为什么做接口测试

1. 更容易实现持续集成
2. 自动化测试落地性价比更高，比UI稳定
3. 大型系统更多更复杂，系统间模块越来越多
4. BUG更容易定位
5. 降低研发成本，提高效率

需求阶段：项目立项、产品设计、需求文档

研发阶段：UI设计、前端开发、后端开发、测试设计、测试开发

测试阶段：环境搭建、多项测试执行、BUG修复、测试报告

项目上线：线上回归测试、上线报告、添加监控

1. 接口测试范围
2. 功能测试：等价类、边界值、错误推断法、因果图、场景法、功能图法、判定表、 正交实验法
3. 异常测试：数据异常（null、空字符串、数据类型）

环境异常（负载均衡架构、冷热备份）

1. 性能测试：负载测试、压力测试、并发测试、稳定性测试
2. 自动化接口测试范围

功能测试+数据异常测试

Git的使用：

1. 配置sshkey
2. cd ~/.ssh（用户目录下的.ssh文件夹）
3. 生成key：ssh-keygen -t rsa -C 邮箱地址
4. 复制id\_rsa.pub的公钥内容到github网站中
5. 配置多个sshkey
6. cd ~/.ssh（用户目录下的.ssh文件夹）
7. 编辑config文件（vim config）
8. Config中添加Host github.com

HostName github.com

User 用户名

IdentityFile id\_rsa文件的绝对路径

git clone URL

git status

git add 文件名

git commit -m “描述”

git push 推送到网站上

git pull 拉取文件

创建本地分支：git checkout -b 文件名

git branch 查看本地分支

git branch -a 查看本地和远程分支

git chechout 分支名

git branch -d 分支名 删除本地分支

git branch -r -d 远程分支名

git push origin :远程分支名

git merge demo 把demo分支合并到当前分支

git reset --hard HEAD^ 本地回退到上个版本

git reset --hard HEAD~12 回退到12版本

git reset --hard 版本号 回退到指定版本

git reflog 查看版本号

TestNG：

@Test（enabled = true）默认

@BeforeMethod 每一个测试方法之前运行

@AfterMethod 每一个测试方法之后运行

@BeforeClass 类运行之前执行

@AfterClass 类运行之后执行

@BeforeSuite 在BeforeClass之前运行

@AfterSuite 在AfterClass之后运行

@BeforeGroup(“”)

@AfterGroup(“”)