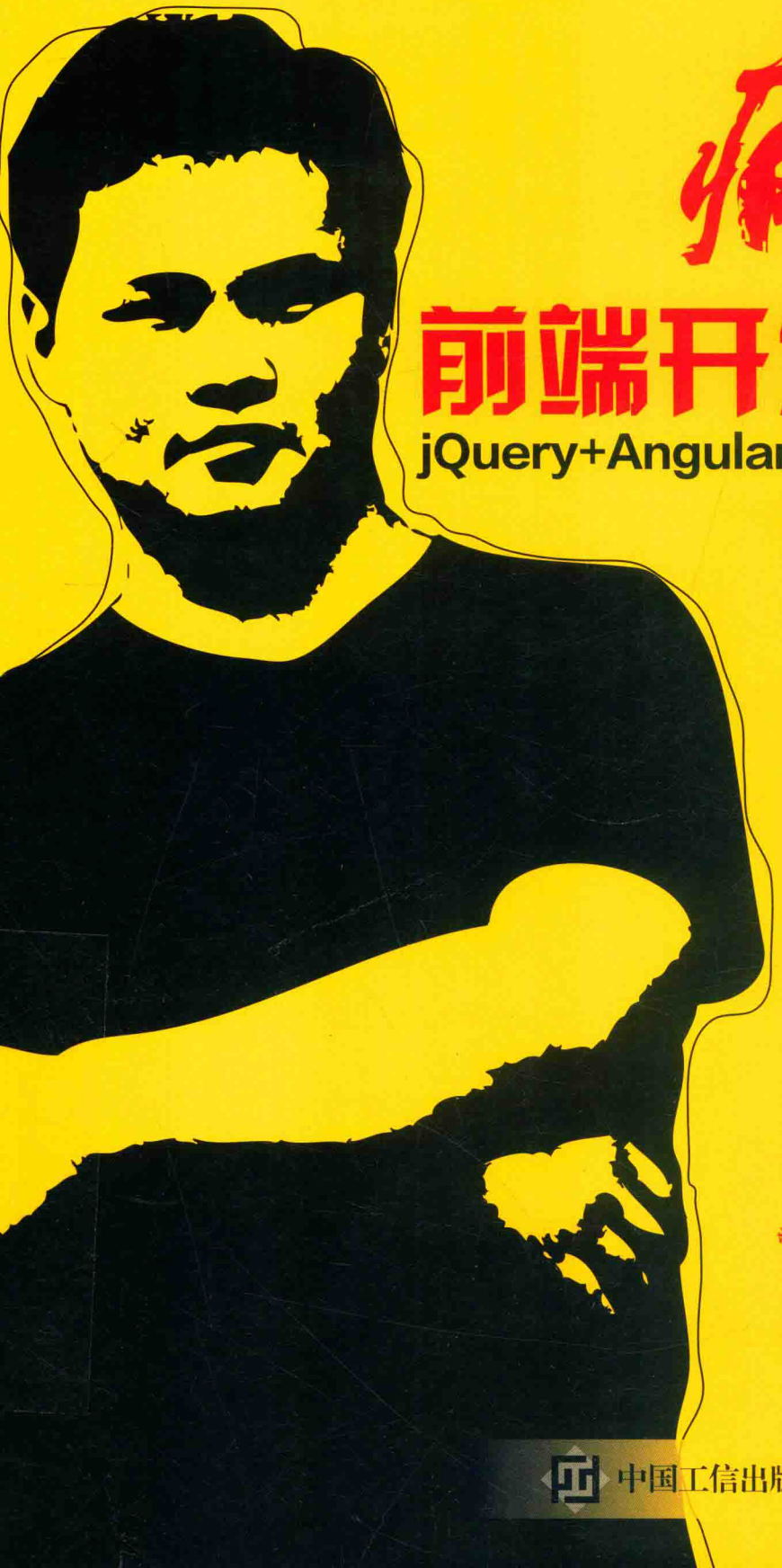
Broadview
www.broadview.com.cn

FK 疯狂软件教育 www.fkjava.org 标准教材



疯狂

前端开发讲义

jQuery+AngularJS+Bootstrap 前端开发实战

李刚 编著

疯狂源自梦想

技术成就辉煌

疯狂源自梦想

技术成就辉煌

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

疯狂前端开发讲义

jQuery+AngularJS+Bootstrap 前端开发实战

李刚 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书基于《疯狂 Ajax 讲义（第 3 版）》部分升级而来，全书升级了 HTML 5.1 支持的 XMLHttpRequest，jQuery 升级到 3.1。本书的重点是新加入的两个目前十分主流的前端框架：AngularJS 和 Bootstrap。本书详细、全面地介绍了 AngularJS 和 Bootstrap 的知识，由于这两个框架是本书的重点，因此本书花了近 400 多页来介绍它们的功能和用法，这部分内容独立出来完全可以作为 AngularJS 和 Bootstrap 的学习手册。

全书主要就是讲解 jQuery 3.1、AngularJS 1.6、Bootstrap 3.3 这三个最常用的前端框架，并针对每个框架都提供了实用的案例，让读者理论联系实际。这部分内容是“疯狂软件教育中心”的标准讲义，既包含了实际前端开发的重点和难点，也融入了大量学习者的学习经验和感悟。笔者以丰富的授课经验为基础，在讲解时深入浅出，力求使读者真正掌握前端开发的精髓。

本书最后提供了两个综合性案例：图书管理系统和电子拍卖系统，这两个项目都综合利用了 jQuery、AngularJS、Bootstrap 前端开发技术，并在后端采用了最流行、最规范的轻量级 Java EE 架构：控制器层->业务逻辑层->数据持久化层。这两个案例对实际项目具有极好的指导价值和借鉴意义。案例既提供了 IDE 无关的、基于 Ant 管理的项目源码，也提供了基于 Eclipse IDE 的项目源码，最大限度地满足读者的需求。如果在阅读本书时遇到任何技术问题，都可登录 http://www.crazyit.org 与本书庞大的读者群交流。

本书并非针对零基础的读者，本书不再包含 HTML、CSS、JavaScript 相关知识，这些知识是阅读本书的基础。本书适合有初步 HTML、CSS、JavaScript 基础的读者，或对企业应用前端开发不太熟悉的开发人员。如果您已经掌握本书上篇：《疯狂 HTML 5/CSS 3/JavaScript 讲义》的内容，将非常适合阅读本书。

图书在版编目（CIP）数据

疯狂前端开发讲义：jQuery+AngularJS+Bootstrap 前端开发实战 / 李刚编著. —北京：电子工业出版社，2017.10

ISBN 978-7-121-32680-6

Ⅰ. ①疯… Ⅱ. ①李… Ⅲ. ①网页制作工具－JAVA 语言－程序设计②网页制作工具－超文本标记语言－程序设计 Ⅳ. ①TP393.092.2②TP312.8

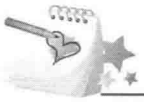
中国版本图书馆 CIP 数据核字（2017）第 223292 号

策划编辑：张月萍
责任编辑：牛　勇
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1092　1/16　　印张：33.5　　字数：900 千字
版　　次：2017 年 10 月第 1 版
印　　次：2017 年 10 月第 1 次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。



前　言

现在企业应用开发越来越重视前端开发，很多企业已经独立出一个专门的岗位：前端开发工程师，前端开发工程师专门负责企业应用的前端编程。前端工程师与美工不同，美工负责的是应用界面的规划和设计，他们只负责做出静态的界面构图，有时也包括规划界面交互效果，但美工通常并不懂编程实现；而前端开发工程师则负责使用 HTML 5、CSS、JavaScript 以及各种前端框架来实现整个前端界面：包括应用界面的搭建、用户交互的实现，这些内容其实都属于 JavaScript 编程。

有些初学者往往容易混淆美工和前端开发工程师的区别：他们以为会用网页设计工具制作网页就算前端开发工程师。实际上会用网页设计工具既算不上美工，也算不上前端开发工程师。美工的重点在“美”，他们需要有扎实的美术功底，他们可用任何工具来“设计”静态界面，甚至可直接在纸上绘制界面草图，因此网页制作工具只是其中的工具之一；而前端开发工程师往往并不使用网页设计工具，他们通常会使用专门的代码编写工具，他们用 HTML、CSS 来编写静态界面，再使用 JavaScript 或各种前端框架来实现界面的动态交互效果。因此，前端开发工程师的本质依然是程序员，他们常用 JavaScript 以及各种 JavaScript 框架，如 jQuery 和 AngularJS 等。

这里必须说明，前端开发并不简单，原因主要有如下两点：

- JavaScript 作为一门编程语言，有其自身的优势和特点，要想真正掌握并熟练使用它，有一定的难度。
- 各浏览器之间存在一定的差异（虽然这种差异正在被相应的规范减少），而且同一个浏览器的不同版本之间也存在较大差异（如 IE 8、IE 9、IE 10 等），这同样给广大开发者带来了较大的困难。

正因为前端开发存在的复杂性，使得前端开发的相关框架也相当活跃，除了最主流的 jQuery 之外，其他还有如 AngularJS、Bootstrap 等。尤其是 jQuery，目前它已经超出一般框架的概念，甚至变成了某种规范，很多前端框架底层都会借鉴或依赖 jQuery。

AngularJS 则是一个非常实用的前端框架，这个框架不是一种简单的工具，它甚至会强迫开发者采用一种优雅的前端架构，这也是笔者对这个框架情有独钟的原因。

对于初级甚至中级的前端开发者来说，他们有一定的 JavaScript 代码功底，如果单纯交代他们实现一个前端功能，他们可能也可以实现出来，但他们的实现风格要么乱七八糟，要么“随心所欲”：今天可能这样实现，明天可能会那样实现——如果整个前端都由一个开发者完成，这样做可能问题不大。但实际企业应用的前端往往需要多人协作开发，这时各前端开发者的风格不统一就会带来巨大的沟通成本，从而导致项目进度受阻。

AngularJS 的价值就在于此，AngularJS 可以极好地规范前端开发的风格，AngularJS 对前端开发进行分层，它将前端开发分为 Controller 层、Service 层、DAO 层和 Model 层，Model 对象则与 HTML 页面（视图）上 HTML 元素进行双向绑定，这样开发者可通过 Controller 调用 Service、DAO 与后端交互，获取后端数据之后，只要修改其中 Model 对象的值，视图页面也会随之动态改变。这个设计架构层次非常清晰，而且具有一定的“强制性”，整个前端团队一旦采用 AngularJS 框架，那么整个前端开发风格会变得简单、清晰，所有团队成员都能采用一致的开发风格，这就是 AngularJS 的魅力所在。

Bootstrap 则更像一个 CSS 框架，使用起来非常方便。对于大部分前端开发者，最令人头疼

的可能并不是“如何实现”某个界面，而是不清楚到底“要实现什么”界面？怎样的界面才能算得上优雅、美观？而 Bootstrap 正好解决了这个痛点，该框架提供了大量优雅、美观的 CSS 样式，开发者直接应用这些 CSS 样式即可实现优雅、美观的页面。从这个角度来看，Bootstrap 的上手非常简单，甚至不要求开发者掌握 JavaScript 知识，只要开发者略懂 CSS 样式即可，因此 Bootstrap 完全是真正简单且强大的前端框架。

本书有什么特点

本书只是一本介绍前端开发的图书，不是一本关于所谓“思想”的书，更不是一本看完之后可以“吹嘘、炫耀”的书——因为本书并没有堆砌一堆“深奥”的新名词、一堆“高深”的思想，本书保持了“疯狂 Java 体系”的一贯风格：操作步骤详细，编程思路清晰，语言平实。

本书带给读者的只是 9 个字：“看得懂、学得会、做得出”，本书并不能让你认识一堆新名词，只帮助你掌握扎实的企业前端开发基础。对于本书，光“看”是不够的，一定要“做”，阅读本书的同时，应该把所有知识点的配套示例都做出来，这样才能真正掌握本书的知识。

无论如何，读者在阅读本书时遇到知识上的问题，都可以登录疯狂 Java 联盟（http://www.crazyit.org）与广大 Java 学习者交流，笔者也会通过该平台与大家一起交流、学习。访问 www.broadview.com.cn/32680 下载相关资源。

除此之外，本书还有如下几个特点。

1．通俗易懂，适合自学

该书吸收了大量学习者的学习体会和心得，并重点讲解了学习过程中难以理解和掌握的知识点，降低了学习者的学习难度。

2．知识丰富，内容全面

本书全面、详细地介绍了 jQuery 3.1、AngularJS 1.6、Bootstrap 3.3 三个框架，它们是企业开发中最主流的前端框架，具有很强的代表性。掌握本书内容即可具备扎实的前端开发功底。

3．深入实用，实践性强

本书并不是一本前端开发的入门图书，本书全面、深入地介绍了企业开发中最主流、最具代表性的前端框架，并将它们真正融入 Java 企业应用开发中，这对实际企业应用开发具有极好的指导意义。

本书写给谁看

本书并非针对零基础的读者，如果您具有 HTML 5、CSS 3、JavaScript 基础，认真学习此书即可让您成为前端开发的实战型人才；如果已经阅读过本书上篇：《疯狂 HTML 5/CSS/JavaScript 讲义》，阅读本书非常合适。如果完全没有 HTML、CSS、JavaScript 基础知识，建议您暂时不要购买、阅读此书。

2017-07-28

目 录 CONTENTS

CHAPTER

1

第 1 章
前端开发与 Ajax 技术

本章要点

- C/S 模式应用的结构和缺点
- B/S 模式应用的结构和优势
- 传统 Web 应用的不足
- 如何改进传统的 Web 应用
- 现代 Web 应用与前端开发
- 改进的通信方式和增强的 UI 界面
- 体验前端 Ajax 开发
- 使用 Servlet 生成文本响应
- 使用 JSP 生成文本响应
- 获取服务器的响应内容
- 通过 DOM 加载服务器响应
- 前端开发的技术难点

1.1 重新思考 Web 应用

传统的 Web 应用经过多年的发展，在很多方面都是相当完善的。特别是 Java EE、PHP 等技术的广泛应用，使得目前绝大部分应用都采用了基于请求—响应的 Web 架构。

1.1.1 应用系统的发展史

早期应用软件系统大都采用 C/S（客户机/服务器模式）结构，C/S 结构的软件分为客户机和服务器两层。客户机不是毫无运算能力的输入/输出设备，在客户端需要部署大量的应用程序，而且可能还具有一定的数据存储能力。

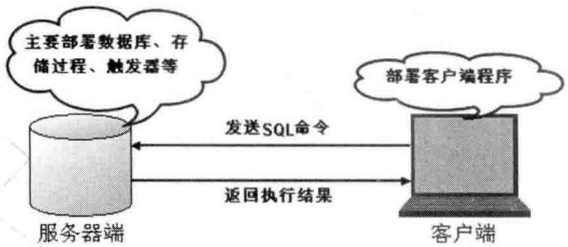


图 1.1　C/S 结构的应用

C/S 结构应用的服务器端通常主要安装数据库管理系统，当然也可能包含一些业务逻辑实现（这些业务逻辑实现通常以函数、存储过程和触发器的形式存在）。通过把软件系统的计算和数据合理地分配在客户机和服务器两端，可以有效地降低网络通信量和服务器运算量。

C/S 结构应用的结构图如图 1.1 所示。

对于 C/S 结构的应用，因为可以直接在客户端部署应用程序，所以可以让应用的人机交互界面更加友好，并可充分美化应用程序的人机界面。但由于服务器连接个数和数据通信量的限制，这种结构的软件适用于用户数目不多的局域网内。早期的大部分 ERP 软件产品即属于此类结构。

随着 Internet 技术的兴起，B/S（浏览器/服务器模式）结构得到了大规模应用。B/S 结构是对 C/S 结构的一种改进。在这种结构下，应用的业务逻辑完全在应用服务器端实现，用户表现完全在 Web 服务器上实现，客户端只需要浏览器即可进行操作，是一种全新的软件系统结构技术。这种结构是当今应用软件的首选体系结构。

在这种应用结构下，客户端的所有处理请求都以 HTTP 请求形式发送，而服务器端则将响应以 HTML 页面的形式送回客户端，由客户端浏览器负责显示 HTML 页面。B/S 结构应用的结构图如图 1.2 所示。

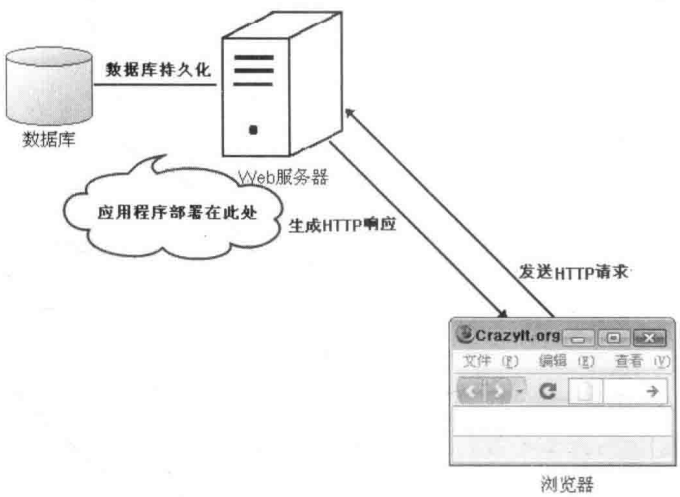


图 1.2　B/S 结构的应用

B/S 结构得到迅速推广是有原因的。在大部分情况下，B/S 结构的应用比 C/S 结构的应用更加优秀，适应性更广。相对而言，B/S 结构的系统有如下方面的优势：

- **数据安全性高**。由于 C/S 结构软件的数据分布特性，客户端所发生的火灾、地震、盗抢、病毒、黑客攻击等都成了可怕的数据杀手。另外，对于集团级的异地软件应用，C/S 结构的软件必须在各地安装多台服务器，并在多台服务器之间进行数据同步。因此，每个数据点上的数据安全都会影响整个应用的数据安全。对于跨区域的大型应用，C/S 结构软件的安全性是令人无法接受的。而对于 B/S 结构的系统，数据集中存放于总部的数据库服务器（服务器数据可以通过多种方式备份存放），客户端不保存任何业务数据和数据库持久化信息，无须进行数据同步，所以这些安全问题自然也就不存在了。
- **数据一致性好**。在 C/S 结构的解决方案里，对于跨区域的大型企业应用，都采用各地安装区域级服务器，然后再进行数据同步的模式。每天必须对这些服务器进行同步之后，总部才可得到最终的数据。由于局部网络故障等原因，可能造成个别数据库无法正常同步。即使都能正常同步，但各服务器往往不能同时同步，因而数据也无法绝对一致，这当然不能用于决策。而对于 B/S 结构的系统，数据是集中存放的，客户端发生的每次数据修改都直接进入到中央数据库，不存在数据一致性的问题。
- **数据实时性好**。对于大型的跨区域应用，C/S 结构不可能随时跟踪各客户端的业务发生情况，因为数据都不是实时更新的，因而看到的数据都是滞后的；而 B/S 结构则不同，因为其数据都是实时存入服务器端的数据库，因而服务器端可以实时看到当前发生的所有业务，能提供更好的企业决策支持。
- **系统更新方便**。软件供应商提供的软件不可能是完美无缺的。即使是一个绝对完美的软件系统，当具体的业务环境发生改变后，系统也应随之改变。因而，必须经常对已部署的软件产品进行维护、升级。对 C/S 结构的软件而言，由于其应用是分布式的，需要手动对每一个节点进行程序安装，所以，即使非常小的程序缺陷都需要很长的时间来重新部署。重新部署时，为了保证各程序版本的一致性，必须暂停业务进行更新（即“休克更新”）。在很多情景下，这是不可忍受的。而 B/S 结构的软件则不同，其应用程序集中于服务器上，各应用节点没有任何程序，应用的更新只需要更新服务器端程序即可，因而可以做到快速的服务响应。

传统的 C/S 结构软件开发工具早期的有 Visual Basic、Visual FoxPro 等，这些开发工具目前基本已经被淘汰。目前使用较少的开发工具如 PowerBuilder、Delphi 等也趋于被淘汰。除早期的一些系统外，现在新开发的系统大部分使用 B/S 结构。

目前的 B/S 结构应用通常都会严格遵守 MVC 模式。MVC 模式将应用分成 Model（M：模型）、View（V：视图）和 Controller（C：控制器）三个部分。MVC 模式分离的数据访问和数据表现，给系统提供了更好的解耦。MVC 架构的核心思想是，将程序分成相对独立而又能协同工作的三个部分。通过使用 MVC 架构，可以降低模块之间的耦合，提供应用的可扩展性。MVC 中的每个组件只关心组件内的逻辑，不应与其他组件的逻辑混合。

Java EE 的出现，则更加规范了 B/S 结构应用的开发。Java EE 推荐将应用分为数据持久层、业务逻辑层和 Web 层，各层之间以松耦合的方式组织在一起。

1.1.2 传统 Web 应用的优势和缺点

B/S 结构虽然是一种非常优秀的结构，但它依然存在如下方面的不足：

- **独占式的请求**。比如一个任务需要多步骤或多选项才能完成。在 HTML 里，一个多步

骤的任务可以在单页内表达出来。但是由于 HTML 的互动性有限，便可能产生一份很长的页面，使用户感到混乱、笨拙而难以使用。或者将多个步骤分成几个页面分别提交，但传统的独占式请求表现为，如果前一个请求没有得到完全响应，则后一个请求不能发送。用户在等待服务器的响应期间，浏览器一片空白。这种独占式请求如图 1.3 所示。

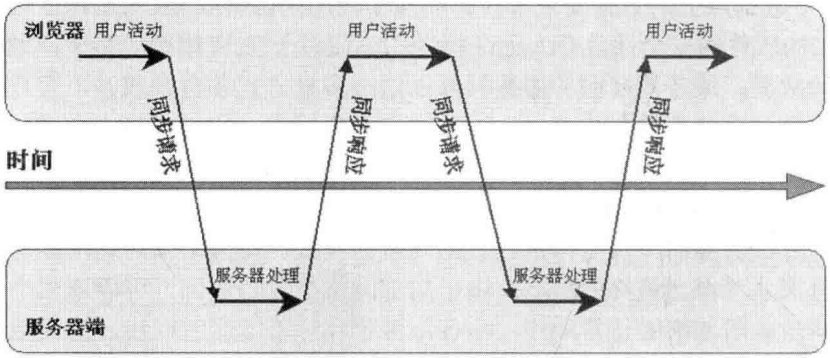


图 1.3　独占式请求的示意图

- **频繁的页面刷新。**传统的 Web 应用基本上采用请求－页面的对应模式，每个请求都需要丢弃当前页面来重新加载新页面。频繁的页面刷新不仅让客户处于不连续的体验中，也使服务器的负担加重。
- **简陋的页面。**传统 Web 应用因为需要频繁刷新页面，因而不可能制作出具有丰富表现力的页面。具有丰富表现力的页面导致页面文件过大，下载速度较慢，而且页面频繁刷新。一个表现丰富的页面下载需要相当多的时间，但随着请求的提交，又需要重新下载新页面，这样系统开销相当大。因而传统 Web 应用的页面不可能非常出色。

在传统的 Web 应用里，用户总是需要加载新页面时才能提交请求，而提交请求后又需要等待服务器响应，所以如果服务器响应还没有完全结束，则用户只能等待，不能继续发送请求。

在传统的 Web 应用里，每个请求对应一个页面，不管客户端以 POST 还是 GET 方式提交请求，每次请求都会丢弃当前页面，等待服务器生成新页面。在等待期间，旧的页面已经丢弃，新的页面还没有完全生成，整个浏览器将一片空白，而用户什么都做不了，只能等待。这对于用户而言，是一种不连续的体验，感觉非常不好。

1.2　重新设计 Web 应用

传统 Web 应用的不足，一直凸显在用户面前，用户常常抱怨系统的响应速度太慢。除了网络带宽的限制、业务逻辑复杂、硬件设备制约等因素外，频繁的页面刷新，每次响应都必须下载整个响应页面，也导致了响应速度慢。因此，传统 Web 应用必须重新改进。

1.2.1　富 Internet 应用

RIA，是 Rich Internet Application 的缩写，即富 Internet 应用。B/S 结构已成为应用程序开发的默认结构，用户对应用程序复杂性要求日增，Web 应用程序对完成复杂逻辑始终差强人意。

用户与复杂的 Web 应用程序交互时，体验并不能令人满意。Web 模型是基于 HTML 页面的模型，缺少客户端智能机制。传统的 Web 应用几乎无法完成复杂的用户交互（如传统的 C/S 应用程序和桌面应用程序中的用户交互）。因而，Web 在许多应用程序中难以发挥。

为了提升用户体验，出现了一种新类型的 Web 应用，那就是 RIA。这些应用程序吸收了桌面应用程序的反应快、交互性强的优点，改进了 Web 应用程序用户交互差的缺点，可以提供一种更丰富、更具有交互性和响应性的用户体验。

可以将 RIA 架构理解为运行于 B/S 结构上的 C/S 应用。应用客户端采用标准的浏览器，但在浏览器内支持类似 C/S 应用的操作，所以 RIA 应用可以提供强大的功能，让用户有高交互性、高效率响应的体验。同时，RIA 又是基于 Internet 浏览器的应用，所以，用户使用 RIA 非常方便。

使用 RIA，用户无须安装任何的客户端软件，只需拥有浏览器即可。一个典型的富客户端应用是地图。大部分 Web 地图支持鼠标的拖动和放大、缩小。地图随着鼠标的拖动而移动——明显加载了新的数据，但页面本身却无须重新加载。如果鼠标拖动得太远，可能出现部分空白区域，但这种空白只是因为地图区域在加载，而不是整个页面在加载。

当使用鼠标单击地图上的提示点时，地图上会出现该点的更详细的介绍。图 1.4 显示了高德地图中的广州地图。

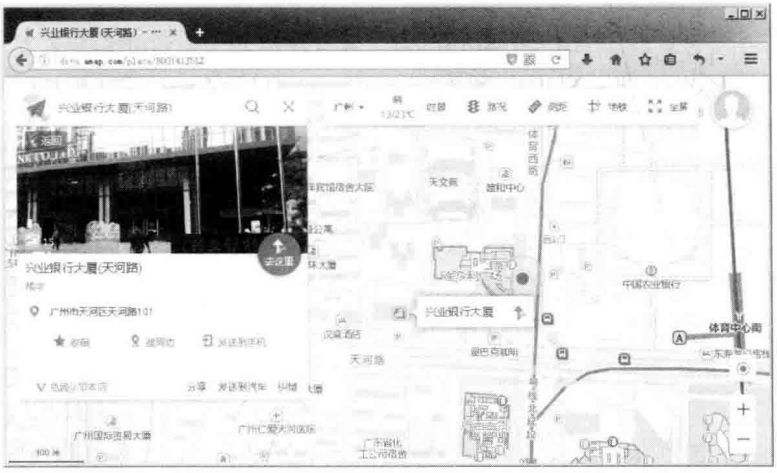


图 1.4 高德地图应用

RIA 代表着目前 Web 应用的发展趋势，RIA 应用相当于同时吸收了 C/S 结构和 B/S 结构优势后的产物：RIA 应用既能利用 B/S 应用中数据安全、系统更新方便的优势，也能利用 C/S 应用中界面美观、交互丰富的特点，因此 RIA 应用是目前 Web 应用的主流。

为了开发出 RIA 应用，仅依靠传统服务端编程是远远不够的，此时必须对传统的客户端表现界面进行改进——传统的客户端表现界面通常使用静态 HTML 页面来实现，这种静态 HTML 页面主要具有如下两个劣势：

- 无法与服务器实时地进行异步数据通信。
- 传统 HTML 页面界面比较丑陋，缺乏用户交互性。

1.2.2 改进的服务器通信

传统的 Web 应用亟须改变的就是浏览器与服务器的通信方式，传统的同步请求—响应的通信方式无法满足 RIA 应用的网络通信要求。

2005 年开始出现的 Ajax 正是对传统 Web 通信方式的改进，Ajax 使用 XMLHttpRequest 异步发送请求，XMLHttpRequest 发送请求不要求重新加载页面。

XMLHttpRequest 发送请求后，无须等待服务器响应，而是可以继续原来的操作。在服务

器完成响应后，客户端使用 JavaScript 通过 XMLHttpRequest 获取服务器响应。

图 1.5 显示了异步发送请求的示意图。

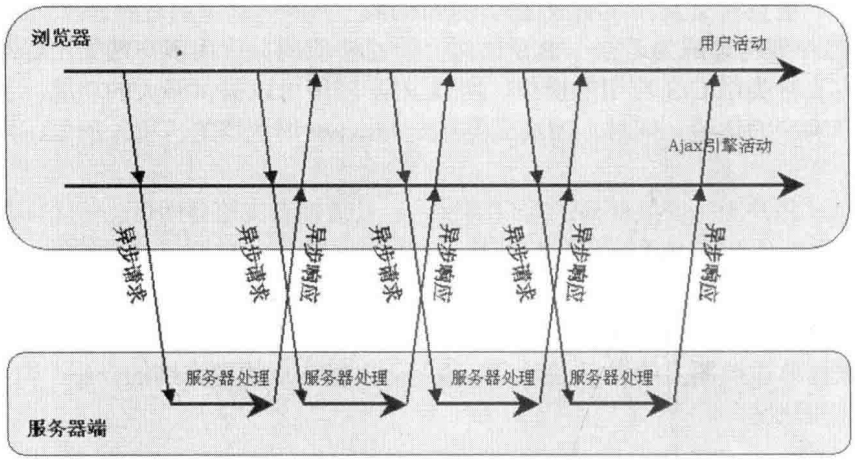


图 1.5　异步发送请求

Ajax 除了异步发送请求外，还能动态加载服务器响应数据。使用 Ajax 能避免频繁刷新页面，服务器响应的是数据，而不是整个页面内容。Ajax 负责获取服务器数据，然后将服务器数据动态加载到浏览器中。

此外，客户端与服务器端的通信还包括如下几种方式：

- **WebSocket 通信技术。**WebSocket 会在服务器与浏览器之间建立基于 TCP 协议的 Socket 连接，该连接一旦建立，浏览器可以实时地向服务器发送数据，服务器也可随时向浏览器发送数据。由此可见，WebSocket 可在浏览器和服务器之间进行双向的数据传输。重要的是，WebSocket 已经成为 HTML 5 技术规范之一，被所有主流浏览器所支持。
- **Server-sent Events 技术。**这是一种简单的服务器推送技术，这种推送技术只允许服务器将数据推送给浏览器，浏览器无法向服务器发送数据，因此这种技术规范也被称为服务器推送技术。与 WebSocket 相比，服务器推送技术更简单一些，但功能也弱一些。一般来说，WebSocket 适用于需要进行复杂双向数据通信的场景。对于简单的服务器数据推送场景，使用服务器推送事件就足够了。
- **COMET 技术。**COMET 技术采用的是长轮询的实现。这种实现方式在每次请求时，服务器端会保持该连接在一段时间内处于打开状态，而不是在响应完成之后就立即关闭。这样做的好处是在连接处于打开状态的时间段内，服务器端产生的数据更新可以被及时地返回给浏览器。当上一个长连接关闭之后，浏览器会立即打开一个新的长连接来继续请求。不过 COMET 技术的实现在服务器端和浏览器端都需要第三方库的支持。但遗憾的是，COMET 技术目前还没有成为 Web 规范。

1.2.3　丰富的客户端交互

除了改进的服务器端通信之外，现代 Web 应用还必须具有如下两个特征：

- 优雅、美观的用户界面。
- 丰富的客户端交互。

为了实现上述两个特点，HTML 5 做出了巨大的努力，HTML 5 添加了大量新的元素和属性，这些新元素和新属性极大地丰富了 Web 的前端开发，而且增强了用户交互性。此外，HTML