## 接口测试流程

我们在拿到接口文档后，对文档中的每个接口进行分析，弄清楚每个接口的功能和业务逻辑，分析接口的测试场景。然后进行接口测试的用例设计，运用等价法、边界值等方法实现每个接口的测试用例。借助接口调试工具添加测试数据，进行断言，生成脚本，用代码执行每个测试用例。在这过程中将发现的Bug提交到缺陷管理系统，例如禅道等，由开发修复BUG后再次回归验证。最终输出接口测试报告，对接口测试情况进行评估和风险分析。

## 接口测试工具使用

我们在接口测试的过程中，会使用到Postman或者Apifox这样的接口调试工具。在Postman中，一个项目创建一个工作空间，在工作空间中的集合存放各个模块的接口。每个接口我们都可以根据接口文档设置请求方式，例如get，post，put等。还有请求URL，以及为get请求方式添加查询参数，或者为post请求方式设置请求体。

在发送请求后，我们能看到响应报文，包括响应状态码，例如2开头的表示请求成功，4开头的表示客户端错误，5开头的表示服务器错误。

在项目中，也会进行环境变量的创建，例如在开发环境和生产环境中的ip或域名不同，需要创建两个环境dev和pre，分别创建base\_url变量存储不同的ip地址。使用双花括号加变量名读取。做到快速的环境切换。一些共用的值我们都存储在全局变量中。

在接口测试的过程中，接口可能依赖另一个接口返回的数据，我们会在script中的Pre-request中编写脚本，使用pm对象的sendRequest方法发送请求，将获取到的数据保存到环境变量中供接口临时使用。对于断言我们会在Post-response中使用pm对象的test方法，判断接口是否满足文档的要求，例如响应状态码断言，响应时间断言，响应体中的内容是否具有某字段断言，包含某字符串断言，值等于断言等。

如果一个接口依赖的另一个接口还未实现，我们使用Mock模拟一些数据，在不依赖实际接口实现的情况下继续进行测试。

然后在集合中，使用runner工具，批量执行接口并查看测试结果。我们也会将集合中的接口导出，使用cvs文件中的数据作为测试数据，然后用newman工具批量执行，导出html文件查看结果。

还有Apifox，它与Postman相似，Postman的功能它都支持，它更符合国内开发者的使用习惯，还提供全中文界面和本地化服务支持。

在调试接口的过程中，UI或接口出现错误的情况下，我们使用抓包工具，例如fiddler，在过滤页签下对所有的请求进行过滤，找到有问题的请求，查看协议，请求头，请求方式，请求地址和响应报文等信息。有时我们也会使用composer发送请求来协助调试。在AutoResponser页签下，我们还能够设定规则对请求拦截，并返回指定的内容。

我们有时会使用fiddler对移动端进行抓包，使移动端和fiddler所在的电脑处于同一网络，并设置fiddler允许远程连接，在移动端设置代理，填写fiddler电脑的ip地址和默认的8888端口，然后通过ip地址加端口在浏览器访问，下载证书并安装。然后正常抓包操作。