件。

### Jmeter 参数化

目标

学习掌握Jmeter中常用参数化方式

**1.** 为什么要参数化？

**1.1** 需求新增**10**条数据

学院-新增

1) 请求方法：POST

2) 请求地址：http://127.0.0.1:8000/api/departments/

3) 请求JOSN报文：

4) 调用传入的json串如下（可新增多条，之间用,隔开）：

{

"data": [

{

"dep\_id": "T01",

"dep\_name": "Test学院",

"master\_name": "Test-Master",

"slogan": "Here is Slogan"

}

]

}

1. 键所对应的值都是写死的，只能手动更改

2. 无法解决新增大数量（1000条）的问题

**2.** 什么是参数化？

概念：根据需求动态获取数据并进行赋值的过程

CSV Data Set Config

用户参数

用户定义的变量

函数

**1. CSV Data Set Config**（数据集配置）

概念：一种从外部读取数据功能的组件

**1.1** 实施方案分析

1. 基于测试计划->线程组

2. 基于线程组->配置元件->CSV Data Set Config

3. 基于线程组->Sampler->HTTP请求

4. 基于测试计划->HTTP信息头管理器

5. 基于测试计划->监听器->察看结果树

**1.2** 组件要点分析

1. 线程组:循环次数10

2. CSV Data Set Config 读取变量配置

3. HTTP请求：Body Data填写(JSON报文) 方法(POST)

4. 参数化引用格式：${参数名} 如：${ keyId }

5. HTTP信息头管理器：Content-Type:application/json;charset=utf-8

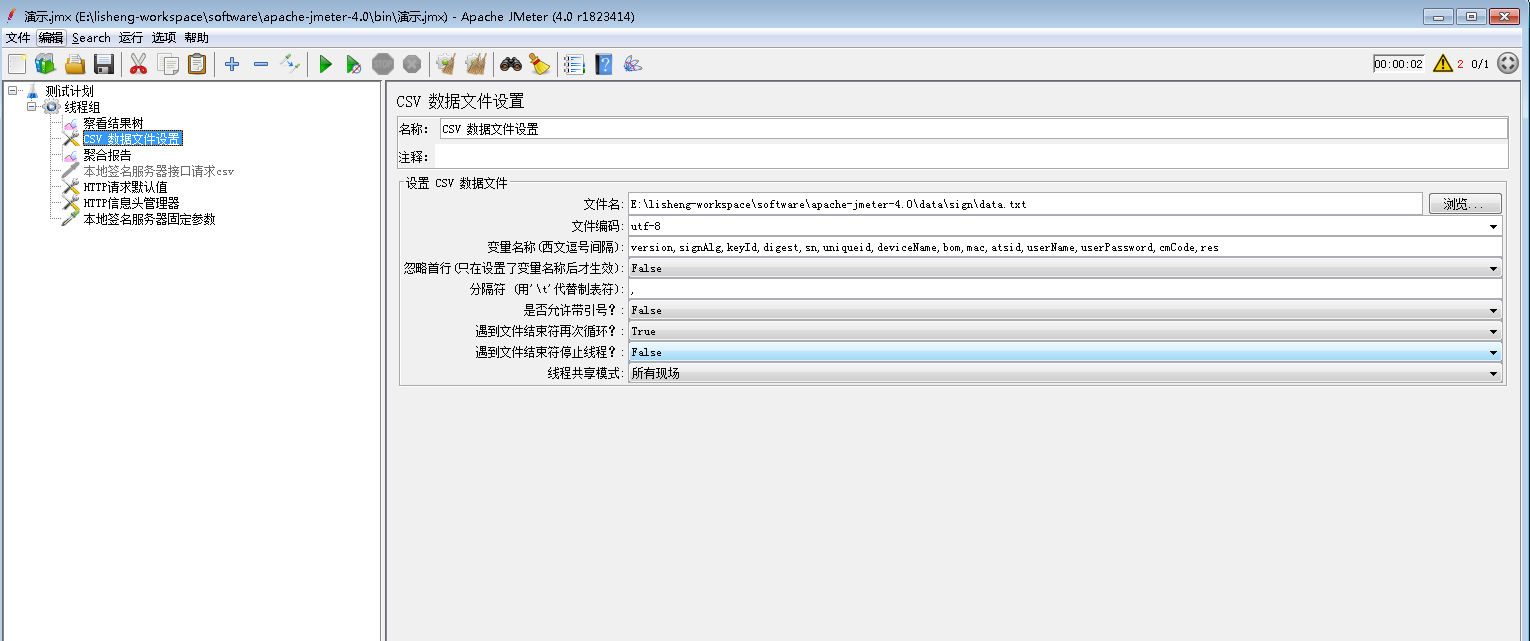
**1.3 CSV Data Set Config** 参数配置图

1. Filename:文件路径+文件名+后缀名 如：d:/a.txt;

2. File Encoding:文件编译字符编码，一般设置utf-8;

3. Vaiable Names:读取参数后保存的变量名称;

4. Delimiter:如文件中使用的是逗号分隔，则填写逗号；如使用的是TAB，则填写\t;



**1.4 HTTP**信息头管理器 参数配置图

作用

告诉服务器请求的数据格式

1. Content-Type:指定请求信息格式-类型名称

2. application/json:指定请求信息为-JSON格式

3. charset=utf-8:字符编码

**1.5 CSV Data Set Config-**总结：

1. 参数化概念

2. CSV Data Set Config 配置参数设置

3. 参数化引用格式

4. HTTP请求

5. HTTP信息头管理器作用与设置

