# 无标题

1. 接口测试是否可以用unittest 来实现？可以不使用unittest、pytest测试框架吗？
   1. 当然可以。框架的好处。
2. 接口测试跟界面测试有什么差别？
   1. 接口是测试后台。
   2. 接口测试的成本比较小。
   3. 比界面的自动化简单。
   4. 界面覆盖不到的可以使用接口覆盖。
   5. 测试左移（前移）。
3. 项目：接口 jmeter/postman 实现自动化，界面：代码实现的自动化
   1. 不合理。界面自动化前提，接口自动化已经很完善了。
4. 迭代开发的思路开发自动化测试框架：
   1. 1.先实现一个最小可用的框架。
      1. 1.1. 环境数据写到文件 env.ini，封装一个方法，读取环境文件
      2. 1.2. 测试数据的yaml文件放到工程data\_case中，封装一个方法，读取yaml文件
      3. 1.3. 封装一个类，新建一个session，用session发送请求
      4. 1.4. 封装业务的方法，注册、登录
   2. 自动化的原则
   3. 脚本跟脚本之间，没有关联，没有执行的先后顺序。
   4. 登录（环境准备：注册用户，环境清理：删除注册用户 ）
   5. 充值（环境准备：注册用户、登录，环境清理：退出登录，删除）
5. 持续集成（git）
6. github：开源代码管理网站
7. Fork：复制代码到自己的仓库
8. 提交本地的代码到github：
   1. git config --global [user.name](http://user.name) "lhp83cn"
   2. git config --global [user.email](http://user.email) "lhp83cn@[163.com](http://163.com)"
   3. git init 初始化
   4. git add -A 给仓库中添加文件
   5. git remote add origin <https://github.com/li-huipu/ApiAutoTest.git> 跟代码库建立关联
   6. git commit -m "注释：提交代码的原因" 提交到仓库
   7. git push -u origin master
   8. git config --global credential.helper store 保存git的账户，避免每次输入
   9. git config --global --list
9. 修改文件
   1. git add ""
   2. git commit -m "修改原因"
   3. git push -u origin master
10. 删除文件
    1. git rm -v ""
    2. git commit -m "删除原因"
    3. git push -u origin master
11. git clone 克隆到本地















