# fiddler学习

1. 在哪些情况下使用fiddler？
2. 在没有接口文档的情况下，可以通过fiddler来提取需要测试的接口
3. 设置弱网
4. 捕获到客户端和服务端通讯，排查错误原因
5. 设置断点，修改数据观察特殊情况
6. fiddler功能
7. 自定义左侧会话列表的表头
8. filter过滤器
9. 通过一个实际请求，看下如何在Inspectors中提取需要的信息
10. 利用Composer模拟发送请求
11. 打断点
12. 常用快捷键

ctrl+x清楚会话

# 测试流程

* 1. 搭建测试环境
  2. 熟悉被测系统
* 项目背景调研：项目进展、项目复用程度、项目成果物、项目开发人员
* 编制组织架构：系统、子系统
* 编写测试范围列表 ：编号、测试名称、测试类别、path、优先级
  1. 制定测试计划

测试目的、测试范围、测试需求、明确测试组织和岗位职权、测试轮次安排和时间确定、

测试方法简述、测试标准模板：用例模板、缺陷模板、日志

* 1. 测试方案
* 测试策略：里程碑技术
* 测试发布标准
* 测试风险评估：需求、人员、时间
* 测试实体标准的判定：优先级、严重度、状态标准
  1. 测试设计

测试用例设计、自动化测试数据和脚本准备

* 1. 测试执行

冒烟测试、全面测试、交叉测试、生产环境主干流程

# 用例设计

* 1. 点：元素、按钮
* 输入：可见输入（长度、类型、组成规则、是否重复、是否为空）不可见输入（）
* 处理：正常处理、异常处理
* 输出：可见输出（前台、后台）、不可见输出（数据库）
  1. 线：业务场景

场景：各种功能的集合

测试设计方法：深度图和广度图、流程图

分析系统功能点

形成一个n\*n矩阵

确认矩阵中每个功能的可达性

绘制深度图或广度图：

深度图：确认一个起始功能点，绘制该功能的所有可达性，从左至右每次选择一个从未分析过的功能点，分析其功能可达性，其余所有功能结束掉，循环第三部，直至所有功能点全部分析完成。

广度图：确认一个起始功能点，绘制该功能的所有可达性，从左至右一次分析每一个未被分析过的功能的可达性，直到该层全部功能被分析完毕为止，循环第三步，直至所有功能点分析完毕为止。

从开始到结束的任意一条路径就是一条测试用例

* 1. 面：质量模型
* 功能性（重要基础）
* 效率
* 性能：时间维度、资源利用率
* web项目：浏览器、web服务器
* HTTP协议：get请求和post请求的区别
  + - * get参数在url中；安全性弱；参数是字符串类型；参数数据量小
      * post参数在请求内容中；安全性强；参数可以是任意类型；参数数据量大
      * post请求不会被浏览器缓存
* TCP和UDP：

TCP：传输控制协议

UDP：用户数据报协议

区别：TCP面向连接的，UDP无连接的

TCP可靠的，UDP不可靠的

TCP对于资源消耗大，UDP对于资源消耗小

TCP传输的数据是有顺序的，不会丢失；UDP传输的数据是无序的，存在丢包的现象

应用场景：TCP：HTTP协议，stmp，pop3协议

UDP：音视频的传输（QQ视频电话）

* 可靠性

安全性：数据加密：md5、3des、hash

用户权限：管理员、应用人员

SQL注入：or 1 = 1

* 可移植性

兼容性：

web系统：

OS操作系统：windows、linux、mac

浏览器：渲染引擎、js引擎

内核：IE、trident内核；

firefox gecko内核；

safari webkit内核；

chrome blink内核（基于webkit，谷歌和opera sofuware共同开发）

opera blink内核

* 易用性

用户体验、用户学习理解的成本

* 可扩展性

高内聚低耦合、模块化

* 1. 体：UI界面除外：接口、代码

接口维度：内部接口、外部接口

抓包： httpwatch 兼容性差 IE/Firebox