

接口测试

1.接口

主要是各模块或各系统间交互并相互作用的部分，这里说的接口是广义的，客户端与后台服务间的协议，插件间通信的接口，模块间的接口，再小到一个类提供的方法，都可理解为接口

2.接口种类

常见分类：

get , post, delete, put

系统对外接口：从别的网站或服务器获取数据信息，别人肯定不会把数据库分享给你，只能提供一个他们写好的方法来获取数据，你引用他提供的接口就能使用他写好的方法，从而达到数据共享的目的

程序内部接口：方法与方法之间，模块与模块之间的交互，程序内部抛出的接口，比如bbs系统，有登录模块，发帖模块等等，要发帖就必须先登录，那么这两个模块就得有交互，那么它就会抛出一个接口，供内部系统进行调用

接口的分类： webservice接口 http api接口

webService接口：走soap协议通过http传输，请求报文和返回报文都是xml格式的，我们在测试的时候都是通过工具才能进行调用，测试

http api接口：走http协议，通过路径来区分调用的方法，请求报文都是key-value形式的，返回报文一般都是json串，有get,post等方法

json是一种通用的数据类型，所有的语言都认识它，它的本质是字符串，可以转化为其他语言的数据类型

各接口之间区别

get的提交方式是明文提交，把提交的参数跟在url后面发送给服务器，所以不安全，而且get提交的参数是有字符限制且可以被当做书签保存

post的提交是ssl加密，参数是放在表单里，所以不会存在字符限制，而且因为参数是放在表单里，不容易被看到，所以比get更安全

3.接口测试目的

因为前后端进度不同，所以针对最先出来的接口以及需要调用其他公司的一些接口进行接口测试以及验证数据，从安全层面来说，只依赖前端验证进行限制已经不能满足系统的安全要求，需要后端同样进行控制。

4.接口测试流程

需求讨论，需求评审，场景设计，编写用例，准备数据，执行测试

4.1如何进行接口测试

通过工具模拟客户端向服务器端发送请求并接受服务器返回的数据来对接口的功能，逻辑业务，异常，安全进行测试

4.1.1功能测试

测试这个接口的功能是否实现，并且测试这个接口是否按照接口文档来进行开发的（比如说接口文档规定了一些关键字，而开发有时候把关键字改成了其他的关键字，因为在整个项目周期，并不只有一个开发而是有多个，所以可能因为在开发过程中因为关键字不一样导致某些开发的功能异常，还有自动化脚本也会发生异常）

4.1.2逻辑业务

主要指一些逻辑业务依赖关系（比如支付宝提交订单的时候要保证你在登录的情况下，如果没有登录就提交成功了，这就是异常，可以修改请求的cookie来测试）

4.1.3异常测试

参数异常：关键字异常（应用其他的关键字替换进行测试），参数为空，参数多少（通过添加参数增添个数），参数错误

数据异常：关键字数据（填入的数据用其他的数据语言的数据替用），数据长度，数据为空，数据错误

5.接口测试用例设计思路

目的：测试接口的正确性和稳定性

原理：模拟客户端向服务器发送请求报文，服务器接受请求报文后对相应的报文做处理，并向客户端返回应答，客户端接收应答的过程

重点：检查数据的交换，传递和控制管理过程，还包括处理的次数

核心：持续集成是接口测试的核心

用例设计重点：通常情况下主要测试最外层的两类接口：数据进入系统接口（调用外部系统的参数为本系统使用）和数据流出系统接口（验证系统处理后的数据是否正常）

ps:设计用例时还需要注意外部接口提供给使用这些接口的外部用户什么功能，外部用户真正需要什么功能

5.1输入（可按照参数类型设计）

输入参数主要从以下几个方面设计：

a.必填项校验

接口文档中是否有必填的说明，参考接口文档即可

b.参数长度校验

参考接口文档即可

c.参数值的有效性校验

如身份证的校验：设计的数据虽然符合身份证的规则，但是并不是真实有效的身份证号，这种情况就要看身份证号的校验规则是什么样的，一般都是用的现成的身份证校验器。所以参数有效性的校验就需要结合实际业务场景，判断哪些数据是真实有效的数据，一定要确保所有真实有效的数据是可以验证通过的

d.参数组合校验

不同的参数组合可能会存在不同的业务场景

e.如果参数是枚举值，一定要各种枚举值都要测试，因为可能不同的枚举走不同的业务流程

f.参数值的默认值的校验

参考接口文档

g.某些参数具有特定的生成规则，要单独针对生成规则设计测试用例，一定要保证真实有效的数据是可以通过验证的

5.2接口逻辑（可按照逻辑进行设计）

设计方法：分支覆盖->路径覆盖->场景覆盖，同时也要结合实际业务场景，根本不发生的场景就是无效的测试用例

a.第一步先把业务流程图画出来；

b.依据路程图中的分支分别设计，不同分支不同场景，这里就要把异常场景考虑。如接口超时，接口异常，接口请求成功或失败，成功后怎么处理，失败后流程是否继续进行，失败后数据怎么处理。

以打款接口为例：

打款结果有打款成功或打款失败，成功后怎么处理，需要回写打款成功状态，失败后怎么处理，也需要回写失败状态，失败后的数据可以操作退回，也可以操作重新出款等等；

c.测试逻辑设计完成后要想一想不同的业务场景怎么去测试，需要哪些人员协助

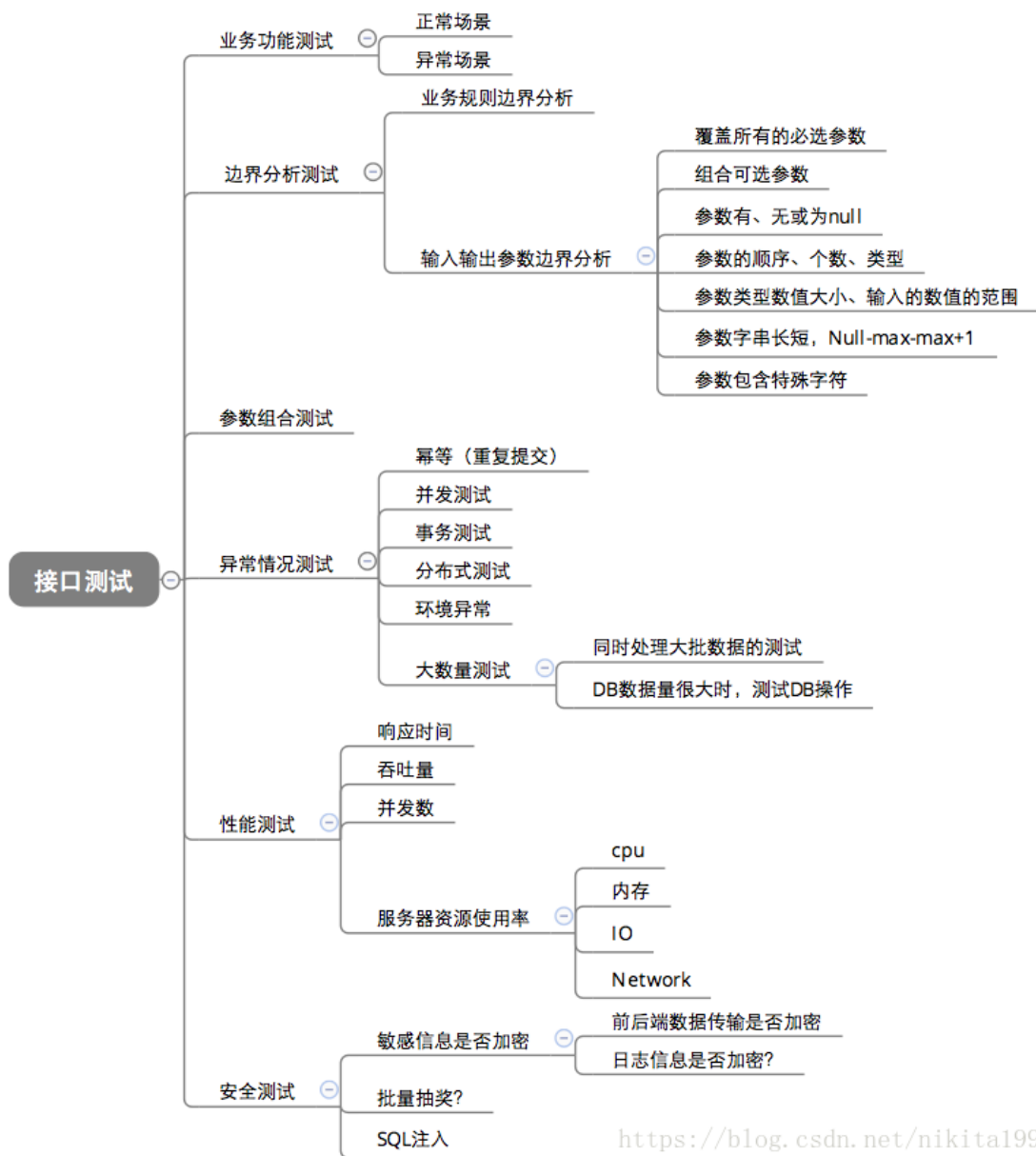
如：接口超时怎么去测试，请求重复怎么去测试，请求并发怎么去测试

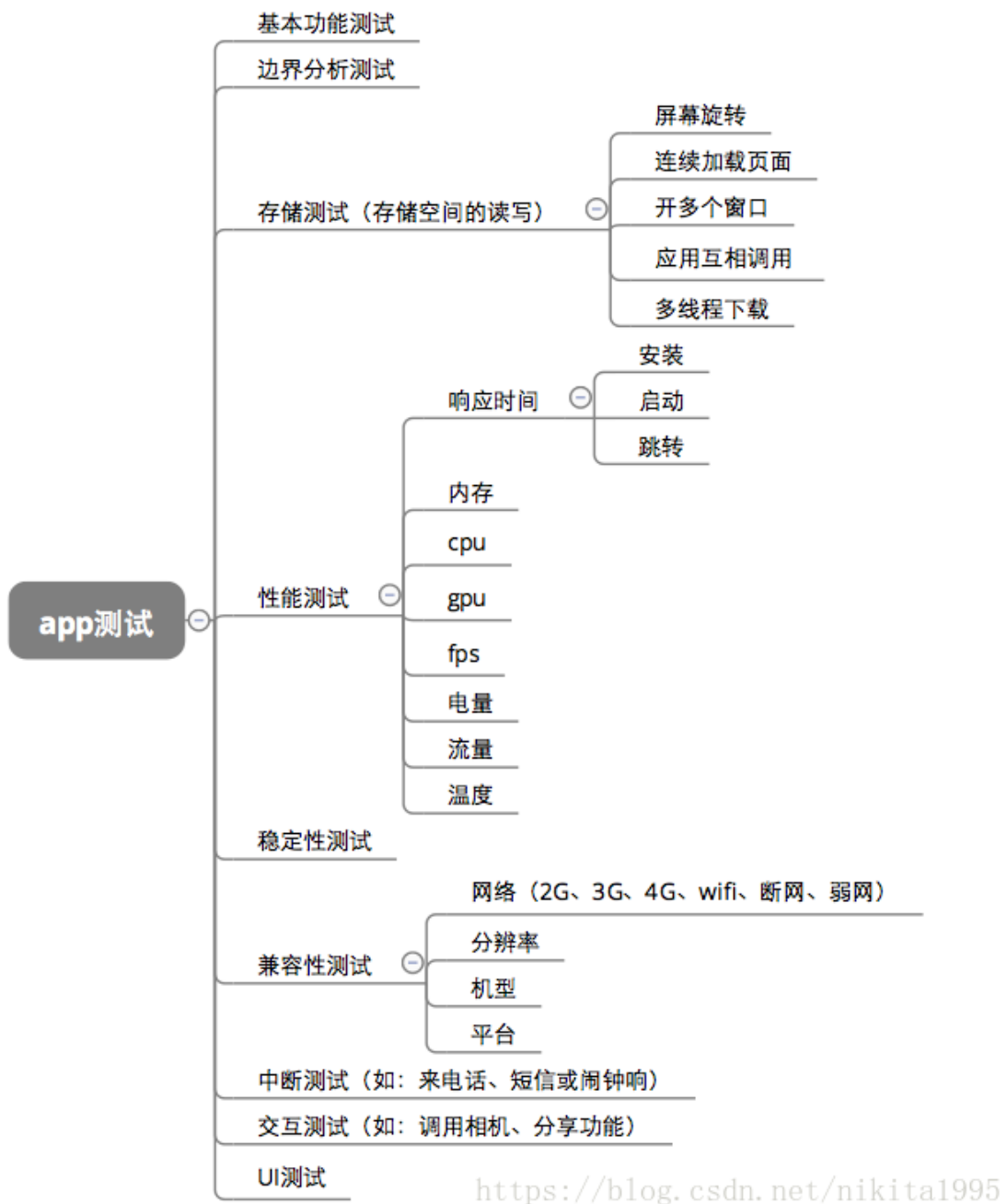
5.3输出（可根据结果进行分析设计）

输出结果：正常输出和异常输出，常用的方法有错误推断法，列举出程序中可能出现的错误和异常，根据他们选择测试用例

5.4 扩展

- 上述都完成后，根据实际情况去掉冗余的测试用例（即实际情况中不存在的流程和输入数据）；
- 如果业务流程涉及到状态转换，要单独设计用户方法：状态转换图；
- 涉及到多个不同金额或者手续费的计算，可能还会用到正交实验法去设计用例；
- 用例设计中还应当包含异常流程中产生的异常数据的处理流程；一通常所说的补偿机制，这块流程能大大的减轻人工运营的工作量，当然，这需要在做系统设计的时候就需要把这部分考虑进去。





接口测试需要掌握的知识：

了解系统及内部各个组件之间的业务逻辑交互（通过需求文档，流程图，思维导图，沟通等）

了解接口的I/O(input/output:输入/输出)

了解协议的基本内容，包括：通信原理，三次握手，常用的协议类型，报文构成，数据传输方式，常见的状态码，URL构成

常用的接口测试工具，比如：jmeter,loadrunner,postman,soapUI

数据库基础操作命令（检查数据入库，提取测试数据等）（菜鸟教程）

常见的字符类型，比如：char,varchar,text,int,float,datetime,string等

get请求，post请求区别

- get使用url或cookie传参，而post将数据放在body中
- get的url会有长度上的限制，post的数据则可以非常大
- post比get安全，因为数据在地址栏上不可见
- 一般get请求用来获取数据，post请求用来发送数据

http状态码

200 2开头的表示这个请求发送成功了，200这个最常见，表示请求发送成功了，服务器也返回了

300 3开头的代表重定向，最常见的是302，表示这个请求重定向到别的地方去了

400 400代表客户端发送的请求有语法错误，401代表访问的页面没有授权，403代表没有访问权限，404代表没有这个页面

500 5开头的代表服务器有异常，500代表服务器内部异常，504代表服务器端超时，没返回结果

cookie与session的区别

cookie数据存放在客户的浏览器上，session数据放在服务器上

cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，考虑到安全应该使用session

session会在一定时间内保存在服务器上，当访问增多，会比较占用服务器的性能，考虑到减轻服务器性能方面，应当使用cookie

单个cookie保存的数据不能超过4k，很多浏览器都限制一个站点最多保存20个cookie

所以一般：将登录等重要信息存放为session，其他信息需要保留，可以放在cookie中