**接口测试:接口**

# 测试步骤

#### 1.拿到API文档,梳理文档.添加数据测试接口是否正确

#### 2.准备数据（txt）去测试接口,测试数据库约束(功能测试)

#### 3.接口自动化测试(此时程序可能已经上线，目的是回归测试的时候保证其他接口程序没有被改动)

#### 4.接口性能压力测试（高并发{秒杀}，高频率{QPS}），在服务器上进行的测试

### 5.测试报告

# 

# 概念:

#### 接口:数据交互的入口与出口,还是一套规范一套标准

###### 硬件:USB接口,网线接口,耳机接口,视频线接口

# 实现:

##### 怎么编写API文档

#### API:Application programming interface 应用程序编程接口

API文档内容:

描述程序中各个模块的各个功能点的具体实现流程，开发要按照此流程编写程序实现

## 接口测试：

**概念：**

**接口测试**：测试后端是否符合接口规范

**作用**：

**易于BUG定位**

**实现：**

**实现流程：**模拟前端直接访问服务器资源,提交数据,然后查看响应是否符合预期.

为什么模拟前端?可以直接提交数据,保证数据一致性,如果使用前端,前端处理数据也有可能产生BUG.

**解决：跨过前端(就是保证前端不出BUG的模拟过程)直接访问后端**

**要素:**访问服务器资源接口**(定位接口**)==URL

**要素**2:设置**提交的数据**

**要素**3:查看响应是否符合预期(**状态码+响应体**)

**接口测试分类**(记住):

**WEB接口测试(BS):**

-**-服务器接口测试**:自实现接口

--**外部(第三方)接口测试**:天气预报,支付接口

**模块接口测试(CS)**

**接口测试:环境搭建**

**流程:**

**1.安装Python环境(3.5)**

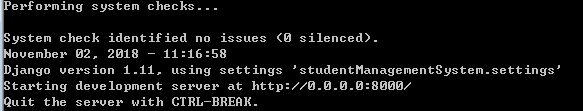
**2.导入项目,安装项目依赖**

联网安装（requirements.txt文件的目录下）

pip install -r requirements.txt -i https://pypi.douban.com/simple

在run\_server.py这个文件夹的路径中输入cmd

输入python run\_server.py



显示这个就是成功

**这个命令窗口不要关掉!!!!**

验证：

浏览器输入 http://localhost:8000/api/

显示 {"departments":"http://localhost:8000/api/departments/"}

**3.启动项目并验证安装成功**

**Method:**

**GET:查询**

**Post:新增**

**Put:修改**

**Delete:删除**

# Jmetet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（数据集配置）

**概念 ：一种从外部读取数据功能的组件**

**CSV：一种常见的数据存储方式**

# 信息头

**Content-Type application/json;charset=utf-8**

在传递键值对，字典等需要加

**参数化\_用户变量**

**使用步骤：**

1. **配置元件——用户自定义变量**
2. **点击添加:设置属性名和属性值**

**例如:lujing为属性名,** **/api/departments/为属性值**

1. **调用刚刚设置的变量:${变量名}**

**函数**

**定义:jmter函数保存在函数命令中**

**使用步骤:点击菜单倒数的第二个按钮,弹出一个对话框:选择想要使用的函数使用**

**1,计数器函数:counter**

1. **第一个参数设置为true(代表每个用户单独计数)**

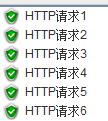
**False(代表每个用户用一个计数)**

**(2)点击生成按钮,可以复制代码到**

**(3)例如两个用户3次循环用TRUE的结果是**



**用false的结果是:**



# Jmeter直连数据库

定义:不通过接口,直接通过书写Sql语句来连接数据库的方式

使用步骤:

1. 点击测试用例,右侧最下方,点击浏览按钮,添加.jar包(安装驱动)
2. 添加配置元件—JBDC cc数据库的连接池,设置要连接哪一个数据库的;
3. 变量名: 随意起名称 例如dbb
4. Database URL :jdbc:sqlite:具体的数据库文件路径
   1. 注意:路径中的单杠换成双杠(防止java的转移)
5. JBDC DC : org.sqlite.JDBC jdbc的一个运行命令
6. 添加sampler---JDBC request 请求:
7. 变量名必须设置刚刚的那个名字,对应
8. 直接在下面输入框写sql查询语句
9. 在结果树中运行,结果树查看结果
10. 结果在处理(拿到第一个请求数据后,把某条数据当做已知条件,另外一个请求中使用)

步骤:

1. JDBC request 最下面找到变量,设置一个自己(目的是吧结果放在这个以变量名为前缀的变量中)
2. 线程组—sampler—debug sampler
3. 运行程序,查看结果树,debug选项,从中可以得到所有数据对应的变量
4. 在想调用的地方,(自己添加的另外一个请求)直接写${变量名}

注:发送请求的同时传递参数数据方法

点击添加按钮:设置属性名和属性值即可

**结果再处理(关联)**

**定义:有两个请求,第一个请求使用xpath获取想要定位的数据,放到第二个请求中执行**

**步骤:**

1. **新建2个请求**
2. **第一个请求—添加==后置处理器—xpath**
3. **点击use tidy按钮(定位标签)**
4. **引用名称:自己起名字.里面存储的是xpath定位的值**
5. **Xpath query:xpath 定位语法 //title**

**3.在第二个请求中直接传递参数,把上一步中的变量获取即可${变量名}**

**结果再处理（正则法）**

**正则法和xpath区别：**

**Xpth是获取标签，元素，标记的〈title〉如果数据中没有剑括号，就需要使用正则来匹配（定位）**

**正则法：普通的数据，使用正则方法匹配**

**步骤：**

1. **在一个请求身上添加后置处理器—正则**
   1. **引用名称：就是变量**
   2. **正则表达式：根据想要获取的时间书写正则表达式语法（复制想要的代码即可）**

**，把想要的具体获取的位置写上固定的语法：（。\*？）**

**. 匹配任意字符**

**\*匹配任意格式**

**? 结束**

**() 划定范围**

**C)模板:**

**2.匹配数字:从1开始,代表获取第几个,0代码随机获取一个**

1. **在第二个请求中直接调用刚刚设置的变量即可.**

# 断言

定义：程序帮助我们进行判断操作

使用步骤:

1. 添加http请求
2. 点击HTTP请求—添加—duany
   1. 要测试的字段
   2. Document (text):响应结果是文档,可以使用他进行断言,但是兼容性很差,我们一般不用
   3. <URL:地址>
   4. 响应代码
   5. 响应信息:response message:OK
   6. Response headers:响应头
   7. Lgnore 如果响应位4xx,5xx的服务器报错,就要勾选此项匹配规则:
      1. 分为两组,前两个是正则语法的匹配,我们不使用,后两个是我们常用的
      2. Equals:判断相等
      3. Substring:判断包含

点击添加按钮写具体要判断的值例如:判断200代码响应信息

或者200 OK包含信息

# 长度断言

1. 长度:

http请求点击:---断言—size

设置大小的输入框,里面要写判断的字节大小(注意:要判断响应码200,应该写3才可以

1. 持续时间:

http:点击—断言—断言持续时间:

持续时间:以毫秒为单位,就是判断地址的打开速度;

注意:1000毫秒=1秒

# 集合点(模拟高并发场景)

场景：秒杀

使用步骤：

1、http请求点击一定时器--sync。

Group by：一组能放多少人．这个数字等于自己的设置的线程数即可

毫秒数,默认0即可

# 聚合报告

使用场景：当结果很多,需要查看一些跟性能有关的数据，我们要使用聚合报告来查看

使用步骤：

1、监听器一聚合报告：

samplers：请求的教量

Average：平均响应时间

Min：最小响应时间

max：最大响应时ral

error：错误率（当高并发的时候，非常容易出现错）

**thoughtput: 吞吐量（每秒能执行多少次请求〕普通电脑的200次：好一点的会上千**

，最好的可以几万

received：接受数据的速度（网络情况）

send：发送教据的速度（网络情况）

# 常用的函数

1. Random: 生成随机数一参数1表开始的值．参数2代表结束的值（都是闭区间）

Time (1）如果不设置参数，会生成1970.1,1到现在的时间，以毫秒计，程序员为［防

止缓存，而写的一串数字：}

1. (2）设置第一个参数：  
   yyy 代表年

MM 代表月

dd 代表日

hh或HH 代表时

mm 代表分钟

ss 代表秒

### Jmeter:分布式

1. 是什么?

多台设备协作完成测试任务

分布式架构:

必须要有控制机和执行机

工作场景:

1.当一台机器无法进行测试时(性能不够),我们需要跟上级领导申请测试机协助我们完成任务

2.申请成功后,会给我们执行机的地址

3.把地址放在控制机的bin目录下的jmeter.properties文件中的remote\_hosts的值里

格式为 remote\_hosts=127.0.0.1:xxxx 目的是为了让控制机和执行机建立联系

4.远程打开执行机中的jmeter文件中的bin目录jmeter\_server.bat目的是让这台机器中的jmeter启动待命

5.在控制机上正常制作脚本,在执行的时候,点击运行按钮—远程全部启动

注意:此时执行的命令的就是执行机,而控制机就是负责发布指令和收集结果的

100个请求—设置的时候设置50个即可,因为要给2个执行机去执行

B、为什么?

安全: 数据存储

高效、功能强大:计算能力强大

灵活: 可插拔设计

# QPS（吞吐量。，每秒访问次数）

使用场景：每秒以20次的频率去访问，就必须要使用jmeter中的“常量定时器”，来

模拟这个频率的设置

1. 确定访问多少次：运行的总秒 \* 频率数 10 \* 20 =200 设置线程组的循环次数
2. 点击http请求 -- 添加—定时器—constant常量定时器：

Minute： 频率 \*60 20\*60=1200

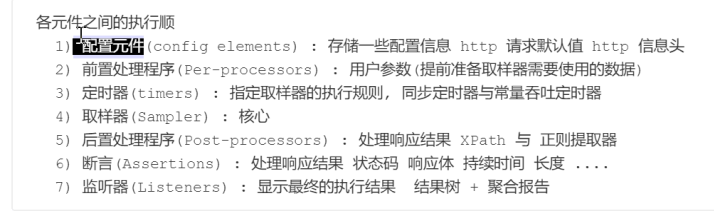
3.相看结果推荐使用聚合报告来查看，因为可以看出一些 错误率，平均时间

注意：qps只就是每秒访问的次数，就是频率，吞吐量；最后查看结果中的吞吐量自己设置的频率是一定的误差。

# 作用域范围总结

1. 逻辑控制器只能控制子集

# Jmeter组件执行顺序



**逻辑控制器比较特殊，只对子集起作用。**

# 图形化监听插件（组件）

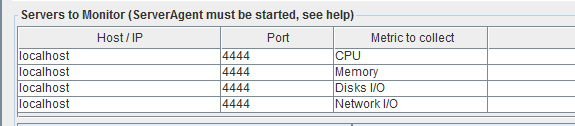
定义：工作中我们需要给服务器添加一个监听的程序，来随时查看服务器的各种运行状

态，保证服务器的正常运行；

使用步骤：

1.需要在服务器端安装窃听程序serveragent，点击文件夹中的stertagent.bat

会出现cmd

1. 自己测试机也需要安装一个
2. 必须从新启动jmter软件jp@gc - PerfMon Metrics Collector
3. 正常写的http（www.baidu.com）请求，为了方便查看服务器运行运行状态，把循环次数设置为无限
4. 添加监听器

点击add row添加想要监听的模块

1. cpu
2. memory 内存
3. Disks i/o
4. Network i/o 网络的读取写入
5. 点击运行等待图形化写入

# FTP的上传下载

常用端口号 http=80 https=443 ftp=21 ftps=22

1. ftp就是文件传输协议：

安装好ftp的服务器软件，Quick Easy FTP Server V4.0.0.exe

必须设置好根目录

1. 点击左上角绿色按钮，即可ftp服务

# 测试片段（解耦和）（不常用）

定义：测试片段就跟程序中的函数理念一样的，函数封装好之后，不会执行，必须调用才会执行，所以测试片段也必须调用才可以执行

方法1：同一文件调用

1. 测试计划—testfragment—点击选中—添加http请求
2. 自己在新建一个线程组，点击新建逻辑控制器，到此相当于实现了函数的封装

c．点击模块控制器右侧的下拉菜单中，想要调用那个testfragment到此相当于实现了函数的调用

方法2：不同文件调用

1. 写好带有testfragment的jmx文件留存
2. 在新的jmx文件中，新建线程组-逻辑控制器—inclunde
3. 浏览找到选择即可

# 接口测试阶段的功能测试

接口测试和前端页面都需要进行功能测试;因为前台的验证非常容易使用软件绕过,所以后台也必须进行功能测试.

步骤:

1. 准备txt格式的数据:就是使用txt格式写的测试用例,因为我们使用jmeter软件的csv直接可以读取使用的是(utf-8格式,数据中可以再添加一条注释的内容,方便进行结果对照)
2. Jmeter脚本设置，使用csv引入刚刚的txt文件，别忘了设置循环次数与txt文件中的条数相符，为了方便查看，可以使用计数器，需要加请求头
3. 点击运行之前，需要先清空数据库中的数据，方便插入数据后进行对照，
4. 点击运行查看结果，对照预期结果与实际结果，如果不相符就提交bug

# 

# 自动化脚本

当功能测试结束后,数据已经可以保证没有问题了,此时到了最后项目上线,程序肯定还会进行修改,我们必须进行回归测试,而进行回归测试的时候,无需再次测试具体的数据详细功能测试 ,只需测试,增删改查的基本功能模块正常即可.

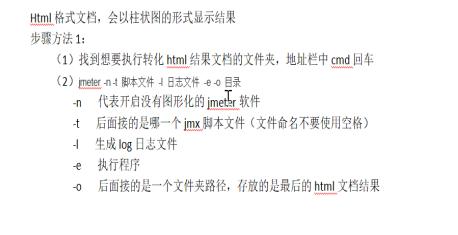
为了实现自动化脚本

步骤:

1. 保证数据库不被我的测试数据所污染:
2. 在执行自动化脚本的时候要走通一遍增删改查
3. 先增加,在修改,在查询,最后删除
4. 写jmeter脚本
5. 增删改查四大功能要放在单独的线程组中,,目的是方便后期修改
6. 先新增,在设置删除, 别忘了改测试计划中”独立运行每个线程组”必须先增后删
7. 因为resful风格添加数据后,会返回我添加过的数据,我们就可以使用后置处理器中的正则表达式匹配想要获取的id值
8. 把id值存放在全局变量中,再删除的时候调用全局变量即可
9. 查和改正常写脚本,要使用全局变量来设置
10. 必须给每一个请求添加断言,保证每个请求如果出现问题,都会爆红

# 了解html结果文档

方法1：



方法2：

