

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，运用等价划分法、边界值等方法实现每个接口的测试用例。借助接口调试工具测试接口，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。

我们在工作中一般使用 Postman 接口调试工具。在工作空间中的集合存放各模块的接口。每个接口我们都需要根据文档选择请求方式，例如 get, post, put 等。还有请求 URL，以及为添加查询参数，或者设置请求体。

例如测试一个登录接口，把开发和测试环境的 ip 存入环境变量，使用双花括号读取，加上接口地址，请求方式设置为 Post 方式，运用等价类划分法和边界值法，设计多个用例，用户名为空，密码为空，长度超过限制等。

在接口测试的过程中，获取用户信息接口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，将登录接口获取到的 token 保存到环境变量中。然后在 Post-response 中使用 pm 对象的 test 方法，第一个参数为断言的描述，第二个参数为一个函数，在函数体中使用 pm.expect 断言。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，用 newman 工具批量执行，-n 可以指定迭代次数，-d 去指定测试数据 cvs 文件，加上 --reporters html 参数导出可视化的报告。

在调试接口的过程中，UI 或接口出现问题时，我们可以使用 fiddler 抓包工具分析，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 AutoResponder 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。