# 第一章 接口测试基础

## 1.1接口及接口测试

### 1.1.1什么是接口**：**

接口又称API(Application Programming Interface), 是为外部提供服务的一组程序。

接口

接口地址

请求

指定参数

规定格式

响应

特定格式的返回数据

身份验证

解析参数

业务处理

返回结果

### 1.1.2 接口的特点：

#### 1. 接口即服务，是上层UI界面操作的基础

#### 2. 接口提供外部访问途径

#### 3. 访问接口必须使用指定的参数和规定的格式

#### 4. 接口一般都具有一定的鉴权（身份验证）机制

### **1**.1.3 接口的分类：

常见的接口主要有HTTP接口和RPC两种类型：

#### 1. HTTP接口

使用HTTP/HTTPS协议传输数据或资源的接口，简单灵活。RESTful API（表征性状态转移）是一种HTTP接口的设计风格或规范

#### 2. RPC接口

RPC及远程方法调用，为解决跨服务器，跨语言调用问题（像调用本地函数/方法一样调用远程的（包括其他语言）的函数/方法）。常见的RPC解决方案有Web Service，Dubbo接口，Thrift，gRPC等

### **1**.1.4 什么是接口测试：

接口测试处于集成测试阶段，用于验证接口功能及数据交换是否满足预期，此外一般还要验证接口的容错性，安全性以及性能。

### 1.1.5 为什么要做接口测试：

在单元测试（开发人员自己完成） -> 集成测试（接口测试）-> 系统测试的流程中，在系统测试之前进行接口测试可以使测人员尽早的介入并发现大部分服务端的问题，有效的节省测试成本。

### 1.1.6 接口测试的一般流程：

服务端接口提测->接口功能测试（拿到需求文档和接口文档->梳理测试点或接口测试用例->测试接口）->接口性能测试（如必要）-> 客户端或UI提测 -> 系统测试

### 1.1.7 接口测试的一些注意点：

1. 首先应核对接口功能是否覆盖了需求中所有的功能点和场景
2. 由于请求接口必须使用指定的参数和规定的格式，接口文档是接口测试的必要条件，如果没有，需要跟响应的开发沟通了解接口请求方式，参数，格式，鉴权方式等必要信息
3. 接口测试用例应至少覆盖所有的响应结果
4. 注意接口的字段，返回值类型是否与前端约定的一致
5. 注意边界值的验证
6. 注意接口是否有重复提交的限制
7. 注意用户场景（多接口联动）的验证

### 1.1.8 接口测试需要掌握的知识：

1. 了解OSI网络模型，TCP/UDP协议，掌握HTTP/HTTPS协议，了解RPC, Web Service及RESTful，理解Cookie/Session，Token，授权协议，数字签名等常见接口鉴权策略
2. 掌握常用的接口测试工具如curl，Httpie，Postman, JMeter, AB等
3. 掌握基本的抓包工具如Chrome开发者工具,Fiddler，Charles等
4. 掌握一门编程语言Python或Java
5. 了解Nginx, Apache, Tomcat等服务器中间件
6. 掌握数据库基本查询命令，用于检查响应结果
7. 掌握基本的Linux日志查询和筛选命令

## 1.2 网络基础知识

### 1.2.1 IP、端口、域名及DNS

IP是为每一个网络和每一台计算机分配的一个逻辑地址，采用统一的格式。

查看IP命令：

* Windows：ipconfig
* Linux： ifconfig

端口为设备与外界通信的出入口，一个IP地址一般有65535个端口，端口分为：公认端口：0-1023，用于标准协议，注册端口：1024-49151，用于软件固定注册某个服务，如MySQL(3306),Tomcat(8080)等，动态端口：49152-65535：用于软件临时绑定一些服务



查看端口命令：

Windows: netstat -ano

Linux: netstat -ntlp

域名及DNS: 由于IP地址不容易记忆，为IP地址赋予了一个利于记忆的别名，称为域名  
如，百度的IP为： 61.135.169.125，对应的域名为 [www.baidu.com](https://links.jianshu.com/go?to=http%3A%2F%2Fwww.baidu.com)



### 1.2.2 OSI七层网络模型

上三层---应用层，控制软件方面

* 应用层：文件传输，电子邮件，文件服务，虚拟终端 TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet
* 表示层：数据格式化，代码转换，数据加密
* 会话层：解除或建立与别的接点的联系(会话)

下四层---数据流层，用来管理硬件

* 传输层：提供端对端的接口 TCP，UDP
* 网络层：为数据包选择路由 IP，ICMP，RIP，OSPF，BGP，IGMP
* 数据链路层 传输有地址的帧以及错误检测功能 SLIP，CSLIP，PPP，ARP，RARP，MTU
* 物理层 以二进制数据形式在物理媒体上传输数据 ISO2110，IEEE802，IEEE802.2

### 1.2.3 TCP及UDP协议

TCP：传输控制协议，特点：

1. TCP是面向连接的
2. 每条TCP连接只能用于两个断点，一对一
3. TCP提供可靠交付的服务：连接传输数据、无差错、不丢失、不重复、并且按序到达
4. TCP提供全双工通信
5. 面向字节流。TCP根据对方给出的窗口和当前网络拥塞的程度来决定一个报文应该包含多少个字节

UDP: 数据报文协议，特点：

1. 无链接
2. UDP使用尽最大努力交付，不保证可靠性
3. UDP是面向报文的，UDP对应用层交付下来的报文，既不合并，也不拆分，而是保留这些报文的边界。应用层交给UDP多长的报文，UDP就照样发送，即一次发送一个报文
4. UDP没有拥塞控制
5. UDP支持一对一、一对多、多对一和多对多的交互通信
6. UDP的首部开销小，只有8字节

## 1.3 HTTP协议

HTTP：超文本传输协议，是用于从WWW服务器传输超文本到本地浏览器的传输协议。  
HTTP协议是一种无状态协议，主要包含请求和相应两大部分：

### 1.3.1 请求(Request)

请求是我们发送给接口的数据对象，包含接口地址（URL），请求方法，参数，请求头（Headers), Cookies, 数据等



请求原始格式-GET（Raw格式：Fiddler抓包得到）

GET https://www.sojson.com/open/api/weather/json.shtml?city=%E5%8C%97%E4%BA%AC HTTP/1.1  
Host: www.sojson.com  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7  
Cookie: \_\_cfduid=dccd65c484a7657b468327b66023fefc41534934250; yjs\_id=59d1c42afa817b578b4b562d1f72651f; ctrl\_time=1

* 第1行： 请求方法 接口地址 HTTP协议版本
* 第2-N行：请求headers(如果有Cookie，最后一行为Cookie)
* 空一行
* 请求数据（POST等方法使用，此处为空）