

概述

对于很多软件测试从业人员而言，可能并不具备协议相关基础，更别说去独立的构建协议抓包环境、分析协议报文，并在实际测试过程中结合业务进行接口级自动化测试。

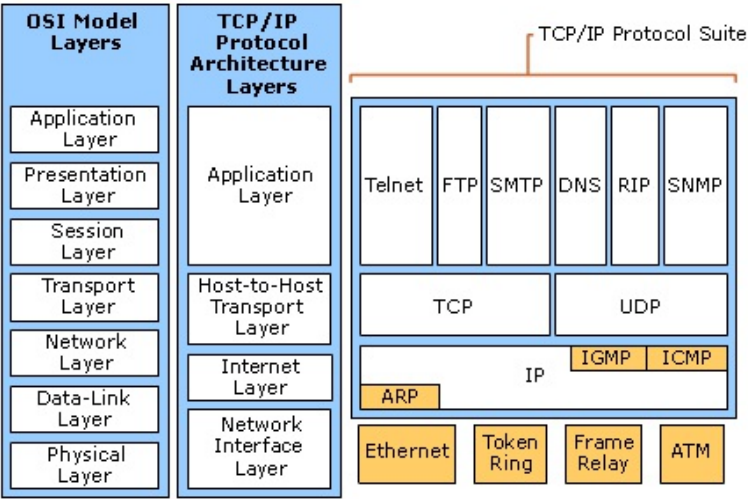
下面本节从笔者实际从业过程中，进行一些总结来阐述说明软件测试从业人员应该了解、掌握的一些协议基础、重点、服务、工具等等。、

- TCP/IP协议
- HTTP协议
- Web Service服务
- RESTful风格
- 常用工具

TCP/IP协议

首先你需要理解TCP/IP的基本说明，百度百科简要说明如下：

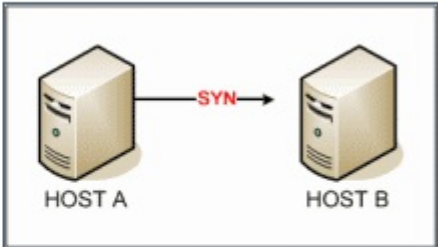
Transmission Control Protocol/Internet Protocol的简写，中译名为传输控制协议/因特网互联协议，又名网络通讯协议，是Internet最基本的协议、Internet国际互联网的基础，由网络层的IP协议和传输层的TCP协议组成。



其次你需要了解并掌握TCP/IP的架构：对

于上图，本文不做任何说明，请自行闭关领悟。

再次你则需要了解并掌握TCP三次握手建立连接过程：

EVENT	DIAGRAM
<p>Host A sends a TCP SYNchronize packet to Host B</p> <p>Host B receives A's SYN</p> <p>Host B sends a SYNchronize-ACKnowledgement</p> <p>Host A receives B's SYN-ACK</p> <p>Host A sends ACKnowledge</p> <p>Host B receives ACK.</p> <p>TCP socket connection is ESTABLISHED.</p>	 <p>TCP Three Way Handshake (SYN,SYN-ACK,ACK)</p>

注：如果

你说看不懂英文，需要看中文，请离开，谢谢

TCP四次挥手断开连接过程，本文不再列举，请自行找资料学习。

详细的TCP/IP指南请参见：<http://man.chinaunix.net/develop/rfc/RFC1180.txt>

HTTP协议

请参见：《工具篇 - HTTP协议报文结构及示例03》

HTTP 1.0: <http://man.chinaunix.net/develop/rfc/RFC1945.txt>

最新的http协议RFC文档已经被差为以下6个文档：

RFC7230 - HTTP/1.1: Message Syntax and Routing - low-level message parsing and connection management

RFC7231 - HTTP/1.1: Semantics and Content - methods, status codes and headers

RFC7232 - HTTP/1.1: Conditional Requests - e.g., If-Modified-Since

RFC7233 - HTTP/1.1: Range Requests - getting partial content

RFC7234 - HTTP/1.1: Caching - browser and intermediary caches

RFC7235 - HTTP/1.1: Authentication - a framework for HTTP authentication

注：请根据RFC编号,在<http://www.faqs.org/rfcs/>输入编号查阅文档原文。

懒啊，只好把英文的链接放这了，不爱看的请绕路~~

重点补充，对于http至少要掌握以下几个方面的知识：

1. HTTP报文结构
2. HTTP方法(PUT\GET\POST\HEAD等)应用场景
3. 常见的状态码含义(200\302\400\500等等)

4. HTTP头的构成
5. Cookies的管理

RESTful风格

作为当下web开发最为潮流的一种设计风格，软件测试从业人员必须了解RESTful风格的特性。

RESTful

REST（英文：Representational State Transfer，简称REST）描述了一个架构样式的网络系统，比如 web 应用程序。它首次出现在 2000 年 Roy Fielding 的博士论文中，他是 HTTP 规范的主要编写者之一。在目前主流的三种Web服务交互方案中，REST相比于SOAP（Simple Object Access protocol，简单对象访问协议）以及XML-RPC更加简单明了，无论是对URL的处理还是对Payload的编码，REST都倾向于用更加简单轻量的方法设计和实现。值得注意的是REST并没有一个明确的标准，而更像是一种设计的风格。

RESTful关键

RESTful的关键是定义可表示流程元素/资源的对象。在REST中，每一个对象都是通过URL来表示的，对象用户负责将状态信息打包进每一条消息内，以便对象的处理总是无状态的。RESTful的第二大问题是组合管理及流程绑定。企业对正规的（基于SOAP）SOA最大的反对声之一便是，这种等级的发现和绑定灵活性不足以适应复杂性。

理解下RESTful:

要理解RESTful架构，需要理解Representational State Transfer这个词组到底是什么意思，它的每一个词都有些什么涵义。

下面我们结合REST原则，围绕资源展开讨论，从资源的定义、获取、表述、关联、状态变迁等角度，列举一些关键概念并加以解释。

- 资源与URI
- 统一资源接口
- 资源的表述
- 资源的链接
- 状态的转移

更详细请参见Roy Thomas Fielding在他2000年的博士论文：

<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

在后续的文章中，会结合实例演示如何测试restful风格的api。

常用工具

要掌握协议，必不可少的要掌握分析工具, 对于http抓包分析，笔者推荐使用浏览器自带的插件，例如：

- firefox下的firebug
- chrome直接按F12调出调试工具

第三方抓包工具，大家可以使用：

- fiddler
- httpwatch

专业的抓包工具可以使用：

- wireshark

总结

本文由拼凑的各种资源整合而成，同时有N个外链跳接出去学习，仅此而已

扫一扫关注微信公众号：

