

本书附赠55个讲解视频, 总价值149元

体验全新AR扫图看视频模式

精通HTTP协议, 快速掌握Fiddler抓包和JMeter发包

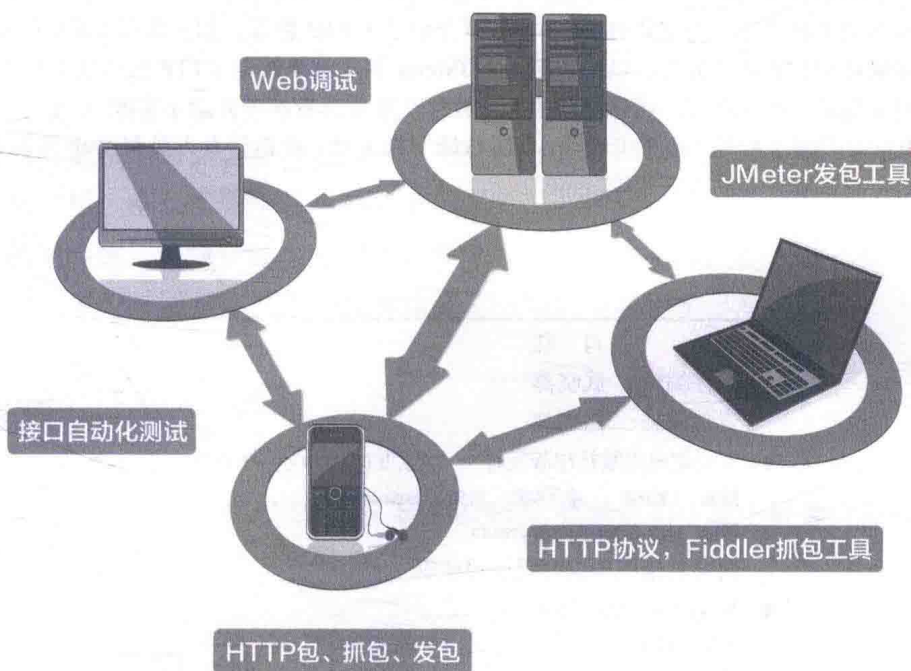
# HTTP抓包实战

肖佳 著



# HTTP抓包实战

肖佳 著



人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

HTTP抓包实战 / 肖佳著. -- 北京 : 人民邮电出版社, 2018.6  
ISBN 978-7-115-48119-1

I. ①H… II. ①肖… III. ①计算机网络—通信协议  
IV. ①TN915.04

中国版本图书馆CIP数据核字(2018)第053123号

## 内 容 提 要

HTTP 抓包利用 Fiddler 抓包工具来捕获 HTTP 数据包, 然后对其进行重发、编辑等操作。HTTP 抓包的用途非常广泛, 主要用于 Web 开发调试、软件自动化测试、接口自动化测试、性能测试和网络爬虫等方面, 也用来检查网络安全。抓包也经常被用来进行数据截取等。

本书主要围绕抓包展开。全书共有 22 章, 着重介绍了 HTTP 协议、如何使用 Fiddler 对 HTTP 包进行抓取、如何对 HTTP 进行分析, 以及如何使用 JMeter 等工具来发送 HTTP 包以实现软件的自动化测试。本书图文并茂、实例丰富, 还有配套的视频教程, 方便读者参考并动手实践。

本书适合前端开发工程师、测试工程师、线上故障支持人员、接口开发人员和 Web 开发人员阅读, 也适合对自动化测试感兴趣的人员阅读。

---

◆ 著 肖 佳

责任编辑 武晓燕

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

固安县铭成印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 13.25

字数: 298 千字

印数: 1-2 400 册

2018 年 6 月第 1 版

2018 年 6 月河北第 1 次印刷

---

定价: 59.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

# 前言

## 为什么写这本书

我从事软件测试工作 10 多年，一直有写博客的习惯，在博客园发表了上百篇关于开发和测试的高质量文章。把平常工作学的知识和业余时间倒腾的技术总结成博客，对我来说是非常好的沉淀方式，同时也可以帮助很多的技术人员。这些技术文章构成了本书的重要素材。

最近看到身边很多朋友都已经出书了，为什么我自己不出本书呢？我还曾经在培训机构担任软件测试的培训讲师，培养过上千名的软件测试人员，对于培训非常有经验。所以我感觉，写一本技术图书是水到渠成的事情了。

为了把书写好，我就开始在外面“吹牛皮”，说我要出书了。先把话说出去，然后逼着自己每天花 2 小时写书。否则牛皮吹出来了，书没写出来就丢人了。

## 本书资源

我为本书创建了一个学习 QQ 群：656587652。我将在 QQ 群中解答读者的问题，并且还会给大家一些补充的学习资料。

## 本书的核心思想

这是一本讲抓包的书。本书的核心思想可以总结为 3 个词：包、抓包、发包。

本书内容主要包括 HTTP 数据包长什么样子，如何使用工具来抓包，如何使用工具来发包。想要学会 HTTP 协议，就要同时熟练使用 Fiddler 抓包工具。想要学好 JMeter 工具，就必须先学好 HTTP 协议。

HTTP 协议、Fiddler 抓包工具、JMeter 和 Postman，这几个方面是相辅相成的，应该一起学。

## 本书的独特之处

目前市面上已经有很多经典的图书来讲 HTTP 协议和 Fiddler，但它们都是独立的书，而本书巧妙地将这些结合起来，包含了很多有趣的小例子，深入浅出地用实际的操作例子来讲述相关知识，而且还有配套的视频教程。读者可以轻松掌握从第一章到最后一章的内容，学习的过程会很顺利。

## 本书适合谁看

本书适合前端开发工程师、测试工程师、线上故障支持人员、接口开发人员和 Web 开发人员阅读。

## 本书介绍的技术适用场景

开发人员可以使用本书介绍的技术来进行 Web 开发、Web 调试等。

测试人员可以将本书介绍的技术用于做基于 HTTP 协议的自动化性能测试、Restful API 自动化测试和接口测试等。

本书介绍的技术还适合用于开发测试工具、邮箱自动登录以及开发网络爬虫等。

## 本书的内容和组织结构

本文着重介绍了 HTTP 协议，以及如何使用 Fiddler 来抓 HTTP 包，如何分析 HTTP 包。本书还介绍了如何使用 JMeter 等工具来发送 HTTP 包，实现软件自动化测试。

本书配有生动有趣的实例。本书分为 22 章，每章的内容并不多，但配了很多的图，方便读者参考并动手实践。

## 致谢

在我写书的时候，我 33 岁，感觉到了非常严重的中年危机。工作十多年了，还没什么大的成就。

写书的过程的确很耗费时间和体力。白天上班，周末还要兼职当讲师，业余时间还要教小孩英语，送小孩去各种培训班。写作本书期间，我每天晚上只睡 5 个小时，把其他时间都用在了写书上面。

要感谢高博老师的鼓励。高博老师是我以前在 VMware 的同事。有一天高博老师打电话给我，鼓励我写这本书。

另外要感谢人民邮电出版社的陈冀康编辑在本书的写作过程中给予的大力支持。

# 资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

## 配套资源

本书提供如下资源：

- 本书所有示例源代码；
- 本书作者针对书中内容的配套视频讲解。

读者请通过本书封底的刮刮卡观看，也可通过异步社区“课程”频道订阅。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

## 提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



The screenshot shows a web form titled "提交勘误" (Submit Error Report) with three tabs: "详细信息" (Detailed Information), "写书评" (Write a Review), and "提交勘误" (Submit Error Report). The form includes three input fields: "页码:" (Page Number), "页内位置 (行数):" (Page Position (Line Number)), and "勘误印次:" (Error Report Issue). Below these fields is a rich text editor with a toolbar containing icons for Bold (B), Italic (I), Underline (U), Bulleted List, Numbered List, Link, and Unlink. A "字数统计" (Character Count) label is located at the bottom right of the text area. A "提交" (Submit) button is positioned at the bottom right of the form.

## 扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



## 与我们联系

我们的联系邮箱是 [contact@epubit.com.cn](mailto:contact@epubit.com.cn)。

如果您对本书有任何疑问或建议，请您发邮件给我们，请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 [www.epubit.com/selfpublish/submission](http://www.epubit.com/selfpublish/submission) 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

## 关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



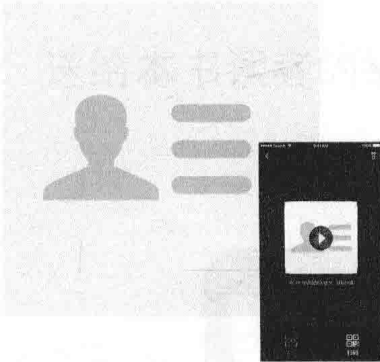
异步社区



微信服务号



## 一图



从书上直接用AR方式看视频！  
只需下载“卷积”应用并扫描“一图一码”  
设计中的AR触发图片，立刻就能看！



## 一码



PRD 001

还没有考虑好下载一个新的应用？  
直接用微信扫码看视频！

关注“内容市场”公众号，  
浏览所有图书和订阅内容。

## 一站式体验



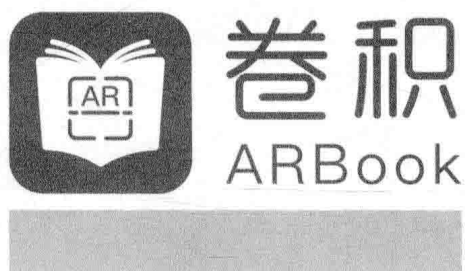
1. 浏览图书详情

2. 进入作者主页

3. 查看作者专栏

4. 订阅精彩内容

送给本书读者的福利就在封底刮刮卡中



每本图书的封底都有一个刮刮卡

刮刮卡一般位于页面底部居中或离定价标签较近的地方

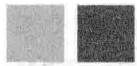
刮开涂层可以看到一个序列号

使用该序列号即可免费解锁并观看本书中包含的所有视频

(一张刮刮卡只能和一个帐号进行一次性的绑定)

卷积文化发展(上海)有限公司是从事以版权开发和版权合作为基础的、以虚拟现实、增强现实、人工智能和海量数据处理等技术研发为驱动的、以付费订阅内容开发和运营为核心业务的新兴传媒公司。

卷积文化发展(上海)有限公司是人民邮电出版社在异步社区的独家IT出版服务合作伙伴。



# 目录

第 1 章 HTTP 协议和 Fiddler 抓包	1
1.1 HTTP 协议介绍	1
1.1.1 什么是 HTTP 协议	1
1.1.2 如何学习 HTTP 协议	1
1.1.3 HTTP 协议的工作原理	2
1.2 Fiddler 的介绍	2
1.2.1 Fiddler 的下载和安装	3
1.2.2 Fiddler 的基本界面	3
1.2.3 Inspectors 选项卡	4
1.2.4 Web Sessions 列表	5
1.2.5 Fiddler 捕获 HTTP 协议的 数据包	5
1.2.6 Fiddler 设置开始捕获和 停止捕获	6
1.3 HTTP 协议报文的结构	7
1.3.1 HTTP 请求报文的结构	7
1.3.2 HTTP 响应报文的结构	8
1.3.3 Fiddler 捕获博客主页, 查看 HTTP 请求和 HTTP 响应报文	9
1.4 Fiddler 抓包的原理	10
1.4.1 什么是代理服务器	10
1.4.2 Fiddler 的工作原理	11
1.4.3 查看 Internet 选项代理 设置	11
1.4.4 Fiddler 如何捕获 Firefox	12
1.4.5 Fiddler 能捕获哪些设备的 HTTP 数据包	13
1.4.6 解压 HTTP 响应	14
第 2 章 HTTPS 协议和 Fiddler 抓包	15
2.1 HTTP 协议是不安全的	15
2.2 Web 通信如何做到安全	15
2.3 什么是 HTTPS	17
2.4 Fiddler 如何捕获 HTTPS 会话	17
2.4.1 添加例外绕过 HTTPS 证件 错误	18
2.4.2 Firefox 中安装证书	20
2.4.3 Fiddler 可以捕获 HTTPS 的 握手验证请求	21
2.4.4 查看 Windows 本地安装的 证书	22
第 3 章 HTTP 协议请求方法和状态码	23
3.1 URL 详解	23
3.1.1 URL 格式	23
3.1.2 URL 中的锚点	24
3.2 HTTP 请求方法	24
3.2.1 GET 方法	25
3.2.2 带参数的 GET 方法	26
3.2.3 POST 方法	26
3.2.4 GET 和 POST 方法的 区别	28
3.3 HTTP 状态码	28
3.3.1 什么是 HTTP 状态码	28
3.3.2 状态码分类	28

3.3.3	常见的状态码	29	第5章	Web 网页抓包和 Fiddler	
3.3.4	200 (OK)	29		修改包	46
3.3.5	204 (No Content, 没有内容)	30	5.1	网页是如何打开的	46
3.3.6	206 (Partial Content, 部分内容)	31	5.1.1	一个网页的组成	46
3.3.7	301 (Moved Permanently)	32	5.1.2	打开一个网页, 浏览器需要发送很多个请求	47
3.3.8	302 (Found)	33	5.1.3	用 Fiddler 查看一个 Web 页面打开的过程	47
3.3.9	301 和 302 的区别	33	5.1.4	用 Fiddler 选择请求	48
3.3.10	304 (Not Modified)	34	5.2	Web 页面简单的性能测试	49
3.3.11	400 (Bad Request)	34	5.3	使用 Fiddler 来查看响应	50
3.3.12	401 (Unauthorized)	35	5.4	Fiddler 下断点, 修改 HTTP 报文	51
3.3.13	403 (Forbidden)	36	5.4.1	Fiddler 中设置断点修改 HTTP 请求	51
3.3.14	404 (Not Found)	36	5.4.2	实例: Fiddler 修改 HTTP 请求	52
3.3.15	500 (Internal Server Error)	36	5.4.3	Fiddler 中设置断点修改 HTTP 响应	53
3.3.16	503 (Server Unavailable)	38	5.4.4	Fiddler 修改网页的标题	54
			5.4.5	伪造 Referer	55
第4章	HTTP 协议 Header 介绍	39	第6章	HTTP 协议中的缓存	56
4.1	HTTP Header 介绍	39	6.1	缓存的概念	56
4.2	Fiddler 查看 HTTP 请求 Header	39	6.2	缓存的优点	56
4.2.1	Cache 相关的 Header	40	6.3	Fiddler 可以方便地查看缓存的 Header	57
4.2.2	Cookies	41	6.4	如何判断缓存新鲜度	57
4.2.3	Accept	41	6.5	通过最后修改时间来判断缓存新鲜度	58
4.2.4	Accept-Encoding	41	6.6	与缓存有关的 Header	59
4.2.5	Accept-Language	41	6.7	ETag	60
4.2.6	User-Agent	42	6.8	浏览器不使用缓存	61
4.2.7	实例: Fiddler 修改 User-Agent, 伪装客户端	42	6.9	直接使用缓存, 不去服务器验证	62
4.2.8	Referer	43	6.10	如何设置 IE 不使用缓存	63
4.2.9	Connection	44	6.11	公有缓存和私有缓存的区别	64
4.2.10	Host	44			
4.3	Fiddler 查看 HTTP 响应 Header	44			
4.4	Fiddler 查看和复制 Header	45			

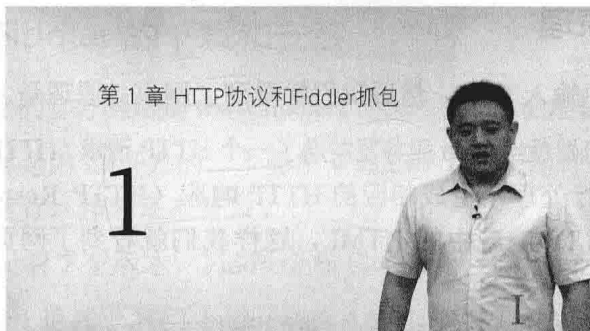
<b>第 7 章 HTTP 协议压缩和 URL Encode</b> ··· 65	
7.1 HTTP 压缩的过程····· 65	
7.1.1 实例: Fiddler 观察 HTTP 压缩····· 66	
7.1.2 内容编码类型····· 67	
7.1.3 压缩的好处····· 68	
7.1.4 gzip 的不足之处····· 68	
7.1.5 gzip 是如何压缩的····· 68	
7.1.6 HTTP 请求也是可以 编码的····· 68	
7.1.7 HTTP 内容编码和 HTTP 压缩的区别····· 69	
7.2 URL Encode 介绍····· 69	
7.2.1 查询字符串中包含汉字··· 69	
7.2.2 POST 中的数据包含汉字·· 70	
7.3 Fiddler 中的 Text Wizard····· 71	
<b>第 8 章 Fiddler 使用技巧</b> ····· 72	
8.1 Fiddler 和其他抓包软件的比较··· 72	
8.2 Fiddler 抓不到包应该怎么解决··· 73	
8.3 如何找到想抓的包····· 73	
8.4 Fiddler 异常退出后无法上网···· 73	
8.5 Fiddler 排序····· 74	
8.6 Fiddler 中查询会话····· 74	
8.7 Fiddler 中保存抓到的包····· 75	
8.8 Fiddler 中编辑会话····· 75	
8.9 过滤会话····· 75	
8.10 常用快捷键····· 76	
8.11 QuickExec 命令行的使用····· 77	
8.12 Fiddler 比较会话的不同····· 78	
8.13 Fiddler 插件····· 79	
8.13.1 JavaScript Formatter····· 79	
8.13.2 Gallery 插件····· 79	
<b>第 9 章 Fiddler 前端快速调试</b> ····· 81	
9.1 如何在服务器上调试 JavaScript 文件····· 81	
9.2 Fiddler AutoResponder 的工作 原理····· 82	
9.3 Fiddler 在线调试 JavaScript 文件··· 82	
9.4 浪漫的程序员····· 85	
9.5 替换网页中的图片····· 86	
<b>第 10 章 Fiddler 的 Script 用法</b> ····· 89	
10.1 Fiddler Script 介绍····· 89	
10.2 Fiddler Script Editor····· 90	
10.3 CustomRules.js 中的主要方法···· 91	
10.4 Fiddler 定制菜单····· 92	
10.5 修改 Session 在 Fiddler 的显示 样式····· 92	
10.6 修改 HTTP 请求····· 93	
10.6.1 修改 HTTP 请求中的 Cookie····· 93	
10.6.2 替换 HTTP 请求的 Host 地址····· 94	
10.6.3 修改 HTTP 请求中的 Header····· 94	
10.6.4 修改 HTTP 请求中的 Body····· 95	
10.7 修改 HTTP 响应····· 95	
10.8 读写 txt 文件····· 96	
10.9 使用正则表达式····· 96	
10.10 保存 Session····· 97	
10.11 读取 Session, 并且使用 Fiddler 来发送····· 97	
<b>第 11 章 深入理解 Cookie 机制</b> ····· 98	
11.1 HTTP 协议是无状态的····· 98	
11.2 会话机制····· 98	
11.3 Cookie 机制····· 100	
11.4 Cookie 是什么····· 100	
11.5 Cookie 的作用····· 101	
11.6 抓包观察上海科技馆网站的 登录····· 102	

11.7	Cookie 的属性	103
11.8	Cookie 的分类	104
11.9	Cookie 保存在哪里	104
11.10	使用和禁用 Cookie	105
11.11	网站自动登录的原理	106
11.12	Cookie 和文件缓存的区别	106
11.13	Cookie 泄露隐私	107
<b>第 12 章</b>	<b>Fiddler 实现 Cookie 劫持</b>	
	攻击	108
12.1	截获 Cookie 冒充别人身份	108
12.2	Cookie 劫持的原理	108
12.3	Cookie 劫持实例介绍	109
12.3.1	找到登录的 Cookie	109
12.3.2	浏览器中植入 Cookie	111
12.4	网站退出的作用	112
<b>第 13 章</b>	<b>HTTP 基本认证</b>	113
13.1	什么是 HTTP 基本认证	113
13.1.1	路由器管理页面使用基本认证	114
13.1.2	HTTP 基本认证的优点	117
13.1.3	HTTP 基本认证的缺点	117
13.1.4	使用 TextWizard 工具	117
13.1.5	客户端的使用	117
13.2	摘要认证	118
<b>第 14 章</b>	<b>Fiddler 手机抓包</b>	119
14.1	环境准备	119
14.2	Fiddler 截获手机原理图	119
14.3	截获手机发出的 HTTP 包有什么作用	120
14.4	手机抓包	120
14.4.1	配置 Fiddler 允许“远程连接”	120
14.4.2	获取 Fiddler 所在机器的 IP 地址	121
14.4.3	手机上设置代理服务器	121
14.4.4	测试 Fiddler 捕获手机发出的 HTTP	123
14.4.5	捕获手机上的 HTTPS	123
14.4.6	Apple 设备需要使用插件制作新证书	123
14.4.7	iOS 设备安装证书方法	124
14.4.8	Android 设备安装证书方法一	125
14.4.9	Android 设备安装证书方法二	126
14.4.10	测试 Fiddler 捕获手机的 HTTPS	127
14.5	设置过滤	127
14.6	如何卸载证书	128
14.7	手机抓包提醒	128
<b>第 15 章</b>	<b>Fiddler 发送 HTTP 请求</b>	129
15.1	Fiddler Composer 发送 HTTP 请求	129
15.1.1	Composer 发送 Get 请求	129
15.1.2	Composer 的编辑模式	130
15.1.3	Composer 发送 Post 请求	130
15.1.4	Composer 编辑之前捕获的 HTTP 请求	131
15.2	Fiddler 重新发送 HTTP 请求	131
15.2.1	Replay 菜单	131
15.2.2	简单的性能测试	132
15.2.3	先编辑再发送	133
15.3	安全测试之重放攻击	133
15.3.1	重放攻击是怎么发生的	133
15.3.2	重放攻击的危害	133
15.3.3	重放攻击的解决方案	133
15.3.4	APP 验证码重放	134

15.4 查找和登录相关的 Cookie	136	19.5 处理 JMeter 中 HTTP 响应乱码	163
<b>第 16 章 Fiddler 实现弱网测试</b>	139	19.6 添加验证点	164
16.1 什么是弱网	139	19.7 使用用户自定义变量	165
16.2 弱网环境带来的问题	140	19.8 正则表达式提取城市地区代码	166
16.3 弱网测试的目的	140	19.9 获取天气	167
16.4 弱网的场景	140	<b>第 20 章 JMeter 中 BeanShell 的用法</b>	169
16.5 Fiddler 模拟网络延迟	140	20.1 什么是 BeanShell	169
16.6 精确控制网速	141	20.2 操作变量	169
16.7 Fiddler 模拟网络中断	142	20.3 JMeter 有哪些 BeanShell	170
16.8 实例: Fiddler 返回 500 状态码	142	20.4 BeanShell 调用自己写的 jar 包进行 MD5 加密	171
16.9 Fiddler 模拟网络超时	143	<b>第 21 章 自动登录禅道和自动开 Bug</b>	174
<b>第 17 章 自动化测试和接口测试</b>	144	21.1 HTTP Cookie 管理器	174
17.1 自动化测试分类	144	21.2 HTTP 请求默认值	175
17.2 分层的自动化测试理念	144	21.3 禅道介绍和部署	175
17.3 Web 自动化测试的两种思路	146	21.4 禅道操作和抓包分析	176
17.4 什么是接口测试	146	21.4.1 第 1 步, 自动登录禅道	176
17.5 接口测试工具	147	21.4.2 第 2 步, 创建一个新的 Bug	181
17.6 Web 原理	147	21.4.3 第 3 步, 找到刚刚新建 Bug 的 ID	184
17.7 Web 自动化测试原理	148	21.4.4 第 4 步, 修改 Bug 状态为“已解决”	185
17.8 性能测试的原理	148	21.4.5 第 5 步, 关闭 Bug	187
17.9 APP 的后台测试	149	21.4.6 总结	187
17.10 如何学习 Web 自动化测试和性能测试	149	<b>第 22 章 JMeter 给网站做压力测试</b>	188
<b>第 18 章 JMeter 工具使用介绍</b>	151	22.1 案例介绍	188
18.1 JMeter 介绍	151	22.2 压力测试的目的	188
18.2 JMeter 的下载和运行	151	22.3 抓包分析 Mozy 网站的登录过程	189
18.3 创建测试任务	152	22.4 抓包分析	190
18.4 添加 HTTP 请求	153	22.5 实现 Mozy 登录	191
18.5 实例: 密码用 MD5 加密	154	22.6 简单的压力测试	193
<b>第 19 章 JMeter 天气接口自动化测试</b>	160		
19.1 天气查询的例子	160		
19.2 天气查询网站抓包	160		
19.3 抓包分析	161		
19.4 获取城市地区代码	162		

# ■■ 第 1 章 ■■

## —— HTTP 协议和 Fiddler 抓包 ——



Web 浏览器和 Web 服务器之间是通过 HTTP 协议相互通信的。HTTP 协议用途非常广泛，HTTP 协议是任何 IT 从业人员都需要掌握的基础知识。

当今 Web 程序的开发技术真是百家争鸣，有 ASP.NET、PHP、JSP、Perl、AJAX 等。无论 Web 技术在未来如何发展，理解 Web 程序之间通信的基本协议相当重要。

### 「 1.1 HTTP 协议介绍 」

#### 1.1.1 什么是 HTTP 协议

协议是指计算机通信网络中两台计算机之间进行通信所必须共同遵守的规定或规则。

超文本传输协议（HyperText Transfer Protocol，HTTP）是互联网上应用最广泛的一种网络协议，它允许将超文本标记语言（HTML）文档从 Web 服务器传送到客户端的浏览器。

目前我们使用的 HTTP 协议是 HTTP/1.1 版本。

#### 1.1.2 如何学习 HTTP 协议

协议是很抽象的东西，想要学好 HTTP 协议，必须先了解 HTTP 协议的基本知识；然



后找一个抓包软件实实在在地看到数据包的内容，并且看到数据包是如何在浏览器和 Web 服务器中进行交互的。这才是学习 HTTP 的正确方法。

Fiddler 就是我们需要的抓包工具。你对 HTTP 协议越了解，你就能越掌握 Fiddler 的使用方法。你越使用 Fiddler，它就越能帮助你了解 HTTP 协议。HTTP 协议和 Fiddler 是相辅相成的，应该一起学习。

我们通过 Fiddler 抓包的方式来学习 HTTP 协议。

### 1.1.3 HTTP 协议的工作原理

我们打开浏览器，在地址栏中输入 URL，然后我们就看到了网页。原理是怎样的呢？

实际上，我们输入 URL 后，浏览器就给 Web 服务器发送了一个 HTTP 请求 (HTTP Request)，Web 服务器接到 HTTP 请求后进行处理，生成相应的 HTTP 响应 (HTTP Response)，然后发送给浏览器。浏览器解析 HTTP 响应中的 HTML，这样我们就看到了网页。该过程如图 1-1 所示。

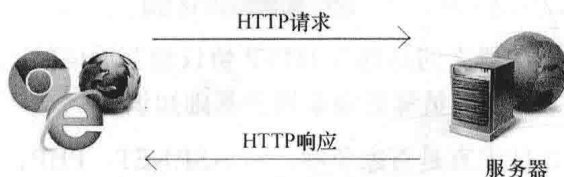


图 1-1 HTTP 协议工作原理

浏览器客户端和 Web 服务器之间是通过 HTTP 协议来交流的。我们每天都会用浏览器浏览各种网站。目前主流的 Web 浏览器有微软的 Internet Explorer、Firefox 和 Google 的 Chrome。

Web 浏览器会给 Web 服务器发送一条 HTTP 请求，服务器会把 Web 对象发送给浏览器，浏览器解析 Web 对象，这些对象就显示在屏幕上了。

通过上面的介绍我们已经了解了 HTTP 协议的工作原理。那么 HTTP 请求和 HTTP 响应的数据包 (报文) 具体有哪些内容呢？协议是抽象的东西，是看不到的。下面我们使用 Fiddler 来抓包查看里面的内容。

就好比如果你想学习 TCP/IP 协议，你可以使用 Wireshark 来抓包学习里面的内容。

## 「 1.2 Fiddler 的介绍 」

Fiddler 是世界上最强大最好用的 Web 调试工具，可以称得上是“神器”。其用途非常