

Programming .NET Web Services

book review by Bob Swart

Programming .NET Web Services by Alex Ferrara & Matthew MacDonald, pub. O'Reilly, price: £28.50 (£22.78 + p&p from Amazon via the BUG website), 396 pages.

This book was published in September 2002 (so it's brand new), and is very pleasant to read - it took me just over a week to read it from cover to cover (but I must admit that I already had some knowledge and experience with web services, so some chapters were easier to read than others).

The book shows how to create web services in the .NET Framework with ASP.NET, but does not spend time on the (ASP).NET fundamentals or C# language syntax, as some other books on .NET web services sometimes do.

The book consists of 11 chapters and two appendices. Not all of them equally interesting, but pretty complete. The first chapter is more a little history overview that puts web services into perspective. Interesting to read (even for managers). It also discusses related techniques such as SOAP, WSDL and UDDI, and how they work in .NET.

The second chapter is the place where we start to build our first web services in C# and ASP.NET. For the first example, you don't even need Visual Studio .NET, only the .NET Framework (and a text editor, but even notepad will do). Later examples show how we can use Visual Studio .NET to build as well as deploy the web service engines. The next chapter (3) is focused on consuming web services, explaining the publish/find/bind model, and then continuing in detail on the binding process. Again, apart from using Visual Studio .NET, the chapter also shows how we can consume web services without it (using wsdl.exe and csc.exe).

After these basics, chapter 4 continues with data types in web services, from simple .NET primitive data types to more complex data types, including .NET DataSets.

Chapter 5 explains how we can manage state in web services, starting with the great state debate (the three options developers face when state is required), and a discussion on stateless vs. stateful design. The chapter then describes ASP.NET's session state facility for use in web services, and finally the use of SOAP Headers.

Chapter 6 is all about asynchronous services - where a client can make a call or request to a web service engine, and returns to pick it up at a later time. This ensures that the client (consumer) does not have to wait (or is blocked) while the call is executed. The chapter shows how to implement this in different ways.

The next two chapters (7 and 8) are about caching, profiling and debugging web services. These chapters can be used to learn how to optimise and debug your web services using ASP.NET and Visual Studio .NET. For logging purposes, we'll even see how to implement SOAP extensions.

Chapter 9 is about security and authentication. It starts by listing a number of potential vulnerabilities, such as the network, operating system, web server, ASP.NET platform



and finally the web service application itself. The focus is then turned to the last three: from IIS to ASP.NET and our own application (using a ticket system).

Chapter 10 is an interesting one, about publishing and discovery. This is where UDDI (universal description, discovery and integration) and tModels come in. In my view UDDI is best used in an intranet environment, but this book also believes in UDDI for the internet. Time will tell, but it's good to learn about UDDI anyway.

Chapter 11, finally, is about interoperability of web services implemented in different environments, and using different SOAP message formats (like document style versus RPC style). The SOAP interoperability test - consisting of eleven echo methods - is also covered, but it doesn't go into some of the more complex issues such as interoperability of .NET DataSets. Unfortunately, only a few SOAP toolkits are actually mentioned, so we have to use the references and URLs mentioned in the chapter to get more information about the SOAP interoperability between Visual Studio .NET and - for example - Borland SOAP tools such as Delphi or JBuilder (or plain Java for that matter).

Appendix A contains a quick reference for the System.Web.Services namespace, and Appendix B contains a list of descriptions of some of the more commonly used web service technologies. Finally, there's a thirteen page index, which I have not had to use (yet), since the chapters themselves are very clear about their content, so the table of contents may be a better way to find a specific topic.

A strong point of the book is that we do not always have to use the full-blown Visual Studio .NET environment to follow along, although some more advanced topics don't hesitate to show the detailed screenshots and steps in Visual Studio.NET. C# is used as programming language throughout the book, which is also positive in my view (at least there is no VB.NET in there, although the sample covers at Amazon.com and O'Reilly's own website have the by-line "Building .NET Web Services with C# and VB.NET", my official copy of the book has the modified by-line "Building Web Services with ASP.NET and C#"). The only (minor) downside is that the book never tries to go beyond the .NET environment to show that web services are cross-language and cross-platform, but it's up to the reader to experiment and find out just how interoperable the .NET web services will be for your needs.

For BUG members who have been introduced to web services using Delphi only, this book will contain some familiar information, but also a lot of additional techniques and ways to build web services for .NET specifically (I especially found the detailed configuration instructions for ASP.NET very helpful). If you're serious about web services for .NET, and already know a little bit of C# or ASP.NET, then I can recommend this book to you without any hesitation.