**Short Description**

Learn how to leverage the features and capabilities of the ASP.NET Web API to build a RESTful web

service from start to finish. This book will first explain the REST architectural style, and then build on that knowledge, the ASP.NET Web API, and some helpful patterns, tricks, and techniques, to go from a blank slate to a fully functional, secure REST service.

**Book Description**

The ASP.NET MVC Framework has always been a good platform on which to implement REST-based  
services, but the introduction of the ASP.NET Web API Framework raised the bar to a whole new level.  
Now in release version 2.1, the Web API Framework has evolved into a powerful and refreshingly usable platform. This concise book provides technical background and guidance that will enable you to best use the ASP.NET Web API 2 Framework to build world-class REST services.

New content in this edition includes:  
• New capabilities in Web API 2 (currently version 2.1).  
• Support for partial updates, or PATCH.  
• API versioning.  
• Support for legacy SOAP-based operations.  
• How to handle non-resource APIs using REST  
• How to best expose relationships between resources  
• JSON Web Tokens, CORS, CSRF

Get ready for authors Jamie Kurtz and Brian Wortman to take you from zero to REST service hero in  
no time at all. No prior experience with ASP.NET Web API is required; all Web API-related concepts are introduced from basic principles and developed to the point where you can use them in a production system. A good working knowledge of C# and the .NET Framework are the only prerequisites to best benefit from this book.

**What You’ll Learn**

* Introduction to the REST architecture
* How to design a REST API
* New capabilities in ASP.NET Web API 2
* Understanding ASP.NET Web API controller activation
* Automatic lifetime management for database connections and transactions
* Using NHibernate with ASP.NET Web API
* Easily secure a REST service, using standards-based authentication and authorization and JSON Web Tokens
* Supporting legacy SOAP callers with ASP.NET Web API
* How to expose relationships between resources
* Supporting partial resource updates under REST
* Web API versioning