

.NET 开发经典名著



ASP.NET SignalR 编程实战

Professional



[美] José M. Aguilar 著
明道洋 译



清华大学出版社

.NET 开发经典名著

ASP.NET SignalR

编程实战

[美] José M. Aguilar 著

明道洋 译

清华大学出版社

北 京

Authorized translation from the English language edition, entitled SignalR Programming in Microsoft ASP.NET, 9780735683884 by Aguilar, Jose M., published by Pearson Education, Inc, publishing as Microsoft Press, Copyright © 2014 by Krasis Consulting S.L.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by TSINGHUA UNIVERSITY PRESS LIMITED, Copyright © 2015.

北京市版权局著作权合同登记号 图字: 01-2015-2180

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET SignalR 编程实战/(美) 阿吉拉(Aguilar, J. M.) 著; 明道洋 译. —北京: 清华大学出版社, 2015
(.NET 开发经典名著)

书名原文: SignalR Programming in Microsoft ASP.NET

ISBN 978-7-302-39479-2

I. ①A… II. ①阿… ②明… III. ①网页制作工具—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2015)第 036202 号

责任编辑: 王 军 韩宏志

装帧设计: 孔祥峰

责任校对: 曹 阳

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×230mm

印 张: 15.5

字 数: 337 千字

版 次: 2015 年 4 月第 1 版

印 次: 2015 年 4 月第 1 次印刷

印 数: 1~3000

定 价: 49.80 元

产品编号: 060677-01

译者序

直到目前，可用于实时 Web 应用开发的框架有很多，如著名的 Comet、WebSocket、Flash Socket 等。也许以持久连接(Persistent Connection)和集线器(Hub)为主要特征的 SignalR 只能算是后起之秀！但作为微软 ASP.NET 下的一个类库，SignalR “定时事件触发，异步输入输出”的理念以及和 ASP.NET 技术的无缝集成等特性，正逐步得到无数 ASP.NET 技术开发者的青睐。

SignalR 已经和现有 ASP.NET 框架进行了集成，对服务器和客户端代码都做了很好的抽象，这意味着不需要了解很多的底层协议细节；当客户端浏览器不支持 WebSocket 时，它可以智能地使用其他通信机制；此外，它能将消息广播给所有客户端，也可以给特定客户端发送消息，并发连接的处理也非常优秀；最后，它居然是开源的！

相对于一些其他的实时 Web 应用开发技术，可能有人会说 SignalR 的最大的特色莫过于它是微软出品。诚然，要看到更远的风景，就要站在“巨人”的肩膀上……

本书共分为 9 章，是第一本全面阐述 ASP.NET SignalR 技术的书籍。主要探讨 SignalR 实时多用户系统开发的不同方面。本书将从零开始，逐步深入该框架的一些高级功能的实现。书中很多内容都结合了大量的开发实例，非常适合那些正打算步入实时 Web 应用开发“殿堂”的初学者，有一定经验的 Web 开发人员也可以将本书用作参考。

感谢清华大学出版社的编辑们，他们在本书的翻译过程提出了很多宝贵意见，他们的帮助确保了本书的顺利翻译和出版。本书全部章节由明道洋翻译，参与翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。在此一并表示感谢！

翻译是艰苦的工作，虽然译者始终保持着一一种审慎的态度，但囿于水平，译文中的错误和失误在所难免，如有任何意见和建议，请不吝指正。

前 言

SignalR 是微软对 Web 开发技术栈的扩充。它是一种框架，可方便地用来创建神奇的实时应用程序，如在线协作工具、多用户游戏、实时信息服务等，诸如此类的应用程序以往开发起来相当复杂。

本书不仅将从头开始完整展示 SignalR 开发有关的技术，也将处理一些高级主题。读完本书后，你应该对该框架有所了解，并能成功地将这些知识付诸于实践，创建任意规模的实时系统。本书也可以作为参考手册，虽然它并不全面，但包含了 SignalR 开发中的大多数实际应用以及完全掌握 SignalR 的一些必需基础知识。

本书读者对象

本书旨在帮助开发人员了解和编写基于 SignalR 的组件或应用程序。如果需要在现有应用程序中实现实时立即性，或是需要基于这种模式创建新系统，那么你可能对本书特别感兴趣。

专门从事后端的开发人员可以了解如何实现供任何客户端使用的实时服务，并解决那些需要可扩展性或需要通过单元测试进行质量改进的情形。那些青睐于前端的开发人员将看到他们如何使用实时服务，在创建客户端时添加一些丰富多彩的特性。由于推送(push)的使用和这些解决方案的异步性，Web 开发人员将找到一种真正简单的方法来突破 HTTP 领域的局限性。

阅读本书的前提条件

本书假定读者已经熟悉 C#和.NET 环境下的一般编程知识。同时，由于 SignalR 自身以及书中的一些示例和内容大部分都和 Web 世界有关，因此有必要了解 Web 所依赖的协议，同时要具备一定的 Web 编程语言基本知识，如 HTML，尤其是 JavaScript。

虽然并非严格要求，但读者可以从 jQuery、Windows Phone 8 开发或 WinRT 有关的一些必备知识中受益，因为它们和书中的一些开发示例和内容都有关系。熟悉单元测试、模拟、依赖注入(dependency injection)对掌握书中最后一些章节的内容同样也非常有帮助。

哪些人不应该阅读本书

对 .NET 平台和 C# 一窍不通的人将不能受益于本书。如果不具备 JavaScript 的一些必备知识，将很难理解书中的一些解释。

本书结构

本书分为 9 章，主要探讨 SignalR 实时多用户系统开发的不同方面。本书将从零开始，带你逐步深入了解该框架的一些高级功能的实现。

第 1 章“Internet、异步、多用户！”和第 2 章“HTTP：客户端即上帝”纯粹是介绍性的，这两章有助于我们理解一些技术背景以及这个新框架所依赖的一些基础。

在第 3 章“SignalR 概述”中，我们将在一个较高的层次上对 SignalR 进行介绍，展示它在 Microsoft Web 开发栈中的位置以及其他一些相关概念，如 OWIN 和 Katana。

从第 3 章开始，我们将对 SignalR 应用程序开发的一些方法进行详细介绍。第 4 章“持久连接”和第 5 章“Hub”将专门探讨如何使用持久连接和 Hub 进行不同抽象层级的开发。第 6 章“持久连接和来自其他线程的 Hub”将探讨如何在同一个应用程序中整合这些组件和其他一些技术。第 7 章“实时多平台应用程序”主要探讨如何实现多平台客户端。

第 8 章“部署和扩展 SignalR”将展示 SignalR 提供的不同部署情形和扩展方案，第 9 章“高级主题”将对其他一些更高级的方面进行介绍，如安全、可扩展性和测试等。

在本书中找到你最好的起点

本书采用的是由浅入深的组织方式，尽管你可以从头至尾地阅读，但也可以将本书用作参考书，直接对特定的内容进行查找，这取决于读者原有的知识水平以及个人的需要。

对于第一次接触 SignalR 的开发人员而言，我们建议按照本书章节的顺序，从头至尾地阅读本书。但对于那些精通 SignalR，使用过所有版本的开发人员而言，快速浏览一下前三章就足够了，然后只关注和持久连接或 Hub 有关的一些章节，看看有什么不清楚的地方或是和以前版本相比有什么变化。可以直接从那里转到特定的区域来解决疑问，例如，框架的可扩展特性、实现授权机制或是在 Hub 上执行单元测试的一些过程。

对于任何章节而言，下载并安装相关的示例项目是不错的想法，这将有助于对书中的一些内容进行巩固和练习。

本书的一些约定和特点

为了让本书更具可读性，并且易于理解，我们采用以下约定来呈现内容：

- “注意”特殊段落将提供一些额外的信息，或是用来提供成功完成某项任务的可选方法。
- 两个按键名称之间的加号(+)表示必须同时按下这些键。例如，“按 Alt+Tab”表示在按下 Tab 键的同时必须按下 Alt 键。
- 两个或多个菜单项之间的竖线(例如，“File | Close”)表示应该选择第一个菜单或菜单项，然后是下一个，等等。

系统需求

为能更好地遵照书中的一些示例进行练习，至少应具备下列所示的一些硬件和软件项：

- 配备至少 1.6GHz 处理器的计算机(推荐 2GHz)。
- 2GB RAM(建议 4GB)。
- 兼容 DirectX 9 的显卡，分辨率高于 1024×768。
- 操作系统 Windows 7 SP1、Windows 8 、Windows 8.1 或高于 Windows Server 2008 R2 SP1 的版本。
- Internet Explorer 10。
- Visual Studio 2012 或以上任何更新版本，大多数情况下使用的是 Express 版本。
- Internet 连接。

一些示例可能需要一个具有管理员权限的系统账号，用来安装像 Windows Phone SDK 这样的组件。一些章节中也将用到一些像 Windows Azure 服务这样的外部资源。

示例代码

为了对一些概念进行说明，书中列举了很多示例甚至是完整的项目，其中大多数示例以及其他一些示例可从下列地址进行下载：

<http://aka.ms/SignalRProg/files>

请按照说明下载 SignalRProgramming_codesamples.zip 文件。

另外，也可从 <http://www.tupwk.com.cn/downpage> 下载示例代码。

注意：

除了这些示例代码外，你的系统上应该已经安装了 Visual Studio 2012 或 2013。

版本说明

本书写作时使用的 SignalR 版本为 2.0.0，可在其中找到特定版本的各种参考资料。

微软的 SignalR 团队不断努力改善其产品，所以经常会发布一些软件更新。这些更新的版本编号通常为 2.0.x 或 2.x.0，除修订外，这些更新可能包含了一些新的或经过完善的特性，但并非一些重大的更改或是对开发 API 进行的显著修改。

无论在什么时候更新这些新版本的组件，显然都有必要对示例源代码中的引用进行修改，特别是在引用了脚本库函数时。

因此，如果有以下代码：

```
<script src="/scripts/jquery.signalR-2.0.0.min.js"></script>
```

安装 SignalR 2.0.1 版本后，应该将其修改成下面这行：

```
<script src="/scripts/jquery.signalR-2.0.1.min.js"></script>
```

安装示例代码

为安装示例代码，只需按照上文所述下载压缩包文件，然后将其解压到系统上的某个文件夹中。

使用示例代码

解压该文件后，应该已经创建了文件夹结构，文件夹将按照本书的章节顺序进行组织，从第 4 章开始，我们将对一些示例代码进行分析。

Chapter 04 - Persistent connections

Chapter 05 - Hubs

Chapter 06 - External access

...

每一个文件夹中都可以找到一个针对每个示例项目的子文件夹，这些子文件夹将按照书中讲解概念的顺序进行编号。

...

Chapter 08 – Scaling

1-AzureServiceBus

2-SqlServer

...

可从这些文件夹中找到针对每个示例的具体解决方案文件(*.sln)，尽管某些情况下需要对配置进行预先调整，但这些解决方案彼此之间并不相互依赖，并且都包含一个可以运行的、功能齐全的示例(在 Visual Studio 中按 F5 功能键)。此类情况下，项目主页上或 readme.txt 文件中通常会给出有关的详细说明。

致谢

有关的致谢的言辞可能听起来会很老套。这样的一本书，没有很多人的通力协作是不可能完成的，正是他们的辛勤付出才使本书成为现实。这里谨向他们表示谢意！

特别要感谢我的编辑——来自 campusMVP.net 的 Jose M. Alarcón(其 Twitter 为@jm_alarcon)的参与。他在项目管理、修订方面的才华以及他所提出来的一些合理建议使本书的出版成为可能。

Javier Suárez Ruíz(@jsuarezruiz)为本书付出很多，特别是 Windows Phone 或 WinRT 等非 Web 环境下 SignalR 客户端示例的实现，因此他的参与同样是不可或缺的。

我要感谢 campusMVP.net 的 Victor Vallejo，感谢他在文本上的宝贵帮助。

至于微软方面，要感谢策划编辑 Devon Musgrave，他从一开始就对此项目非常感兴趣，没有他的帮助，绝不会有此书。也要感谢项目编辑 Carol Dillingham 专业的工作。感谢技术复审 Todd Meister、文字编辑 Richard Carey、nSight 项目管理 Sarah Vostok、索引编辑 Lucie Haskins。感谢 O'Reilly Media 的 Sarah Hake 和 Jenna Boyd 的支持。

最后要感谢 Damian Edwards 和 David Fowler，感谢他们宝贵的输入。为了能够让本书尽可能贴近实用，很荣幸能够受益于 SignalR 创造者的一些建议和贡献。

勘误表和本书支持

我们已经尽了最大努力来确保本书及附加内容的准确性。本书出版后，读者报告的任何错误将通过下列地址列出：

<http://aka.ms/SignalRProg/errata>

如果你发现某个错误没有列出，可通过上面的页面将错误告知我们。

如果需要其他支持，可以给Microsoft Press Book Support发送电子邮件(mspinput@microsoft.com)。

请注意 Microsoft 软件产品支持采用的并非上述地址。

如果对本书中文版有任何意见或建议，请发送邮件到 wkservice@vip.163.com。

我们想听到你的想法

对于 Microsoft 出版社而言，客户的满意是我们至高无上的宗旨，你的反馈是我们最宝贵的财富。请将你对本书的想法告知我们：

<http://aka.ms/tellpress>

调查很简单，我们将认真阅读每一个意见和想法，首先感谢你的反馈。

联系我们

让我们继续保持沟通吧。我们在 Twitter 上：<http://twitter.com/MicrosoftPress>。

目 录

第 1 章	Internet、异步、多用户	1
第 2 章	HTTP: 客户端即上帝	5
2.1	HTTP 操作	5
2.2	轮询	7
2.3	推送: 服务器采取主动	8
2.3.1	WebSocket	9
2.3.2	Server-Sent Events(API Event Source)	10
2.3.3	如今的推送方式	11
2.3.4	我们需要的不仅是推送	14
第 3 章	SignalR 概述	17
3.1	SignalR 的作用	18
3.2	两个抽象层	19
3.3	支撑平台	20
3.4	OWIN 和 Katana: 新来者	21
3.5	安装 SignalR	24
第 4 章	持久连接	27
4.1	服务器端上的实现	28
4.1.1	映射并配置持久连接	28
4.1.2	持久连接事件	30
4.1.3	给客户端发送消息	32
4.1.4	异步事件处理	35
4.1.5	连接组	35
4.1.6	OWIN Startup 类	37
4.2	客户端实现	39
4.2.1	使用 JavaScript 客户端初始化连接	39

4.2.2	支持老式浏览器	41
4.2.3	支持跨域连接	41
4.2.4	发送消息	43
4.2.5	接收消息	45
4.2.6	给服务器发送其他信息	46
4.2.7	客户端上的其他可用事件	47
4.3	传输协商	48
4.4	调整 SignalR 配置参数	49
4.5	完整示例: 跟踪访问者	51
4.5.1	项目创建和设置	52
4.5.2	客户端实现	52
4.5.3	服务器端实现(TrackerConnection.cs)	54
第 5 章	Hub	55
5.1	服务器实现	56
5.1.1	Hub 的注册和配置	56
5.1.2	创建 Hub	57
5.1.3	接收消息	58
5.1.4	给客户端发送消息	62
5.1.5	给特定用户发送消息	66
5.1.6	状态维护	68
5.1.7	访问与请求上下文有关的信息	69
5.1.8	连接和断开连接通知	70
5.1.9	管理组	71
5.1.10	维护服务器状态	71
5.2	客户端实现	76
5.2.1	JavaScript 客户端	76
5.2.2	生成代理	77
5.2.3	手动生成 JavaScript 代理	79
5.2.4	建立连接	81
5.2.5	给服务器发送消息	83
5.2.6	发送附加信息	87
5.2.7	接收服务器发来的消息	88
5.2.8	日志记录	89

5.2.9	状态维护	90
5.2.10	不具有代理的客户端实现	91
5.3	完整示例: 共享画板	94
5.3.1	项目创建和设置	95
5.3.2	客户端上的实现	95
5.3.3	服务端上的实现	98
第 6 章	持久连接和其他线程的 Hub	101
6.1	从其他线程访问	101
6.1.1	使用持久连接从外部进行访问	102
6.1.2	完整示例: 在服务器上监视连接	104
6.1.3	项目创建和设置	105
6.1.4	站点实现	105
6.1.5	用于跟踪请求的系统(服务器端)	106
6.1.6	用于跟踪请求的系统(客户端)	109
6.2	使用 Hub 进行外部访问	109
6.2.1	完整示例: 进度条	111
6.2.2	项目创建和设置	111
6.2.3	客户端实现	111
6.2.4	服务器端实现	113
第 7 章	实时多平台应用程序	115
7.1	多平台 SignalR 服务器	115
7.1.1	托管在非 Web 应用程序中的 SignalR	115
7.1.2	非 Windows 平台中的 SignalR 托管	124
7.2	多平台 SignalR 客户端	127
7.2.1	从 .NET 非 Web 客户端访问服务	128
7.2.2	从其他平台访问服务	148
第 8 章	部署和扩展 SignalR	151
8.1	增长的烦恼	152
8.2	SignalR 的扩展性	155
8.3	底板上的扩展	158
8.3.1	Windows Azure Service Bus	159
8.3.2	SQL Server	163

8.3.3 Redis	166
8.4 自定义底板	169
8.5 提高 SignalR 服务的性能	172
8.5.1 服务器配置	174
8.5.2 监视性能	175
第 9 章 高级主题	179
9.1 SignalR 中的授权	179
9.1.1 持久连接中的访问控制	179
9.1.2 Hub 中的访问控制	180
9.1.3 客户端身份验证	182
9.2 可扩展的框架	189
9.3 依赖注入	195
9.3.1 手动依赖注入	197
9.3.2 释放依赖项	199
9.3.3 反转控制容器	199
9.4 对 SignalR 进行单元测试	204
9.4.1 对 Hub 进行单元测试	210
9.4.2 对持久连接进行单元测试	216
9.5 在 Hub 中截获消息	219
9.6 与其他框架的集成	224
9.6.1 Web API	224
9.6.2 ASP.NET MVC	227
9.6.3 Knockout	228
9.6.4 AngularJS	231

第 1 章

Internet、异步、多用户

人们对于同时结合了 Internet、异步、多用户合作与交互的应用大加赞赏。有时，我们的确应该为一些现代 Web 系统所能提供的交互性而拍手叫好，如 Facebook、Twitter、Gmail、Google Docs 以及 Office Web 应用等。这些 Web 应用几乎不需要重载页面就可以实时接收更新。

例如，当我们使用 Office Web 应用程序编辑某个文档时，其他用户也能够同时访问该文档。我们可以看到这些用户已经进入该文档，正在对文档进行修改。即便是较常见的情形，例如，简单的网页聊天——我们的朋友刚刚输入的消息像施了魔法一样马上就出现了。这两种系统都使用了同样的解决方案：在服务器和客户端之间实时传输异步数据。

有经验的 Web 开发人员习惯于一些通过协议进行控制的传统方法，即客户端是进行信息请求的活动代理(active agent)，服务器仅进行响应。这是我们看好这种应用的前景的潜在原因：客户端直接通过服务器进行更新，例如，由于某个新用户已经进入并编辑文档，或是我们的聊天室好友在聊天的过程中编写了一条新的消息。纯粹的魔法！

毫无疑问，这个世界确实需要这种即时性。用户需要立即知道他周围环境中发生了什么、他们正在处理的文档、他们的社交网络、他们的在线游戏——日常生活中这样的例子越来越多，用户不用再像几年前那样四处搜寻信息，他们现在希望信息能够马上在眼前产生。

对于 Web 协议层次而言，此类需求已经非常明显。由于 HTTP 存在已久，而且仍保持着其定义之初的形式，已无法有效满足这些新的需求。事实上，定义 Web 标准和协议的这些组织、浏览器开发商们也意识到了这一点，他们正极力寻求一种新的机制。他们沿着相反的方向，一反传统，寻求客户端和服务器之间新的通信机制，即允许服务器在通信中采取主动。

这已经成为现实，不仅产生了新的协议，而且这些协议使用起来还具有某种程度的可靠性；不过，仍未找到统一的解决方案。客户端和服务平台之间，甚至连网络基础设施之间都存在着很多差异，因此这些新机制的普及不仅困难而且缓慢。稍后，我们将就这些方面进行详细探讨。

虽然如此，当我们开发实时多用户应用程序时，这并不是唯一要解决的问题。正如我们所知，网络通信中包含了很多不可靠、不可预测的变量，这使得信息的管理以及针对用户的

分发相当复杂。例如，在聊天室应用程序中，连接到同一房间的用户可能具有不同的带宽，并且在聊天过程中这些带宽还可能有所波动。为防止在这种情况下丢失信息，服务器应该能够暂存这些信息，将它们发送给接收者，并监视哪些用户已经收到。这总要考虑每个用户的通信状况，以及传输过程中可能产生的潜在故障。如有必要，则重发数据。从某种意义上讲，除了实时系统所需的即时性外，这和传统的 FTP 服务器还是很类似的。实现这样的一种系统，其复杂度和难度还是不难预测的。

就该实时应用程序实现过程所存在的问题而言，直到最近都没有一种完整的解决方案，微软公司也没有提供和 .NET 有关的一些组件或框架。当然，有很多技术能够提供面向连接的和面向非连接的服务，例如，我们熟悉的 Web Service、WCF 或是较新的 Web API。然而，它们并非专门为多用户之间进行实时协作的异步环境而设计。事实上，尽管有可能使用这种平台创建这样的系统，但甚至连最有经验的开发人员实现起来都很难。即使能够实现，最后产生的系统往往也非常低效，存在很多性能问题，扩展性也不是很好。

接下来，我们将了解如何通过 SignalR 实现这种最“吸引眼球”的特性。SignalR 是一个了不起的框架，它功能强大，具有较好的灵活性和可扩展性，可用来简化我们的开发任务。

为此，我们将首先简要了解一下开发实时多用户应用程序面临的一些问题，其中一些上文已经有所涉及。我们将大致了解一下开发 HTTP 操作的工作机制，以及它在实时应用和开发方面所存在的一些局限。我们将了解推送(push)这一概念，同时也将了解 W3C 和 IETF 目前正制定的一些标准，以及目前可用来在 HTTP 上实现推送(push)的一些技术。这些内容将有助于我们深入理解实际应用中的一些情形，迎接即时性和交互性应用程序开发所面临的一些挑战。反之，这些内容也有助于我们更好地理解 SignalR 的工作机制及其背后的原理。

接下来，将对 SignalR 进行正式介绍，包括它主要的功能、它在 Microsoft Web 开发技术栈中的位置以及不同的抽象层级(允许跨越底层协议，有助于和底层细节保持隔离，专注于为用户开发更加丰富多彩的特性)。这部分内容也将涉及 OWIN 和 Katana——它们是两种新的代理，在包括 SignalR 在内的各种技术中越来越引人注目。

我们将深入学习该框架所提供的各种技术和抽象概念，充分利用其强大的功能以及灵活性，创建我们自己的交互式多用户实时应用程序(无论是服务器端还是客户端)。当然，为了有助于真正理解其基础，本书提供了很多示例代码，这些示例代码将向我们展示如何在真实项目中使用此框架。

我们也将对 SignalR 如何独立于 Web 环境进行介绍：尽管 SignalR 看似构成了自己的独立环境，但此框架远不止此，它允许任何类型的应用程序提供实时服务，并且几乎适用于任何系统。书中列举了几个这样的示例。

另一个非常重要的方面是 SignalR 应用程序的部署和扩展，本书将用一定的篇幅专门对此进行探讨。我们将了解该平台自带的几个非常便捷的工具。如果这些工具还不足以解决的问题，这里还有其他几个可能的解决方案。此外，我们还将了解几个用来监视服务器状态的工具，并在高并发环境中提高服务器性能。

最后，我们将步入 SignalR 高级编程阶段，深入理解该框架的工作机制，包括安全、使用依赖注入创建解耦组件、SignalR 可扩展性、单元测试等。

欢迎来到多用户实时异步应用程序。欢迎来到 SignalR !