**什么是接口测试**

测试人员通常所说的“接口测试”是针对系统各组件之间接口的一种测试，它属于功能测试。接口能测出普通界面操作难以发现的问题。如，我们都知道系统是由前端后端组成，一些数据在前端做了校验，后端同样也需要校验才能保证安全，界面操作显然只能检查到前端校验这一层，只有直接面对前后端之间的该接口才能检验出后端是否也做了校验。

**接口测试的必要性**

   可以发现很多页面操作发现不了的问题

   检查系统的异常处理能力

   检查系统的安全性、稳定性

   前端随便变，接口测好了，后端不用变

**接口测试的流程**

   需求评审，熟悉业务和需求

   开发提供接口文档

   编写接口测试用例

   用例评审

   提测后开始测试

   提交测试报告

**接口文档**是接口测试的参照，至少包括：

1、接口说明

2、调用url

3、请求方法（get\post ……）

4、请求参数、参数类型、请求参数说明

5、返回参数说明

**接口测试用例设计**

通过性验证：首先保证接口好用，按文档正常传入，查看是否可以返回正确的结果。

参数组合： 按接口文档中对参数的要求进行有目的的组合，比如必填未填是否通过，标志类参数值的切换是否能对应正确的功能等。（这部分很关键）

接口安全：

      1、绕过正常值验证。

      2、绕过身份授权验证。

      3、参数是否加密，加密规则是否容易破解。

      4、密码安全规则，密码的复杂程度校验。

异常验证：不按照接口文档上的要求输入参数，来验证接口对异常情况的反应。

**接口测试用例模板**（可根据项目实际情况设计增减）

1、项目            测试针对哪个项目

2、模块            哪个功能模块

3、用例id

4、接口名称

5、用例标题      测试用途概括

6、请求方式      GET/POST

7、请求url        URL地址

8、请求参数

9、前置条件       执行当前请求依赖的条件，不满足就不能正确执行

10、结果验证     预期结果

11、请求报文     可以不写

12、返回报文 　一定要写，这里应该是你请求返回的真实结果

13、测试结果    通过/失败

14、测试人员

**测试http接口**

请求常见有Get请求和Post请求。Get请求通常用来接收数据，Post请求通常用来发送数据；测Get请求可用浏览器完成，参数都可以写在URL里面，测Post请求需要借助工具如Postman，因为客户端需要提供给服务器的信息较多，你要写body传输大量数据。

接口调用有两种传参方式：key-value形式，Json串传参形式。

key-value形式可以把参数拼接在url的后面由?相连，多个参数之间用&相连，如url?parameter1=key1&parameter2=key2…

Json串传参不能把参数直接连在url中，需要写在请求的body里面，可借助工具Postman，打开请求的body写入Json格式参数（由花括号括起来的‘键：值’对）如

{

 “count”: 1,

 “start”: 0,

 “total”: 1

}

请求发出后，http会返回一个状态码表示请求是否成功，状态码有三位，其中开头一位确定了状态类型：

   2xx： 表示请求发送成功，常见200。

   3xx： 代表重定向，要完成请求必须进行更进一步的操作，或把请求重定向到别的地方了，最常见的是302。

   4xx： 客户端错误，请求有语法错误或请求无法实现。400代表客户端发送的请求有语法错误，不能被服务器所理解；401代表访问的页面没有授权；403服务器收到请求，但是拒绝提供服务，比如没有权限访问这个页面；404请求的资源不存在，比如输入错的URL没有这个页面。

   5xx： 代表服务器有异常，500代表服务器内部异常；503服务器当前不能处理客户端的请求，一段时间后可能恢复正常；504代表服务器端超时，没返回结果。

**测试WebSevice接口**

不需要像测http接口那样拼报文，直接把wsdl地址或wsdl文件（这两个都由开发人员提供）填写或导入到工具SoapUI里面，工具里可显示所有相关接口或报文，直接填入参数发送请求参照接口文档查看结果即可。

**Cookie 和 Session**

Cookie是存在于本地的一个键值对，Session是存在于服务器端的一个键值对，通常保存在数据库或缓存里。Cookie和Session在第一次发送某个请求时成对生成，两端都会记录下生成的时间，超出既定的时限后便会自动删除。当请求在时限内再次发出后，Cookie和Session两者会相互比对，匹配上了便执行某些操作，匹配不上则不允许执行某些操作，以此实现快速处理，它们并不是孤立作用的。