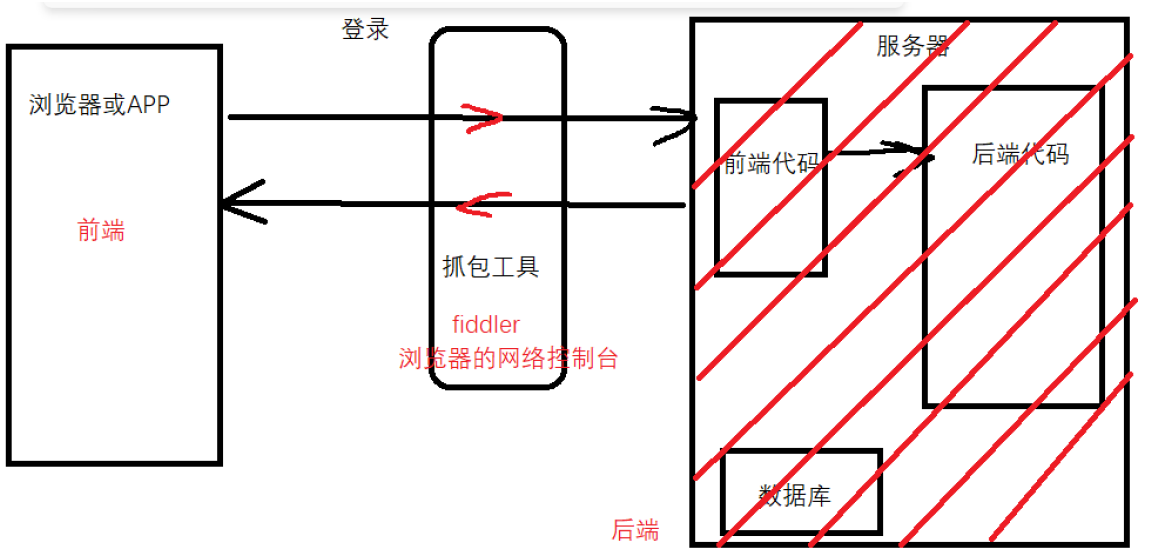
协议、抓包和用例设计

1、什么是接口测试：

- 接口：接口是服务器（后端）提供给客户端（前端），前端通过调用接口来向后端发送请求信息，后端再通过接口返回数据给前端，通过接口来实现前后端的数据交互。常见 json xml 两种数据格式
- 接口测试的原理：模拟客户端向后端发送请求，服务器接收到了请求会做相应的处理并且将处理信息响应给前端



2、接口测试的目的

- **测试类型**：
  - 功能测试：一般需要等到前端和后端联调完成后，才能进行的测试
  - 单元测试：测试代码或代码的流程
  - 接口测试：不需要等到前端的UI界面开发出来，或不需要等到前后端联调完成。只需要后端在将接口开发出来后就可以进行测试。
- **能更早的发现问题**

在进行功能测试之前进入接口测试，能提前发现后端的bug，减少后期的成本

- **缩短产品的测试周期**

  更早的发现问题有助于节省后期修复的时间
- **发现底层的bug**

  有些bug通过功能测试出现会比较复杂，通过工具直接模拟前端发送请求可以发现更直接的问题

  （检验绕过前端向后端直接发送请求的漏洞）

3、接口的分类

- **系统内部的接口：**模块之间的相互调用（下单后-->商家同步订单-->个人也同步订单）
- **系统外部的接口：**
  - 软件接口：
    - 服务器接口：通过调用服务器提供的接口来进行数据之间的交互的
    - 外部接口：第三方支付，三方数据接口等
  - 硬件接口：USB , type-c

4、网络协议

OSI 七层网络模型

- **应用层**
  - 应用层是最靠近用户的一层，负责对软件提供接口让程序能使用网络服务。应用层概统的服务包括文件传输，文件管理以及电子邮件信息处理等。通俗的讲：应用层提供接口。
  - 常见应用层协议：http，ftp ，tftp，smtp(邮局协议)，https等
  - 举例应用层一些工作：ftp(文件传输协议，传输文件)，web (http协议)
- **表示层**
  - 管理数据的解密与加密，还对图片和文件格式信息进行解码和编码
  - .jpg .mp4 解压/压缩 ASCII 加密格式
- **会话层**
  - 建立，管理，终止会话。也是负责在网络中的两个节点之间建立和维持通讯。
- **传输层**
  - 定义传输数据的协议端口号（mysql默认端口：3306），流量控制和差错校验等。
    - 流量控制：基于接收方可以接收数据的快慢程度来决定适当的发送速度
    - 差错校验：将数据进行分段传输，到达对方传输层后可以进行数据重组
  - 传输层的协议：
    - TCP （可靠传输）
    - UDP （不可靠传输）
- **网络层**
  - 进行逻辑地址寻址。实现不同网络之间的路径选择。决定如何将数据从发送方路由到接收方
  - 协议：ICMP IGMP IP（ipv4 ipv6）
  - 路由器
- **数据链路层**
  - 它控制网络层与物理成之间的通讯，将从网络层接收到的数据分割成特定的可以被物理传输层传输的帧
- **物理层**
  - 用于传输比特流，比特流只有0和1。它物理接口的强弱电压
  - 光纤，双绞线等

常见协议

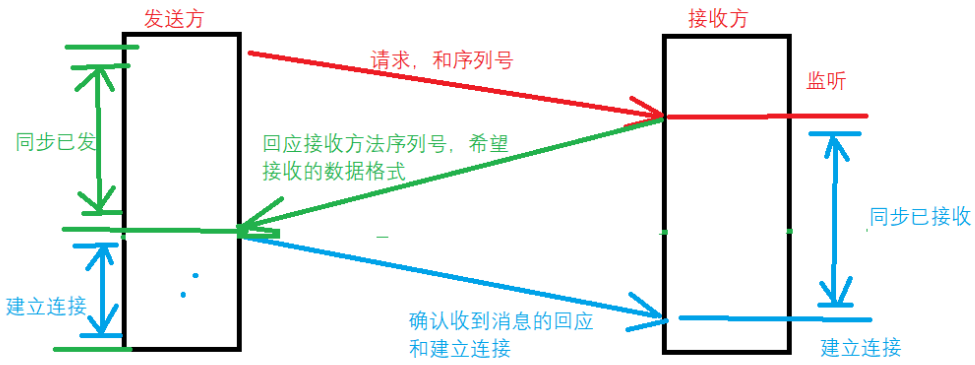
- **http协议：** 超文本传输协议，默认端口是80，浏览网站使用的协议
- **https协议：**超文本传输协议，http+ssl加密版本，默认端口号是443
- **ftp协议：**文件传输协议，端口：20,21,990，
- **pop3:**邮局协议，发送和接受邮件，端口：110。也有pop3 +ssl的加密版本
- **SMTP：**简化版邮局协议
- **telnet:** 默认23 远程终端协议

TCP/IP 四层网络模型

- **应用层**
  - 定义传输的数据和按照接收的数据格式解读数据
  - 常见的几种协议 SMTP HTTP FTP
- **传输层**
  - tcp / udp 协议处理数据，
  - 进行三次握手，数据分割和重组处理
  - 将应用层传过来的数据进行字节处理，每一段字节添加tcp头部再传入网络层中
- **网络层**
  - 对数进行ip协议加工，路由进行分配，定义网络地址
- **网络接口（链路层）**
  - 将数据转化成'数据帧'，通过物理介质传输数据

tcp协议的特点

- **面向连接协议**：在进行传输的时候，需要先建立连接（例如：我们打电话需要先拨通电话）
  - **三次握手**
    - ①发送方先发送一个请求连接的报文，并且附带上了本机的序列号。
    - ②接收方到报文之后，会给发送方进行回应，并且带上本机的序列号和希望接收的数据格式。
    - ③发送方收到回应后，再给接收方发送收到的回应确认消息，此时接收方收到这条确认消息后，连接建立。



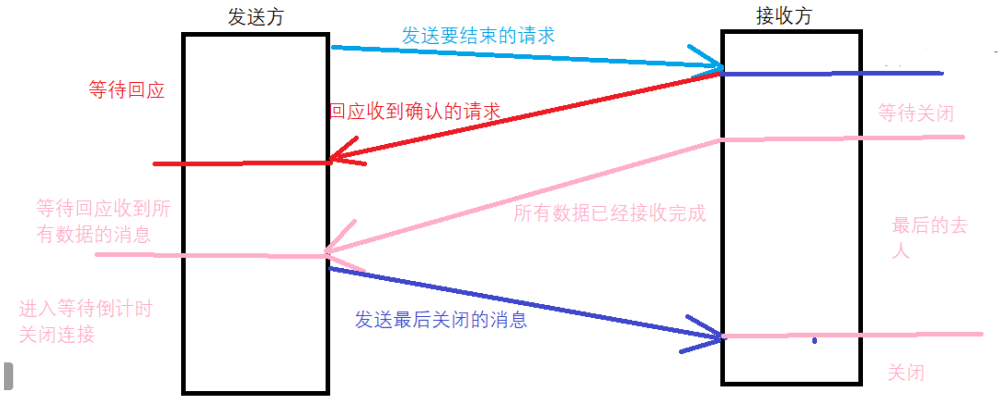
  - **四次挥手**
    - 由发送方发起结束连接的请求（发送后会进入等待确认状态）
    - 接收方收到结束连接的请求后，会给发送方回应一个确认消息。

此时发送方收到消息后，进入等待状态

等待接收方发过来的第二次传输完成确认的消息

此时接收方还是可以给发送方发送消息的

- 此时接收方将数据接收完成后，再给接收方发送传输完成的消息（进入关闭连接的确认状态）
- 发送方收到了传输完成的消息，再给接收方发送最后一个关闭的消息，连接关闭。



- **点对点通讯：**
  - 发送端和接收端就是两个点，两个点之间的相互通讯就是点对点
- **可靠传输**：
  - 在传输数据的时候，接收方会返回一个受到数据的信号，发送方根据受到的确认信号确定这部分数据是否被丢失已决定是否要重发这个部分数据。
- **全双工**：两个计算机在连接的时候相互给对方发送数据
- **字节流**：
  - 对于应用层传输过来的协议，tcp协议会对这些数据当做字节来处理
- **拥塞控制：**
  - 通过拥塞控制算法和慢启动阀值算法来控制网络拥塞。

udp协议：

用户数据报协议

- **面向数据报文传输协议**：

  非面向连接，是面向对象的

  udp协议不会对应用层传输过来的数据做任何处理，在发送数据时是直接放到udp协议的数据里面
- **无连接协议：**
  - 给目标发送数据的时候，不需要建立连接，直接发送过去
- **传输不可靠：**
  - 因为没有事先建立连接，所以在传输的时候没有监控接收方是否收到数据
- **没有拥塞控制**：udp不管网络是否畅通，都会尽快的吧数据传输出去

5、URl

统一资源定位符

举例：https://www.baidu.com/baidu?tn=monline_3_dg&ie=utf-8&wd=OSI

由四部分组成：

- <协议> : //<主机>：<端口> / <路径> ？ 参数
  - 协议：http , https, ftp, file（本地文件分享到网上）
  - 主机：文件所在的服务器的域名或IP地址
  - 端口：端口号
  - 路径：服务器中文件的路径，文件的位置
  - ？ ： 路径和参数的分隔符
  - & ： 参数之间的连接符号

域名

域名是对IP地址的包装（因为IP地址不好记忆），在访问域名的时候，域名解析器会自动解析成ip地址访问

域名的种类

- **顶级域名**:
  - 国家级域名，如：.cn（中国） .tw （台湾） .us（美国）
- **二级域名**
  - 顶级域名下的二级域名
  - 国内常见的域名：
    - .com .cn （公司或企业）
    - .net.cn ()
    - .gov.cn （政府机构）
    - .edu.cn （学校单位）

6、http协议

Hyper Text Transfer Protocol 超文本传输协议

概念：http协议是一个基于底层协议：tcp/ip通信协议，是超文本超文本传输协议 可靠的

http/https的特点

- 使用B/S架构
- 超文本：超过文字，包含图片，视屏，音频等
- 无连接：每次传输请求一次，响应数据一次，则断开连接，以节省资源
  - keep-Alive ： 是客户端连接到服务器持续有效
- 无状态：指协议对于事务处理没有记忆功能，服务器不知道客户端是什么状态
- 针对无状态的访问解决方案：
  - ①使用cookie 来记录用户信息，每次用户登录之后，服务器会返回一个cookie值，然后这个cookie会被浏览器保存在本地，再下一次请求的时候，带上这个cookie，这样子服务器就知道是哪位用户进行了访问。

- ②使用 Session ， 每次用户登录服务器会生成一个Session ， 会返回到浏览器中，下次请求需要带上Session来进行访问。Session是存储在服务器中。

cookis与Session的区别

cookis	Session
存放在客户端	放在服务器上
单个cookies的保存数据<=4kb	没有上线
只能保存ascii码数据	没有编码限制
cookis对客户端是可见的	更安全
可以设置长期有效	会定期清理，补可长期
不会占用服务器资源	每个用户都会产生一个Session会消耗服务器资源

http和https的区别

http	https
明文传输，不安全	有SSL加密，加密传输
默认端口：80	默认：443
无状态	SSL需要进行握手
	需要申请CA整数

http的请求方式

- get : 用于获取资源
- post : 提交/修改表单
- update : 更新服务器资源
- delete : 删除
- head ， option , put , trace 等

get和post的区别

get	post
从服务器中获取数据	向服务器提交数据
传递数据量较小	不受限制
请求的数据（参数）写在URL中	在请求体（请求正文）中进行传参
只能接收ascii码	不受限制
一般只发送一次请求	一般发送两次，第一次：请求头相关，第二次：发送正文

http协议的请求和响应

- **请求**:
  - **请求头**:
    - Accept-Encoding： 接收的压缩格式
    - Accept-Language： zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
    - Connection： keep-alive 是客户端连接到服务器持续有效
    - Host : 请求的主机
    - Cookie： 身份信息
    - Referer： 重定向
    - User-Agent： 用户代理，不同的浏览器方式去访问服务器
    - Cach-Control : 缓存控制
    - Accept- Charset : 接收到字符集
  - **请求正文**
    - form data : 表单格式
      - 变量名=值
    - json数据格式，类似于python的字段格式
    - xml : <name='变量名' ，value=值>
    - 文件格式：文件的参数以表格的形式存在，存的是文件的路径
- **响应**:
  - **响应状态码**

    状态码是访问状态的解释
    - 200 OK
    - 1XX : 请求没有完成，需要继续请求
    - 2XX： 请求完成
    - 3XX： 请求重定向，重定向的URL会在响应头中获取

        - 4XX： 客户端问题居多，如：路径写错，或服务器文件被删除
        - 5XX： 服务器的问题居多
        - 6XX： 服务器响应问题

        - 301： 请求永久重定向
        - 302 : 请求临时重定向
        - 304： 请求被被重定向到客户端
        - 400： 客户端请求存在语法错误
        - 401： 客户端请求没有授权
        - 403： 客户端的请求被拒绝，一般是权限被拒绝
        - 404： 客户端请求的URL在服务器中不存在
        - 500： 服务器端永久错误

    - **响应头**

Connection	**Keep-Alive**
Content-Length	19088
Content-Type	application/javascript
Date	Fri, 08 Apr 2022 02:28:48 GMT
ETag	"4a90-52fde67300a00"
Keep-Alive	timeout=5, max=99
Last-Modified	Thu, 07 Apr 2016 05:32:24 GMT
Server	Apache/2.4.6 (CentOS) PHP/5.4.16

    - **响应正文**

响应的数据格式：html， 图片，json格式数据，js脚本 , xml

7、抓包

- 常见的抓包工具：
    - 浏览器自带的抓包工具---F12 或 fn + F12 或浏览器空白处点击右键选择“检查”
    - fiddler
    - charles --青花瓷
    - wirechark --功能更齐全
- 学习抓包工具的目的：
    - 通过抓包获取信息，通过请求和响应的数据来判断前端或后端的问题
    - 弱网测试
    - 安全性测试，断点篡改数据

8、fiddler的使用

- 抓取https包
- 向上断点，向下断点
    - 断点篡改数据的使用场景：
        - 在发送请求的时候，对请求进行拦截，然后将数据修改，在放行。验证服务器对篡改数据的判断
- 请求重发：
    - 将同一个请求重发多次，可以达到快速制造测试数据的目的