接口测试基础

目录

2、用例集：Collection　不同接口 管理多条用例
3、文件夹：Folder　　子接口、协议、方法
4、请求：Request　　每一条用例

课程介绍

一、接口测试基础（接口测试工具）

1、接口的本质

2、网络协议和接口的关系

3、Postman

4、Fiddler

5、JMeter

二、接口测试高级（接口自动化测试，Python+requests）

基本内容：线性编码、函数、类、模块、包、测试框架pytest、unittest...

Day01：测试环境搭建、接口基础、postman基本使用

一、环境资料下载和安装

1、首先下载和安装vmware

说明：苹果笔记本用不了

2、下载并解压Windows 10 x64.rar

（1）虚拟机

（2）学习过程中出错，很方便恢复到原始状态

（3）如果真机内存不小于8G，可以使用，否则不建议使用

（4）解压以后，不要往windows 10 x64目录中放任何其他资料

（5）测试工具安装到windows 7/10中

（6）打开windows 10 x64虚拟机

（7）打开以后做快照（相当于还原点）

3、下载并解压WA服务器.rar

（1）被测服务器

（2）被测软件已经安装好

（3）解压、打开（调快照）

（4）查询服务器IP

ifconfig ens33 …

每个人的IP可能会相同

（5）关于数据库

数据库用户名：root，密码：123456

数据库名：wa_test

表：users用户表、info个人信息表

4、安装测试工具

参见《接口测试工具安装.pdf》

5、windows10虚拟机联网方式

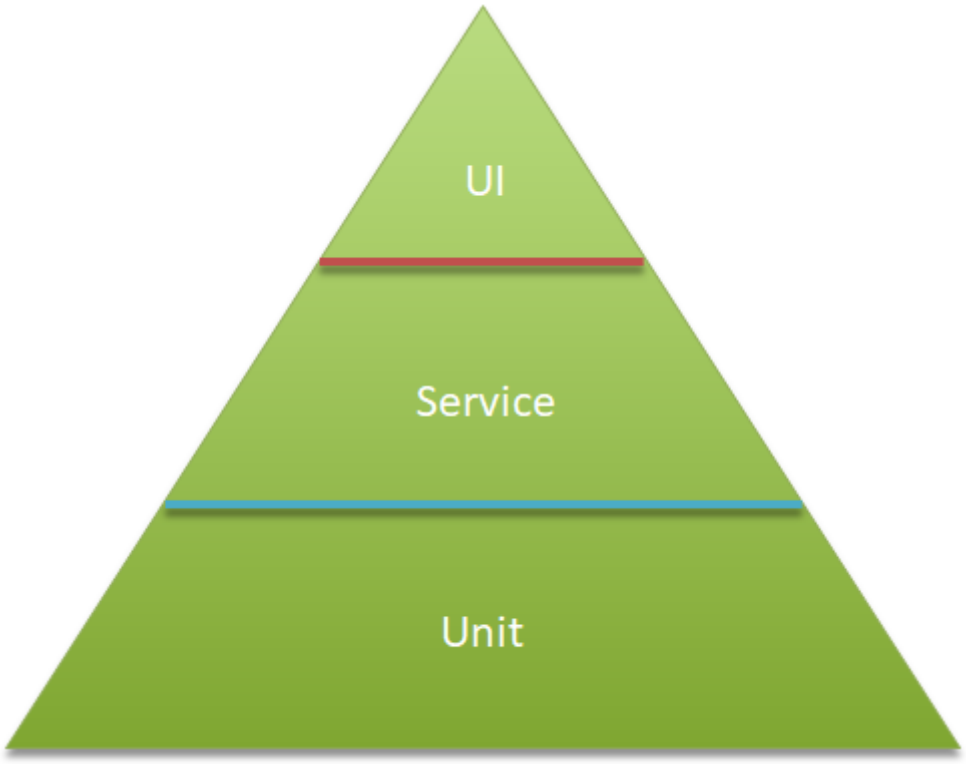
需要设置虚拟机网络连接模式为：桥接（win10虚拟机和WA虚拟机都要做）

之后，给WA服务器拍摄快照

重新查询WA服务器的IP

二、接口基础知识

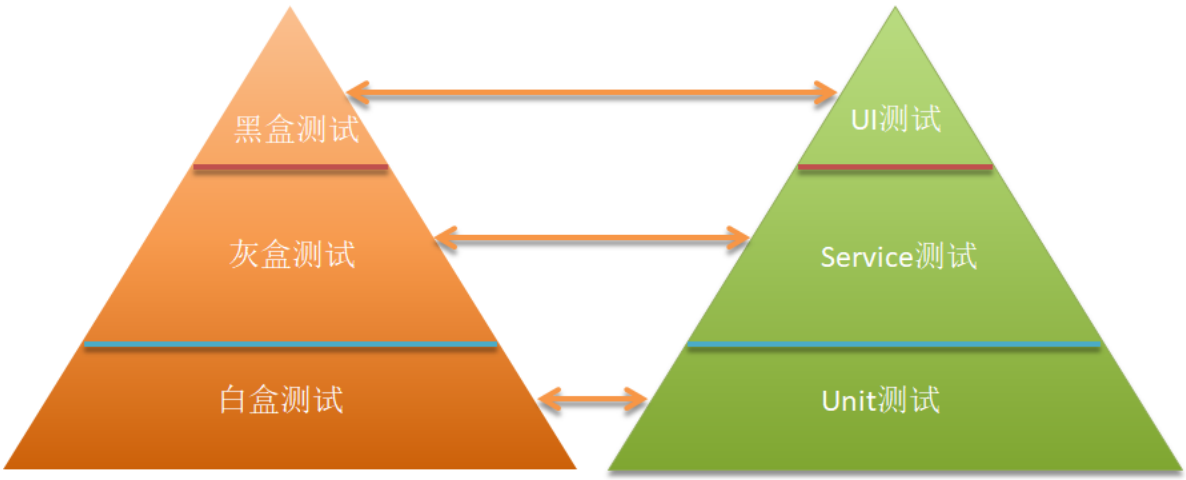
1、分层测试技术/金字塔模型



1）UI层：用户界面(User Interface）便于上手，但发现问题的能力较弱

2）Service层：服务层，也就是接口层，可进行接口测试，要求技术门槛，但是发现问题能较强

3）Unit层：单元层，可进行单元测试，更早介入，更底层，发现问题能力强，技术要求高



2、接口重要概念

（1）接口

目前特指软件接口，以Web项目为背景，主要是Web接口

接口也叫API（Application Programming Interface）应用程序编程接口

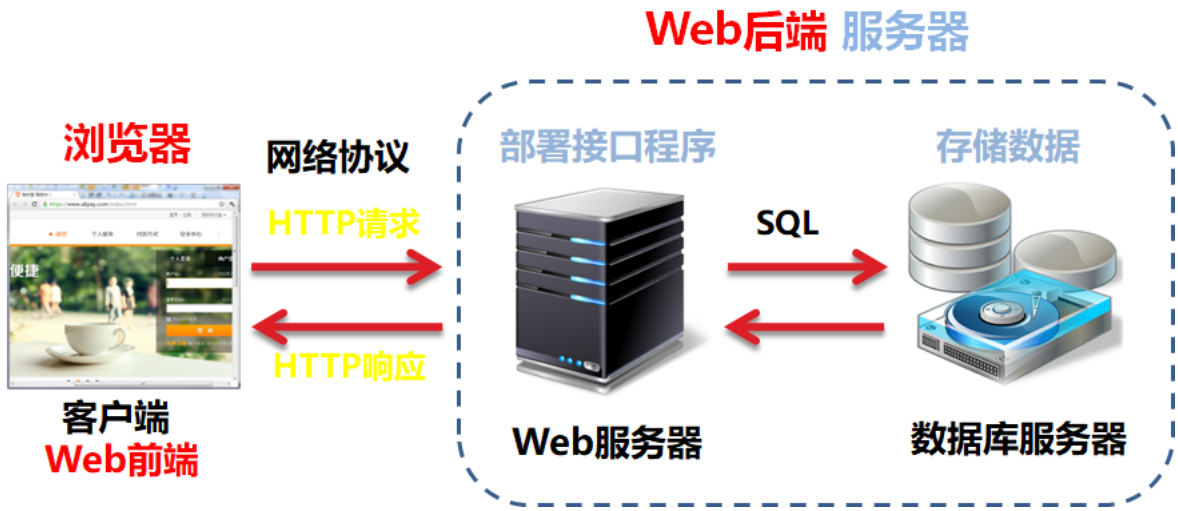
本质上是开发编写的函数或方法，是对某种业务功能的设计和实现

（2）UI测试和接口测试的区别

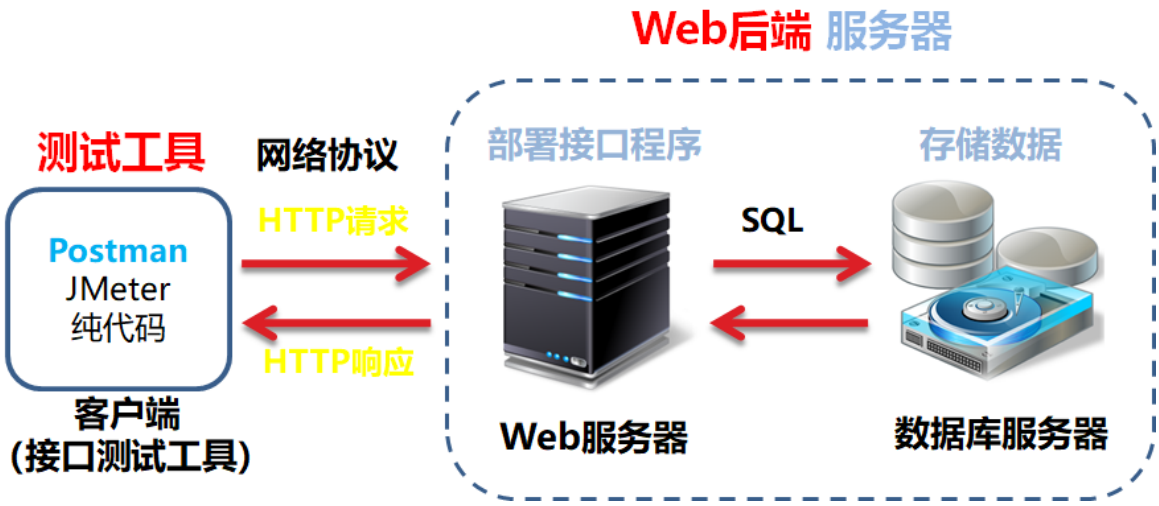
Web前端：以浏览器为代表的技术，比如HTML、JavaScript、CSS，都是在浏览器端执行，出现UI效果

Web后端：有后台服务器开发技术构成，比如Java、数据库、SQL，接口程序在后台执行

<1> UI测试的原理（有用户界面，更直观，减轻用户的访问负担）



<2> 接口测试的原理（没有用户界面，使用专业的接口测试工具充当界面，要求以更专业的方式访问接口）



（3）客户端、客户机

<1> PC（个人计算机）

<2> 安装浏览器，比如：Chrome、Edge、Firefox、IE等

<3> 安装测试工具，比如：Postman、Fiddler、JMeter、Python

<4> 通过某种协议、规则访问服务器

（4）服务器

<1> 分为Web服务器、数据库服务器等

<2> Web服务器：运行接口程序的载体，是Web应用程序的"容器"，内部安装了许多接口程序

<3> 数据库服务器：大量业务数据长期的、海量存储在数据库中，比如MySQL数据库，是接口程序需要访问的"后台"

（5）请求（Request）

客户端向服务器发送HTTP请求（HTTP Request），为了获取服务器的资源、访问服务器的接口程序；请求是将客户端的数据发给接口程序进行处理。

请求包分为请求头部和请求主体：

<1> 请求头: header

请求的附加信息，如客户端的操作系统、客户端使用的浏览器、客户端能接受的字符编码、授权、客户端是否携带Cookie信息等；

包括访问接口的三要素：请求方法（Get或Post）、接口地址、请求参数

<2> 请求主体：body

Post方法发送的数据放在请求主体中携带，Get方法没有请求主体。

（6）响应（Response）

服务器将接口程序处理的数据结果发送给客户端的过程。

响应包分为响应头部和响应主体：

<1> 响应头: header

响应的附加信息，如HTTP响应状态码、响应类型（网页HTML、json文本、xml文本）、是否要求客户端设置Cookie信息等；

<2> 响应主体：body

响应的正文部分，是接口返回值，是测试时最重要的部分。

3、接口的组成要素

（1）接口地址（url地址，url：统一资源定位器）

<1> 形式

http://172.16.100.65:8000/sign/add_event/

<2> 协议

HTTP（超文本传输协议）、HTTPS（安全的超文本传输协议）、FTP（文件传输协议）、SMTP（简单邮件传输协议）、POP3（邮局协议第3版本）

测试时，用什么协议，看需求

<3> 服务器IP，如172.16.100.65

测试时，服务器IP是什么，取决于搭建环境的人员

<4> 端口号，如8000

端口号是用一个数字表示使用服务器上的何种软件或服务

测试时，端口号是什么，取决于搭建环境的人员

HTTP的默认端口号是80，HTTPS的默认端口号是443

<5> 应用名，如sign

被测软件的名字或简称，有的接口路径比较多，会加版本号等

测试时，应用名是什么，取决于需求

<6> 接口名，如add_event

被测接口的名字，实际测试时，接口名字很多

测试时，接口名是什么，取决于需求

接口名/末尾的/能否省略，看需求

（2）请求方法

<1> 如何发送和处理数据

<2> 常见的请求方法

get：主要用于获得资源

post：主要用于提交资源，可以做增加、修改、删除

put：主要用于更新资源，一般是全部更新

patch：主要用于更新资源，一般是部分更新

delete：主要用于删除资源

测试时，使用哪种请求方法，取决于需求

（3）请求类型

post有请求类型，get没有

form表单：x-www-form-urlencoded，把要发送的数据以a=1&b=2的形式放入请求体中

json：application/json，把要发送的数据以b'{"a":1, "b":2}'的形式放入请求体中

form-data：multipart/form-data，用于发送文件，把要发送的数据以b'文件内容'的形式放入请求体中

测试时，选择哪种请求类型，取决于需求

（4）参数

客户端发送给服务器，要求服务器上的接口程序处理的数据

<1> 对于get方法

参数放在url地址中

http://172.16.100.64:8000/sign/add_event/?id=1&name=zhsan

?用于区分接口地址和参数，?后面是参数

id、name是参数名，来源于需求

1、zhsan是参数值，来源于用例

&用于区分不同的参数

<2> 对于post方法

参数放在请求体中

具体形式，参考请求类型

测试时，接口有无参数、参数的含义、哪些参数是必须的、哪些参数是可选的、参数类型是什么、参数有无约束，都要看需求

（5）返回值/响应结果

1）响应状态码/http状态码

200：成功响应，表示服务器成功把结果发给了客户端，但不清楚结果是否正确

301：永久重定向，如网址发生永久改变，访问旧网址时自动跳转到新地址

302：临时重定向，如登录成功后，跳转到新页面

400：域名/服务器地址不存在或者请求错误（主要是参数错误）

401：需要授权（必须先提供有效认证，才能访问接口）

403：客户端错误，如客户端ip被封禁、客户端无读写权限、客户端证书错误

404：客户端提供的接口地址错误，如应用名、接口名、/错误

500：内部服务器错误，如服务器代码错误、服务器重启或关机、服务器太忙

2）响应类型

text/html：网页源码字符串

json：json字符串

xml：xml页面源码字符串

关于json：

javascript object notation，javascript对象表示法

json是一种存储数据的方式，是目前接口返回值的主要形式

json独立于编程语言

形式：

{"name": "Tom", "age": 20, "gender": "male", "ismarried": true, "child": null}

要求：

大括号中保存对象

数据是以键值对形式存在，键的数据类型必须是字符串，字符串定界符必须是双引号

数据对之间用逗号分割

逻辑值/布尔值：true、false

空值：null

数组：[ ]

对比python字典：

{'name': 'Tom', 'age': 20, 'gender': 'male', 'ismarried': True, 'child': None}

字符串用单双引号定界均可

逻辑值/布尔值：True、False

空值：None

列表：[ ]

3）响应正文/体

测试时，必须验证响应正文中数据的正确性（符合需求）

4）对数据库的影响

测试时，必须确保数据库的数据时正确的

三、接口测试与接口自动化测试

1、接口测试

测试接口的功能、性能、安全性

2、为什么做接口测试

原因在于目前开发大多都是前后端分离

界面/ui测试：只关注了前端

接口测试：绕过前端，对后端的api进行测试

3、接口测试的方法

（1）接口测试工具

初级的自动化测试

Postman、fiddler、JMeter

（2）高级自动化测试

python+requests、Java

4、接口自动化测试

（1）靠编码实现测试

（2）自动准备测试数据、自动执行用例、自动验证响应结果的正确性、自动验证数据库数据的正确性、自动生成测试报告、自动发送邮件

（3）接口测试的步骤

需求分析

编写测试用例

编写并调试代码/脚本

搭建测试环境

执行测试

四、Postman基本使用

1、postman简介

postman是一个页面调试工具，发送请求，可以用作接口测试工具

2、postman使用步骤

（1）创建一个工作区（WorkSpace）

类似于一个项目

（2）创建测试集

类似于一个模块、子模块

（3）创建文件夹（Folder）

将不同接口用例分门别类管理

（4）创建请求

对应于一个接口，表示执行某条用例

需要接口地址、请求方法、参数（post还要指定请求类型）

（5）发送请求

客户端把请求（大多数都携带参数）发给服务器

拿到响应结果，判断响应结果正确性

3、案例与练习

案例01：get方法访问百度

接口地址：https://www.baidu.com

请求方式：get

参数：无

返回格式：text/html

案例02：get方法访问京东商城

接口地址：https://list.jd.com/list.html?cat=9987,653,655

请求方式：get

参数：提供商品类别编号

参数名：cat　　参数值：9987,653,655

返回格式：text/html

预期结果：含有手机类别有关的商品页面

练习03：get方法访问无参接口

接口地址：http://接口服务器 IP/apitest/ui-login/

请求方式：get

参数：无

返回格式：text/html

预期包含文本：用户名

练习04：get访问无参接口

接口地址：http://接口服务器 IP/apitest/one-param/

请求方式：get

参数：无

返回格式：text/html

预期包含文本：请使用ID参数进行访问

案例05：get访问无参返回json的接口

接口地址：http://www.httpbin.org/get

请求方式：get

参数：无

返回格式：json

预期包含 json 对象，形如{"key1":"value1", "key2":"value1", ...}

练习06：get获得json数据

接口功能：获得 json 数据

接口地址：http://服务器 IP/apitest/get-json/

请求方式：get

参数：无

返回格式：json

预期：包含 json 对象/json 字符串

案例07：get访问有参接口

接口功能：根据用户 id 查询用户名

接口地址：http://IP/apitest/one-param/

请求方法：get

参数：id（含义：用户编号）

数据库和表：用户信息存储在 wa_test 数据库的 users 表中

响应类型：text/html

预期返回：显示用户名、显示用户信息不存在

练习08：multi-params 接口

接口功能：根据用户 id、 username 查询用户注册时间

接口地址：http://IP/apitest/multi-params/

请求方法：get

参数：id、username

数据库.表：wa_test.users

响应类型：text/html

预期返回：显示用户名和注册时间、显示用户信息不存在

4、导出导入测试集合

（1）导出：为了备份和分享

export

（2）导入：为了复用和恢复

import

Day02：网络协议和Postman的应用

一、网络协议

1、什么是网络协议?

比如：HTTP、FTP、TCP、IP.....
网络 协议
Network Protocol
网络协议：网络通信的规则、规范、约定；
就是网络通信的“语言”

CEO 首席执行官
CFO 首席财务官
COO 首席运营官
CTO 首席技术官
CHO 首席人力官 HR 人力资源
公司管理需要分层

2、网络协议的分层原理

（1）分层的目的：分工
网络通信的过程比较复杂，需要通过不同层面分解，分工协作完成任务；
（2）分层的原理：各司其职、各尽所能，彼此影响

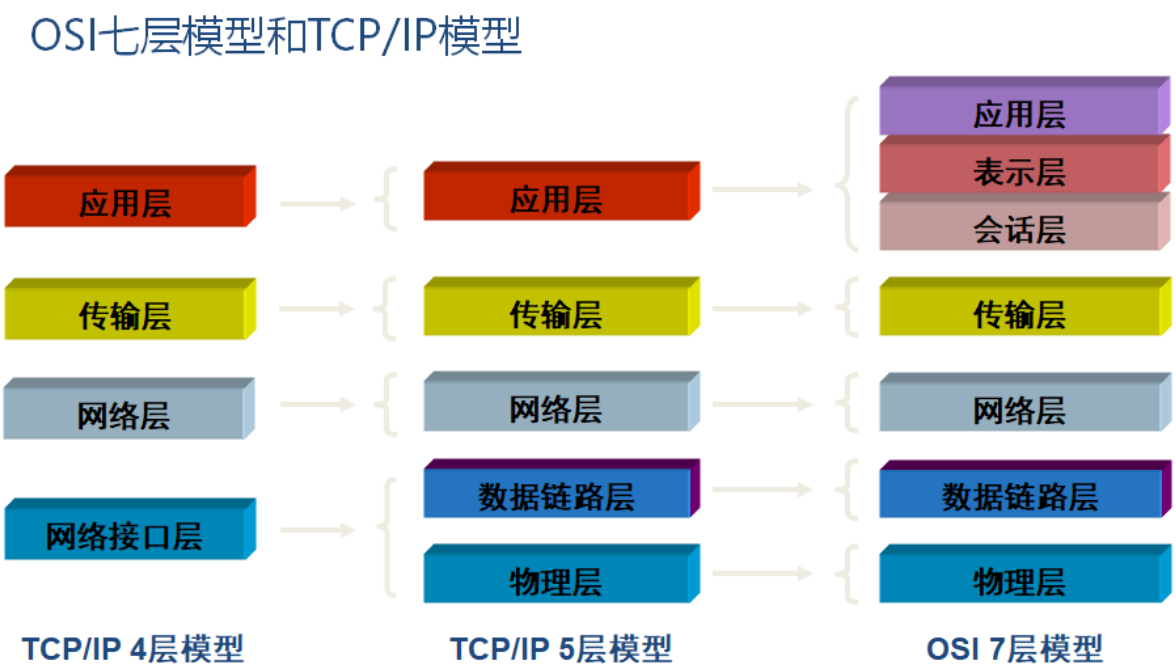
A ---------- B
秘书 ---------- 秘书
机要 ---------- 机要
电信 ---------- 电信

协议同层之间是对等的：规则、约定是一致的；
上下层是相互协作的：对信息的再包装、增加数据包；
下层对上层的数据进行包装，上层对下层的数据进行解包装；
不同层负责不同的功能。

3、OSI 7层网络模型 理论模型（了解）

（1）从上到下：
应用层、表示层、会话层、传输层、网络层、数据链路层、物理层
（2）记忆方法

从下到上：物数网传会表应



4、TCP/IP 4层模型 现实模型（熟悉）

也称为：TCP/IP协议簇 （cù 花团锦簇 簇拥）
以TCP、IP为代表的一群、一堆协议
从上到下：应用层、传输层、网络层、网络接口层（物理层）
规律：越高层，越面向用户；越底层，越靠近网络硬件。
名称 用途
（1）应用层：满足不同功能需求
网页浏览、接口访问（HTTP）
文件传输（FTP）
邮件收发（SMTP、POP3）
远程登录（SSH）
...

```
名称          端口        描述
```

<1> HTTP   80   超文本传输协议 日常请求、应答
<2> HTTPS   443   安全的HTTP   HTTPS = HTTP + SSL
<3> FTP   21   文件传输协议
<4> SSH   22   安全的shell 命令行方式远程访问服务器
<5> Telnet   23   远程登录和控制
<6> SMTP   25   简单邮件传输协议
<7> DNS   53   域名解析服务   域名->IP地址
                    域名解析服务器
          域名 --------> IP地址
      www.baidu.com   182.61.200.7
      www.juhe.cn   203.107.54.210
<8> POP3   110   邮局协议第3版本 比如：foxmail 默认采用POP3
<9> TFTP   69   简单文件传输协议 （不可靠的）

其它服务端口：默认端口 可以修改
  SqlServer：1433
  Oracle：1521
  MySQL：3306
  Tomcat：8080
  Fiddler：8888

访问接口时，在何处写端口号?
  主机名（域名或IP地址）之后，用冒号分隔填写，默认的可省略不写
http://www.baidu.com:80   :80可省略
https://www.baidu.com:443 :443可省略
http://apis.juhe.cn/xzpd/query:80 :80可省略
一般内部接口，访问具体某服务，需要提供，比如:8080，表示访问Tomcat服务器。

（2）传输层：保证传输可靠性
  TCP（可靠）和UDP（不可靠）
（3）网络层：对主机寻址
  IP协议：互联网协议 IP地址（主机名）
（4）物理层（网络接口层）：底层信号的处理

分析：访问接口和4层模型有何联系?
http://apis.juhe.cn/xzpd/query?men=白羊&women=金牛&key=xxxxx
协议名://域名或IP地址/应用名/功能名?请求参数

应用层：HTTP 应用名：xzpd 功能名：query 查询
传输层：TCP 可靠的
网络层：juhe服务器的主机的IP地址
物理层：底层通信过程

二、JSON核心语法

1、对比三种文本格式特点

（1）普通文本：优点：简单 缺点：无法表示复杂的数据信息
A,AB,ABC,CD,BC,...
1,12,123,34,23,...
name=Tom

age=23
salary=16000.6

（2）XML文本：可扩展标记语言 （了解）
优点：可以表示非常复杂的数据层次
缺点：生成、解析、查找数据不方便

```
<?xml version="1.0" encoding="utf-8"?>
<country>
    <name>中国</name>
    <province>
        <name>黑龙江</name>
        <cities>
            <city>哈尔滨</city>
            <city>大庆</city>
        </cities>
    </province>
    <province>
        <name>广东</name>
        <cities>
            <city>广州</city>
            <city>深圳</city>
            <city>珠海</city>
        </cities>
    </province>
    <province>
        <name>山东</name>
        <cities>
            <city>济南</city>
            <city>青岛</city>
        </cities>
    </province>
    <province>
        <name>新疆</name>
        <cities>
            <city>乌鲁木齐</city>
        </cities>
    </province>
</country>
```

（3）JSON文本： （熟练）
优点：既比普通文本复杂，能够存储比较复杂的数据信息；
　又比XML简单，更精简，便于生成和解析；
缺点：几乎没有，目前作为接口采用的数据文本交换格式。

```
{
    "name": "中国",
    "province": [{
        "name": "黑龙江",
        "cities": {
            "city": ["哈尔滨", "大庆"]
        }
    }, {
        "name": "广东",
        "cities": {
            "city": ["广州", "深圳", "珠海"]
```

```
        }
    }, {
        "name": "山东",
        "cities": {
            "city": ["济南", "青岛"]
        }
    }, {
        "name": "新疆",
        "cities": {
            "city": ["乌鲁木齐"]
        }
    }]
}
```

2、什么是JSON？ JavaScript对象表示法

（1）JSON是介于普通文本和XML之间的一种数据文本格式；
既比普通文本复杂，又比XML简单，便于生成、表达、解析。
（2）JSON相比XML更轻量化，提高网络传输效率；
（3）JSON是一种跨平台（操作系统）、跨语言（编程语言）的文本；
（4）JSON具有面向对象特性，主要存储对象的属性数据！
类：各种事物的概念、抽象　动物、车、水果　JSON
对象：某类事物的具体代表、个体、实例　某一个具体的JSON文本
{name: "Tom"}　不是！
{"name": "Tom"} 是！
{"age": 3}　　是！

（5）接口测试经常使用JSON、XML作为返回结果的文本格式；更多会使用JSON。

3、基本语法

（1）JSON对象使用{ }包围；
（2）属性数据以“名值对”表示；
"属性名": 属性值
（3）多对属性之间逗号, 分隔；
（4）数组使用[, ,]表示，内部元素使用逗号, 分隔；
（5）数据类型：
<1> 字符串："文本"
<2> 数字：整数、小数　直接写123　123.567
<3> 逻辑值：true　false　全小写
　　真　假
<4> 空值：null
<5> 对象：{ }
<6> 数组：[ ]
（6）JSON属性值可以是数组、对象，数组中的元素可以是JSON的任务类型，包括对象、数组，可以互相嵌套。

练习1：阅读接口文档，看懂JSON示例；

练习2：使用JSON语法表示一个员工的基本信息；

```
{
```

```
  "id": "10001",
  "name": "张无忌",
  "gender": true,
  "birthday": "1998-09-16",
  "salary": 15000.50,
  "job": "软件测试",
  "tel": "13811880099",
  "desc": null,
  "hobby": ["耍剑", "骑马", "摄影"],
  "skill": [
    {"name": "九阳神功", "grade": 9},
    {"name": "乾坤大挪移", "grade": 9},
    {"name": "太极拳", "grade": 9}
  ]
}
```

4、技巧：在线JSON校验、格式化工具

https://www.json.cn　　浏览器建议使用：Chrome、Edge
原理：通过内置JS脚本处理，存在浏览器兼容性问题。
注意：粘贴文本时要使用ctrl+v 才开触发校验程序！
　缩进

（1）在线校验
通过校验工具检查JSON的语法是否正确；
如果有错误，会加以提示，告知错误行号、具体问题细节；
（2）在线格式化
　<1> 格式化之前：　原始格式　原始的Raw （压缩后的）
　去除了分隔空白、换行符，只需要写一行文本即可；
　网络传输效率更高，节约流量；
　<2> 格式化之后：可读性好，便于阅读分析　美观的Pretty
（3）如果将JSON文本保存，文件扩展名一般为 .json
　　比如：emp1.json　保存一个员工信息
　　使用纯文本编辑器编辑，比如记事本等

三、post方法

1、发送x-www-form-urlencoded文本

案例01：post发送表单数据

　接口功能：判断登录是否成功

　接口地址：http://IP/apitest/text-login/

　请求方法：post

　请求类型：form表单

　参数：username、password

　数据库表：wa_test.users

　响应类型：text/html

　预期包含文本：用户**登录验证成功

练习02：post发送表单数据

接口功能：验证账号和密码是否正确

接口地址：http://服务器IP/apitest/login/

请求方式：post

请求参数类型：form表单

请求参数说明：

username：账号

password：密码

数据库表：wa_test.users

返回格式：json

Status（整型）	Result	Message
1000	Usercheck ok	登录验证成功

2、发送json字符串

使用raw类型下的json

案例03：signup接口

接口功能：接收用户名、密码、确认密码和姓名，实现注册用户的功能

接口地址：http://服务器IP/apitest/signup/

请求方式：post

请求参数类型：json字符串

请求参数：username、password、confirm、name

数据库表：wa_test.users、wa_test.info

返回格式：json

返回值形如：{"Status": 1000, "Result": "Success", "Message": "注册成功"}

案例04：send-json接口

接口功能：对json字符串的键进行排序

接口地址：http://接口服务器IP/apitest/send-json/

发送数据类型：json字符串

json字符串：{"name":"张三", "age":23, "isMarried":false, "child":null }

响应类型：json

响应结果：{"age":23, "child":null, "isMarried":false, "name":"张三" }

3、发送form-data

一般用于上传文件，有的接口要求提供参数，有的接口不要求提供参数

上传文件时，大多数接口会检查文件扩展名、文件大小、文件内容，自动对文件重命名

案例03：uponefile接口

接口功能：上传一个文件

接口地址：http://接口服务器IP/apitest/upload-file/uponefile/

请求方法：post

请求参数：file

文件名：尽量不用汉字

响应类型：text/html

预期：包含文本“文件上传成功”

案例04：upfiles接口

接口功能：上传多个文件

接口地址：http://接口服务器IP/apitest/upload-file/upfiles/

请求方法：post

请求参数：名称自定义

响应类型：text/html

预期：包含文本“上传成功”

4、binary

发送二进制数据

只能上传一个文件，而且没有参数

Day03：Postman高级应用

一、Postman布局和用法

1、工作区：Workspace 项目、模块

2、用例集：Collection 不同接口 管理多条用例

3、文件夹：Folder 子接口、协议、方法

4、请求：Request 每一条用例

（1）方法：get或post

（2）URL：按照接口文档要求填写 或 抓包获取（协议数据包）

（3）参数：

<1> 如果是Get方法：设置Params部分　全称：Parameter 参数

效果：URL后追加?参数名=参数值&参数名=参数值&...

QueryString 查询字符串

原理：在请求的头部header携带！

<2> 如果是Post方法：设置Body部分

原理：主要在请求的主体body携带！

分类：不同的方式，携带数据的结构、用途不同

none 不需要参数 body部分 0字节

form-data 表单数据（综合参数） 功能最强大、全面

文本Text（名值对）、文件File（名称和文件路径）

x-www-form-urlencoded 文本（名值对） 最常用

raw 原始的，适合提交代码 比如：json、xml、html、js...

binary 提交一个文件

（4）设置全局变量：Global Variable 作用范围：整个工作区

方法1：通过界面直接配置“小眼睛”名值对 变量名 变量值

方法2：通过脚本执行

在“预请求脚本” Pre-reqeust Script中填写：

```
pm.globals.set("变量名", "变量值");
```

tq_key xxxxx
xz_key xxxx
tq_url_1 http://apis.juhe.cn/simpleWeather/query

后续使用时：get请求
{{tq_url_1}}?city=北京&key={{tq_key}}
等价于：
http://apis.juhe.cn/simpleWeather/query?city=北京&key=xxxxx
好处：提高了用例的可复用性、易维护性；
如果需要修改内容，无需每个用例逐个修改，只需要修改全局变量定义的位置！