

# 接口的请求和响应

- 1 1.HTTP协议基础(超文本传输协议)
- 2 1.HTTP协议也叫超文本传输协议
- 3 2.是用于从万维网(www:world wide web)服务器传输超文本到本地浏览器的传送协议;
- 4 3.HTTP是一个基于TCP/IP通信协议来传递数据(HTML 文件, 图片文件, 查询结果等)
- 5 2.HTTP协议工作原理
- 6 1.HTTP协议工作于浏览器端-服务端(BS)架构上。
- 7 2.浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送请求。(记住,重点)
- 8 3.Web服务器有:Apache服务器, tomcat, IIS服务器(微软的)(Internet Information Services)等。
- 9 4.Web服务器根据接收到的请求后, 向客户端发送响应信息。(记住,重点)
- 10 5.HTTP默认端口号为80(记住), 但是你也可以改为8088或者其他端口。
- 11
- 12 3.HTTP协议注意
- 13 1.HTTP是无连接的: 无连接的含义是限制每次连接只处理一个请求。
- 14 2.HTTP是媒体独立(超文本)的: 这意味着, 只要客户端和服务器知道如何处理的数据内容, 任何类型的数据都可以通过HTTP发送。
- 15 3.HTTP是无状态: HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息, 则它必须重传, 这样可能导致每次连接传送的数据量增大。另一方面, 在服务器不需要先前信息时它的应答就较快。(如刷新页面)
- 16
- 17 4.HTTP协议结构
- 18 1.HTTP是基于客户端/服务端(B/S)的架构模型, 通过一个可靠的链接来交换信息, 是一个无状态的请求/响应协议。
- 19 2.一个HTTP"客户端"是一个应用程序, 通过连接到服务器达到向服务器发送一个或多个HTTP的请求的目的。
- 20 3.一个HTTP"服务器"同样也是一个应用程序通过接收客户端的请求并向客户端发送HTTP响应数据。
- 21 4.HTTP使用统一资源标识符(Uniform Resource Identifiers, URI)来传输数据和建立连接。
- 22 5.一旦建立连接后, 数据消息就通过类似Internet邮件所使用的格式[RFC5322]和多用途Internet邮件扩展(MIME)[RFC2045]来传送。
- 23
- 24 5.HTTP的请求结构
- 25 客户端发送一个HTTP请求到服务器的请求消息包括以下格式: 请求行(request line)、请求头部(header)、空行和请求数据四个部分组成
- 26 请求行, 请求头部, 空行, 请求数据
- 27
- 28 6.HTTP的请求方法
- 29 1.get
- 30 2.post
- 31
- 32 7.HTTP的响应结构
- 33 HTTP响应也由四个部分组成, 分别是: 状态行、消息报头、空行和响应正文
- 34 8.HTTP的响头(了解)
- 35 9.HTTP响应状态码
- 36 1.当浏览者访问一个网页时, 浏览者的浏览器会向网页所在服务器发出请求。当浏览器接收并显示网页前, 此网页所在的服务器会返回一个包含HTTP状态码的信息头(server header)用以响应浏览器的请求。
- 37 HTTP状态码的英文为HTTP Status Code。
- 38 2.常见状态码
- 39 200 - 请求成功
- 40 301 - 资源被永久转移到其它URL

41           404 - 请求的资源（网页等）不存在  
42           500 - 内部服务器错误

43  
44  
45   General : 通用头

46  
47       Request URL : 当前请求的请求地址

48  
49       Request Method : 请求类型 get、post、put、delete等

50  
51       Status Code : 响应状态码 200、404、503等

52  
53       Remote Address : 域名对应的真实ip:port

54  
55   Response Headers : 响应头

56  
57       Access-Control-Allow-Origin : 在服务器端设置这个属性来控制允许跨域的域名

58  
59       Cache-Control : 请求和响应遵循的缓存机制

60  
61       Connection : 是否保持与服务器的tcp长连接。keep-alive(默认)、close。keep-alive代表服务会保留当前连接一段时间被其他请求重复使用；close代表请求之后关闭连接。

62  
63       Content-Length : 响应体的长度

64  
65       Content-Type : 响应内容的格式/类型text/html;charset=UTF-8标识返回的内容是文本类型，html格式。

66  
67       Date : 消息发出时间(GMT) Sat, 23 Apr 2016 07:47:54 GMT GMT+8=北京时间

68  
69       Expires : 响应过期时间

70  
71       Pragma : 跟Cache-Control: no-cache相同。Pragma: no-cache兼容http 1.0 , Cache-Control: no-cache是http 1.1提供的。因此，Pragma: no-cache可以应用到http 1.0 和http 1.1 , 而Cache-Control: no-cache只能应用于http 1.1

72  
73       Server : 服务器

74  
75       Set-Cookie : 设置cookie

76  
77       X-Powered-By : 表示是用什么语言或者框架编写的，比如：PHP/5.6.30

78  
79   Request Headers : 请求头

80  
81       Accept : 客户端/发送端能够接收的数据类型  
text/html,application/xhtml+xml,application/xml

82  
83       Accept-Encoding : 浏览器可以支持的- web服务器返回内容压缩编码类型 gzip, deflate

84  
85       Accept-Language : 浏览器可接收的语言 zh-CN,zh;q=0.8

86  
87       Connection : 是否保持与服务器的tcp长连接。keep-alive(默认)、close。keep-alive代表服务会保留当前连接一段时间被其他请求重复使用；close代表请求之后关闭连接。

```
88
89     Content-Length : 请求体的长度
90
91     Content-Type : 请求内容的格式/类型 application/x-www-form-urlencoded
application/json
92
93     Cookie : 客户端缓存的Cookie , 在请求发送时会发送到服务端
94
95     Host : 请求的服务器域名
96
97     Origin : origin主要是用来说明最初请求是从哪里发起的 ; origin只用于Post请求 , 而Referer则用
于所有类型的请求 ;
98
```

## 常用的接口测试工具

```
1  1  常用浏览器 ( 接口测试、报文捕获及解析 ) (没有接口文档的时候)
2  2  Postman简介
3      Postman是比较完整的API测试开发环境 , 可以使得API测试开发变得快速、容易使用
4  3  Jmeter简介
5      Apache JMeter是Apache组织开发的基于Java的压力测试工具。用于对软件做压力测试 , 它最初被设计用
于web应用测试 , 但后来扩展到其他测试领域。能够对HTTP和FTP服务器进行压力和性能测试 , 也可以对任何数据
库进行同样的测试 ( 通过JDBC )
6      (全但不精,主要是性能测试,但也可以进行接口测试)
7  4  Fiddler简介
8      Fiddler是一个http协议调试代理工具 , 它能够记录并检查所有你的电脑和互联网之间的http通讯 , 设置断
点 , 查看所有的“进出”Fiddler的数据 ( 指cookie,html,js,css等文件 )。
9      常用来抓包,也可以进行接口测试
10
11
```

## 接口测试用例的设计(重点)

```
1  1.用例名称
2  2.接口地址
3  3.请求方式
4  4.开始:有就写,没有就不写,要和开发人员沟通
5  5.结束:有就写,没有就不写,要和开发人员沟通
6  6.描述:目的(一般先正向,然后再反向)
7  7.HTTP头参数:(HTTP的响头)
8  8.HTTP参数:(传输的数据)
9  9.json参数:(传输的数据)
10 10.状态码:如200,404等
11 11.返回结果:(返回的数据)
12 12.数据库查询是否存在:有就填写,没有就不填
13
```

##

## postman安装

1 | 看安装文档

## postman的介绍

1 | 看文档

## postman发送无参get请求(重点)

1 步骤：  
2 1. 打开postman  
3 2. 创建一个collection, 用于保存所有的接口请求  
4 3. 创建一个请求, 请求方法为get, 请求的url为http://www.baidu.com  
5 4. 点击发送(send)按键  
6 5. 在Response区域, 分析服务器返回的信息  
7 Body区域: 显示服务器返回信息的主体  
8 cookies区域: 服务器返回的所有cookie值(用于验证客户端, 服务端身份信息)  
9 Headers区域: 服务器返回的响应头部信息  
10 Test Results: 断言, 判断服务器返回的信息是否正确(后面会重点学习, 自动化测试的)  
11 响应的其他信息: Status(状态码), Time(时间), Size(响应信息的大小)  
12 6. 请求案例内容保存(ctr+s)  
13  
14 课堂案例：  
15 创建一个请求, 请求方法为get, 请求url为http://www.taobao.com  
16  
17  
18 后台程序案例：

## postman发送有参的get请求(重点)

1 | 参数会在url中显示

## postman发送post请求(重点)

1 1. 表单  
2 2. json

## postman接口测试流程(重点)

- 1 1.获取基本请求四大信息：url,请求方式,参数,请求头
- 2 2.根据参数编写测试用例
- 3 3.使用postman工具设计并执行测试用例
- 4 4.对返回的结果进行分析

## 百度翻译实战案例(重点)

```
1  请求方式get
2  url:https://fanyi-api.baidu.com/api/trans/vip/translate
3  参数:q,from,to,appid,salt,sign(appid+q+salt+密钥的MD5值)
4  响应:
5  {
6      "from": "en",
7      "to": "zh",
8      "trans_result": [
9          {
10             "src": "apple",
11             "dst": "苹果"
12         }
13     ]
14 }
15
16 APP ID : 20210507000816969
17
18 密钥 : sGZsjo0Y20ThznTznsRT
19
20 签名:20210507000816969apple123456sGZsjo0Y20ThznTznsRT
21 dd9a3de579186139fb8964f76f8ae898
22
23 https://md5jiami.bmcx.com/ (md5加密网站)
24 案例一:英文转中文
25 1.阅读api文档
26 2.appid的参数,只有注册为开发者才有,百度管理控制台有显示
27 3.密钥,只有注册为开发者才有,百度管理控制台有显示
28 4.通过文档获取url
29 5.通过文档获取参数
30 q:apple
31 from:en
32 to:zh
33 appid:xxxxxx
34 salt:xxxxx(随机数)
35 sign:(是一个MD5的加密字符串)
36 6.MD5加密:通过工具对字符串进行加密(百度一下,加密工具网站)
37 7.把参数输入到postman中,send一下
38 8.对比请求结果
39 20210507000816969苹果123456sGZsjo0Y20ThznTznsRT
40 appid+q+salt+密钥的MD5值
41 946dd86366e03c0b6eb5da8b04a344be
42
43 案例二:中文转英文
44 1.准备参数
```

```
45 q:春节(注意,url中不能有中文,需要把中文转换为URL encode)(百度一下,搜索转换的网站)
46 from:zh
47 to:en
48 appid:xxxxxx
49 salt:xxxxx(随机数)
50 sign:(是一个MD5的加密字符串)(注意,q的参数直接写中文进行加密)
51
```

## 参数化运行原理

```
1  自动化测试的两大核心:
2  1.参数化:
3  在程序中使用变量代替常量的过程
4  循环实现脚本的运行
5  数据的来源可能是程序本身的集合,字典等,或外部文件中(csv,text)的数据,进行接口测试
6
7  2.
8  断言
9  脚本中添加相应的判断,实现脚本的自动化比较(判断),程序的实际结果和预期结果
10 断言有自定义的和框架内置的
```

## postman的参数化运行(重点)

```
1  步骤:
2  1.准备参数化的文件.csv文件
3  2.在postman中创建请求
4  以整个请求的集合为一个主体,把要参数化运行的请求单独放到一个集合中
5  3.使用设置好的变量名代替相对应位置的常量.postman中用{{}}把变量名包含起来.列如:{{u_id}}
6  4.在postman中,启动runner,运行collection
7  5.选择要进行参数化运行的集合
8  6.设置运行效果
9  设置运行的迭代次数,一般你有多少个数据就填多少
10 设置延时,请求和请求之间的时间
11 选择数据文件,你存储数据的文件
12 设置文件类型和查看预览效果
13 其他信息设置,默认就可以
14 7.开启运行脚本,查看和等待结果
15 8.结果显示:0成功,0失败,是因为我们还没有添加断言,只是运行了,没有任何比较,所以是没有价值的
16
17 用后台程序的案例,get带参的演示
18 注意:创建csv文件时,先创建.txt,然后改格式,输入数据后另存为逗号隔开的csv
19
```

## Json介绍

```
1  JSON的定义
2  JSON 指的是 JavaScript 对象表示法 (JavaScript Object Notation)
3  JSON 是轻量级的文本数据交换格式
4  JSON 独立于语言 *
5  JSON 具有自我描述性，更易理解
6  JSON 使用 JavaScript 语法来描述数据对象，但是 JSON 仍然独立于语言和平台。JSON 解析器和 JSON 库
  支持许多不同的编程语言。
7
8
9
10 JSON的特点：
11
12 JSON 是纯文本
13 JSON 具有“自我描述性”（人类可读）
14 JSON 具有层级结构（值中存在值）
15 JSON 可通过 JavaScript 进行解析
16 JSON 数据可使用 AJAX 进行传输
17
18 JSON 语法规则
19 JSON 语法是 JavaScript 对象表示法语法的子集。
20
21 数据在名称/值对中
22 数据由逗号分隔
23 花括号保存对象
24 方括号保存数组
25
26
27
28
29 JSON 文件
30 JSON 文件的文件类型是 ".json"
31 JSON 文本的 MIME 类型是 "application/json"
```

## postman的断言(重点)

```
1  1.什么是断言:就是判断
2  2.postman的断言
3  postman有个叫tests模块,就是对http响应结果进行判断的地方
4  好处是:postman有很多断言
5  坏处是:要敲代码
6  postman支持javascript语言,包含了一个很强大的node.js
7  postman脚本应用场景:
8  a.断言(tests模块),作用是在接受http响应后进行处理和运行
9  b.预处理信息(Pre-request-script)作用是在接受http响应前进行处理和运行,可以配合参数化进行.
10 通过制定的脚本运行(读取参数文件)
11 c.控制请求的运行,比如跳转到某一个请求,或者循环执行一个请求
12 脚本运行顺序(了解):b->c->a
13
14 案例:配合参数化使用断言
15 1.使用的是后台程序,get有参请求,我们把name的值使用外部文件的数据,就叫参数化
16 2.在请求所在的集合中定义变量aa
17 3.在请求的测试模块中,输入以下代码:
```

```

18 //获取集合变量
19 n=pm.collectionVariables.get("aa");
20 pm.test("Your test name", function () {
21     var jsonData = pm.response.json();
22     var v=jsonData.data;
23     pm.expect(v).to.eql(n);
24
25     //把获取的集合变量转换为数字,才好进行加1
26     n=Number(n)
27     //设置集合变量
28     pm.collectionVariables.set("aa", String(n+1));
29 });
30 4.在集合的运行模块,导入外部文件,然后运行,可以看到所有请求都通过
31
32
33 3.使用断言
34 案例一:
35 1.使用后台程序,显示首页接口
36 2.创建请求,输入url
37 3.选择tests模块,点击右边的Status code:Code is 200和Response time is less than 200ms
38 分别表示返回的code要是200,响应的时间小于200ms
39 4.点击send,看返回的结果的Test Results
40 注意:断言里的数据是可以更改的,内置断言太多,具体用到什么可以百度
41
42 案例二:判断返回的json数据
43 1.使用后台程序,显示首页接口
44 2.创建请求,输入url
45 3.选择tests模块,点击Response body:Json value check,改为下面的代码
46 pm.test("Your test name", function () {
47     var jsonData = pm.response.json();
48     var v=jsonData.msg;(如果json中有数组,用[]下标方式获取)
49     pm.expect(v).to.equal("成功");
50 });
51 4.点击send,看返回的结果的Test Results
52
53 响应的时间小于200ms
54 tests['Response time is less than 200ms'] = responseTime < 500;(了解一下)
55
56
57 案例:判断百度翻译api翻译的内容是否正确
58 在请求的测试模块中,右侧点击 "响应正文:JSON值检查",然后修改代码为如下:
59 pm.test("Your test name", function () {
60     var jsonData = pm.response.json();
61     var v=jsonData.trans_result[0].dst
62     pm.expect(v).to.eql("Apple");
63 });
64

```

## postman的环境变量



```
1 1.提前把数据存储到变量中,后期使用的时候直接用
2 2.postman的右上角的"眼睛"
3 3.Globals和Environment
4 Globals:全局环境变量,所有的请求都可以用
5 Environment:局部环境变量,要配置到某个请求中
6 使用操作看文档
7
```

## postman的脚本导出

```
1 1.在请求的最右边的</>,点击它,然后选择python-requests
2 2.复制代码
3
4 import requests
5
6 url = "http://127.0.0.1:8808/api/block/register?
   name=88&passwd=123456&email=76754438@qq.com"
7
8 payload={}
9 headers = {}
10
11 response = requests.request("GET", url, headers=headers, data=payload)
12
13 print(response.text)
14
15
```

## requests简介及安装

```
1 简介
2 requests是Python HTTP 库,python使用这个库来操作有关http的事务
3 Keep-Alive & 连接池
4 国际化域名和 URL
5 带持久 Cookie 的会话
6 浏览器式的 SSL 认证
7 自动内容解码
8 自动解压
9 Unicode 响应体
10 文件分块上传
11 ....
12
13 安装:
14 cmd中或pycharm中
15 cmd中:
16 打开cmd,输入:pip install requests
17
```

## requests函数和属性介绍(重点)

```
1  函数:
2  1.request:构造一个请求,支持其他方法
3  2.get:获取html的主要方法,至少一个参数,有返回值,返回值就是这次请求的响应结果
4  3.post:向html提交post请求
5
6  属性:
7  response.url:返回请求网站的 url
8  response.status_code:返回响应的状态码
9  response.encoding:返回响应的编码方式
10 response.cookies:返回响应的 Cookie 信息
11 response.headers:返回响应头
12 response.content:返回 bytes 类型的响应体(二进制方式)
13 response.text:返回 str 类型的响应体,相当于 response.content.decode('utf-8')
14
```

## requests的get无参(重点)

```
1
2  import requests
3
4  def mytest01():
5      res=requests.get("https://www.baidu.com")
6      print(res.status_code)
7      res.encoding='utf8'
8      print(res.text)
```

## requests的get有参(重点)

```
1
2  import requests
3  def mytest02():
4      data={"name":"aaa","passwd":"123456","email":"242432@qq.com"}
5      res=requests.get("http://127.0.0.1:8808//api/block/register",params=data)
6      print(res.text)
7
8  mytest02()
```

## requests的post(重点)

```
1  import requests
2
3  def mytest01():
4      data={"name":"aaa","passwd":"123456"}
5      res=requests.post("http://127.0.0.1:8808//api/block/login",data=data)
6      print(res.text)
7
8  mytest01()
9
10 json请求:
```

```
11 import requests
12
13 def mytest01():
14     data={"name":"aaa"}
15     res=requests.post("http://127.0.0.1:8808//api/block/msg",json=data)
16     print(res.text)
17
18
19 mytest01()
```

## 将结果转换为json格式

---

```
1 import requests
2
3 def mytest01():
4     data={"name":"aaa"}
5     res=requests.post("http://127.0.0.1:8808//api/block/msg",json=data)
6     print(res.text)
7     rj=res.json()
8     print(rj['data']['money'])
9
10 mytest01()
11
12
13 作业:
14 1. 请获取网易首页的内容,判断是否有新闻这两个汉字,分别使用postman和requests方式实现测试用例
```