# 一、接口基础

软件测试中的接口测试，是指外部接口（用于与外部系统交互接口）及内部接口（软件系统内部模块/组件或子系统之间[集成测试]）

## 1.接口分类

Https比http协议更安全

http协议接口类型：post、get、put、delete、OPTIONS（请求查询服务器性能）

Post：webform表单形式、Json格式提交数据

Post与get区别：

1. 参数提交形式
2. 安全性不同 ：post更安全
3. 请求数据大小不同：地址栏有数据长度限制

## 2.接口测试流程

1. 需求评审
2. 接口分析
3. 设计接口用例
4. 执行接口测试：查看不同的参数请求，接口的返回的数据是否达到预期
5. 测试报告

（1）为什么设计测试用例

理清思路，避免漏测；跟进测试进度；跟进重复性工作；

1. 用例设计方法【依据接口说明文档】：

* 功能：参数是否满足要求；功能是否正常返回
* 逻辑业务：是否依赖业务；接口依赖上一个步骤；接口关联
* 异常：参数异常（参数多、少、参数错误……）；数据异常（数据为空、错误数据、数据长度……）
* 安全：cookie（删除cookie后，不能返回数据） ；header（删除header信息，不给返回 ）； 唯一识别码（客户端发送给服务端的识别认证）

## 3.接口测试工具

soapUI Jmeter **postman**(常用) Fiddler(抓http包，可设置代理)

Webservice-soap接口、rest接口

## 4.单元测试

（1）单元测试属于白盒测试范畴；

（2）集成测试属于灰盒测试范畴；

（集成测试一般由开发小组采用白盒加黑盒的方式来测试——内部接口测试）

（3）系统测试属于黑盒测试范畴；

（4）接口测试是单元测试、集成测试的一部分；

如何做单元测试

1.编写单元测试代码：对类、方法创建单元测试

2.运行单元测试

3.查看测试结果

## 5.白盒测试

白盒测试是一种测试用例设计方法，基于代码的测试；白盒法需要测试者了解程序内部逻辑结构，对所有逻辑路径进行测试，也就是说，"白盒"法是“穷举路径测试”。

总体上分为静态分析和动态分析两大类。

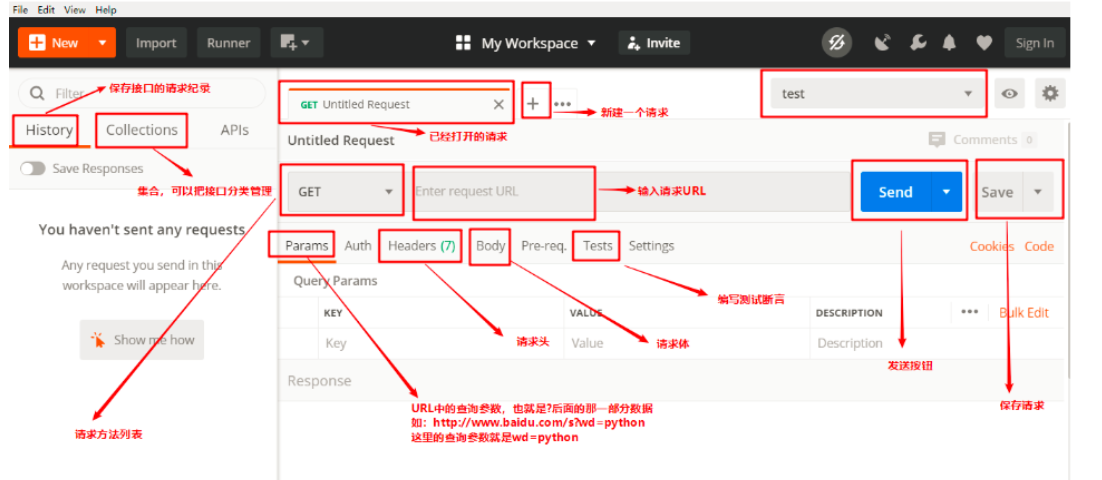
静态分析：不需要执行程序，就可以进行的测试，例如：代码审查、代码扫描

动态分析：需要执行程序，才可以进行的测试，例如：单元测试、覆盖测试

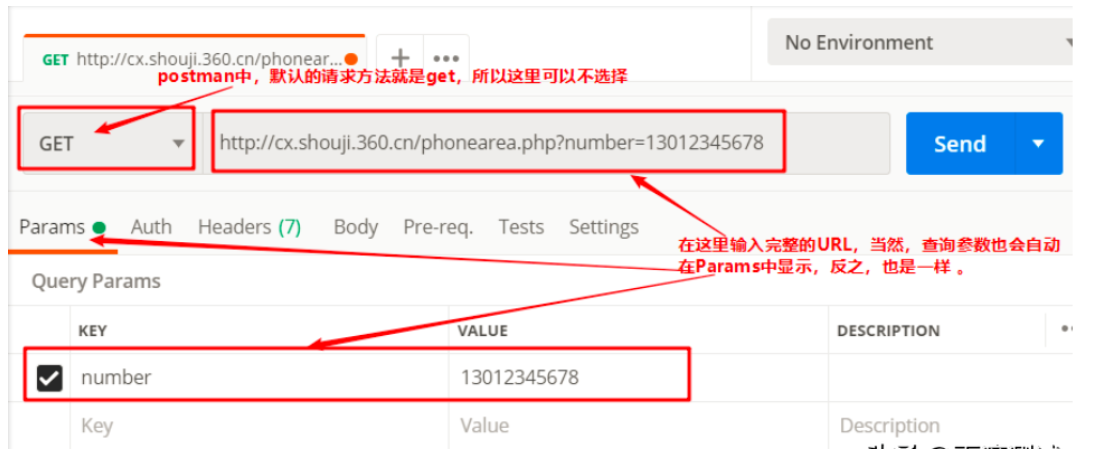
# 二、接口测试方法

## 1.PostMan实现接口测试

整体分布



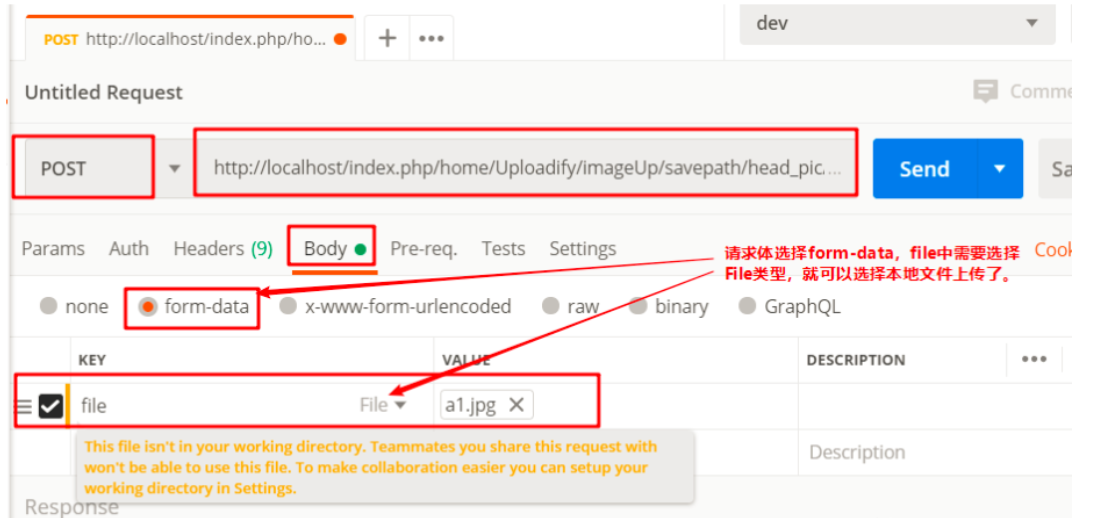
## Get接口



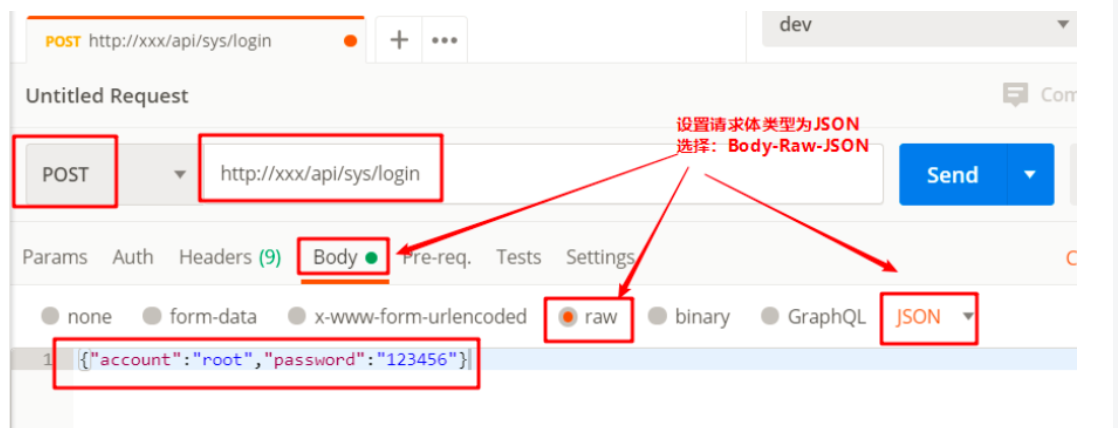
## Post接口（表单类型)



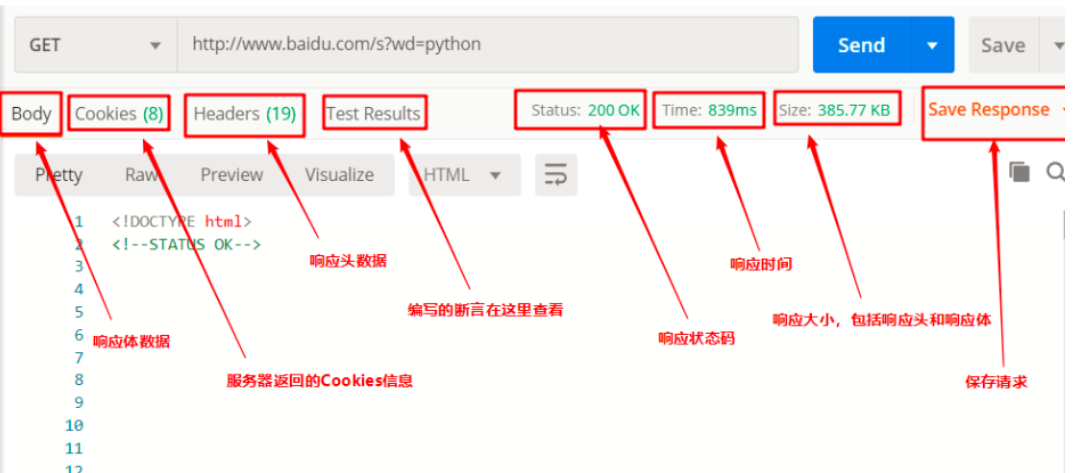
## Post接口(上传文件的表单类型)



## Post接口(Json类型)

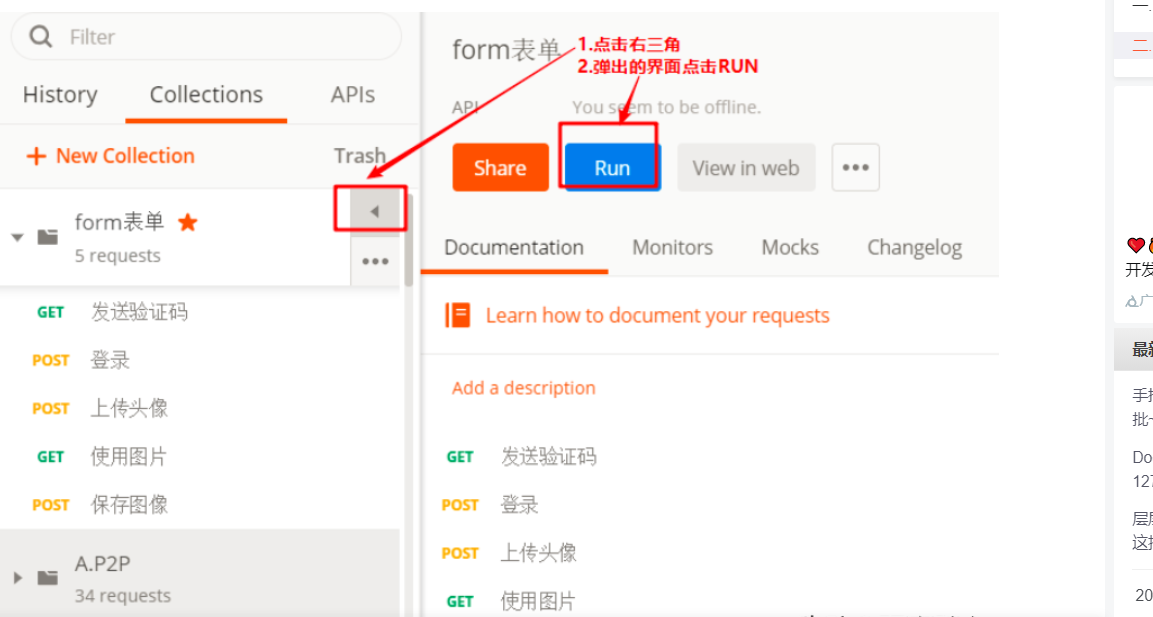


## 接口响应



## 接口管理-collection

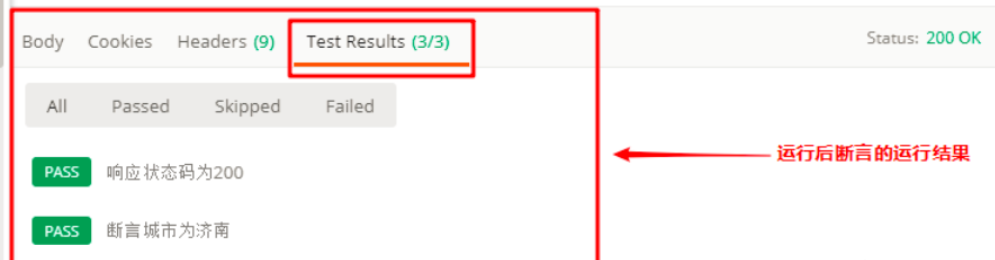
* 用例分类管理，方便后期维护
* 可以进行批量用例回归测试



## 断言

* 断言编写位置：Tests标签
* 断言所用语言：JavaScript
* 断言执行顺序：在响应体数据返回后执行
* 断言执行结果查看：Test Results





## 变量

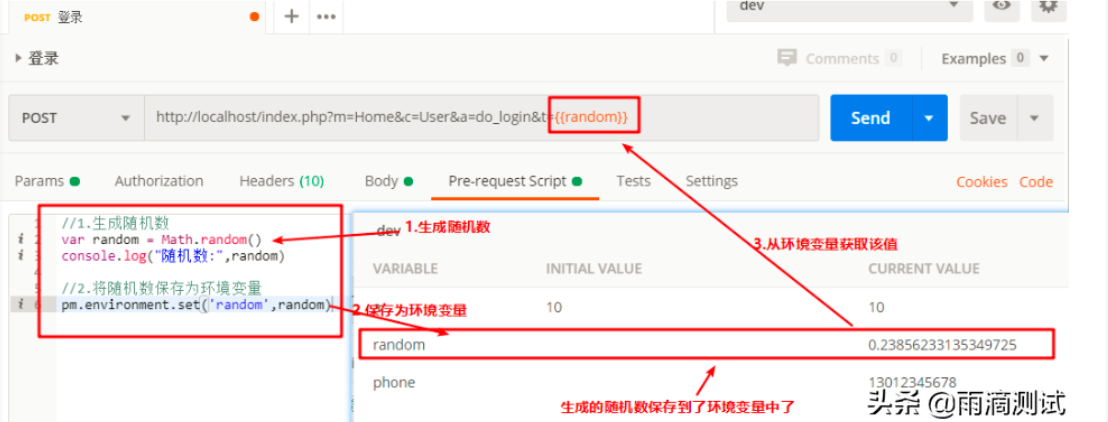
在Tests，Pre-requests Script定义变量：

* 定义全局变量：pm.collectionVariables.set("变量名",变量值)
* 定义环境变量：pm.environment.set("变量名"，变量值)
* 定义集合变量：pm.variables.set("变量名",变量值)

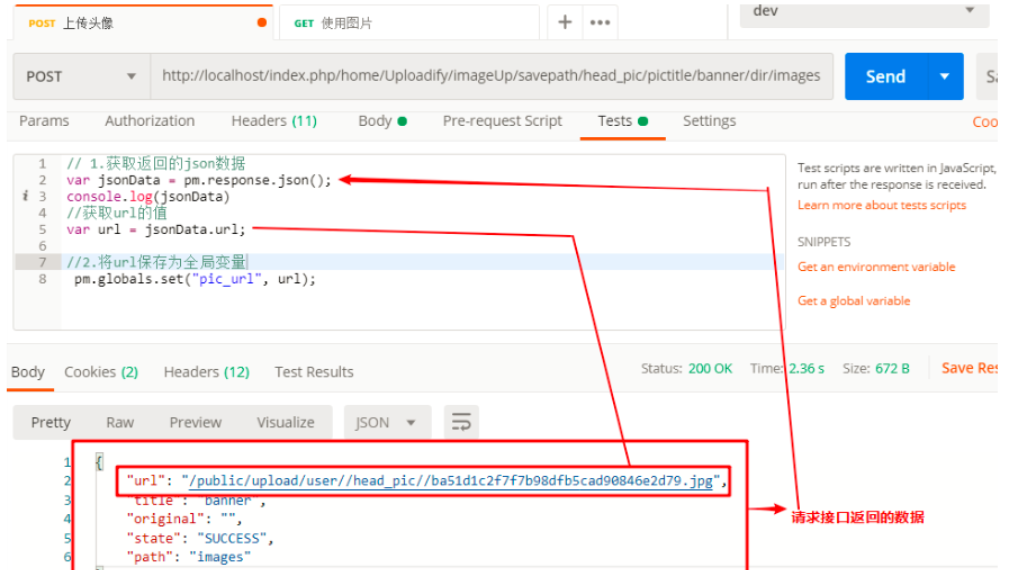
在Tests,Pre-requests Script获取变量：

* 获取环境变量：pm.environment.get(‘变量名’)
* 获取全局变量：pm.globals.get('变量名')
* 获取集合变量：pm.pm.collectionVariables.get.get('变量名')

在请求参数中获取变量，无论是获取全局变量，还是环境变量，还是集合变量，获取的方式都是一样的编写规则：{{变量名}} 。



## 接口关联—从响应中获取变量



## 测试报告（newman执行)

1. 安装newman、nodejs并放在一个文件夹
2. postman导出colection，放在某个文件夹
3. cmd终端下，借助newman命令执行测试，并生成html格式的测试报告

注意：Jemeter工具类似，使用的是Ant

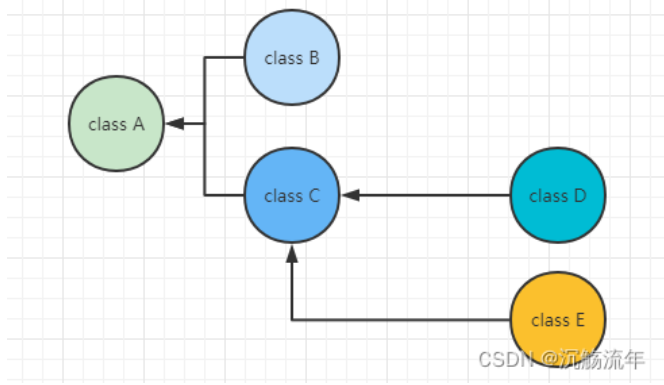
## Jenkins+postman

思路：配置Jenkins的cmd命令，执行postman导出的测试用例脚本，类似于（10）

注意：Jemeter工具类似，配置Jenkins的cmd命令，执行Jemeter导出的脚本

## Mock实现接口测试

Mock 方法 是[单元测试](https://so.csdn.net/so/search?q=%E5%8D%95%E5%85%83%E6%B5%8B%E8%AF%95&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_44614026/article/details/_blank)中常见的一种技术，Mock 就是在测试过程中，对于一些不容易构造/获取的对象，创建一个 Mock对象 来模拟对象的行为。



Class A 要从 Class B 和Class C中获取数据，而 Class C 中该数据又来源于 Class D 和 Class E

这时可以利用 Mock 去构造虚拟的Class B 和Class C对象用于Class A 的测试，因为我们只是想测试class A的行为是否符合预期，并不需要去测试依赖对象。

Mock可以模拟服务端或者客户端等对象。

## 3.SoapUI实现接口测试

## 4.持续集成-代码实现接口测试

**1、Unitest框架**

Unittest+[Requests](https://so.csdn.net/so/search?q=requests&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/songteng2012/article/details/_blank)+Jenkins库实现接口测试

**2、Testng框架**

Testng+HttpClient+Jenkins实现接口测试