

Cross-Platform Mobile Apps Without the Kludge



# Cracking Windows Phone and BlackBerry Native Development

**Matthew Baxter-Reynolds**

Apress®

# **Cracking Windows Phone and BlackBerry Native Development**

**Cross-Platform Mobile Apps  
Without the Kludge**



**Matthew Baxter-Reynolds**

Apress®

## **Cracking Windows Phone and BlackBerry Native Development: Cross-Platform Mobile Apps Without the Kludge**

Copyright © 2011 by Matthew Baxter-Reynolds

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN 978-1-4302- 3374-9

ISBN 978-1-4302- 3375-6 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning

Lead Editor: Jonathan Hassell

Technical Reviewer: Matthew Fitchett

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick,

Jonathan Hassell, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey

Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt

Wade, Tom Welsh

Coordinating Editor: Anita Castro

Copy Editor: Mary Ann Fugate

Compositor: Bytheway Publishing Services

Indexer: BIM Indexing & Proofreading Services

Artist: April Milne

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit [www.springeronline.com](http://www.springeronline.com).

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at [www.multimobileddevelopment.com](http://www.multimobileddevelopment.com).

*For Martha and Elliott*

# Contents at a Glance

■ About the Author.....	xi
■ About the Technical Reviewer .....	xii
■ Acknowledgments .....	xiii
■ Chapter 1: Introduction.....	1
■ Chapter 2: The Six Bookmarks Server Service .....	7
■ Chapter 3: Application Architecture and Functional Specification .....	21
■ Chapter 4: Windows Phone 7: Installing the Toolset .....	39
■ Chapter 5: Windows Phone 7: Building the Logon Form and Consuming REST Services .....	49
■ Chapter 6: Windows Phone 7: Persisting Bookmarks Locally .....	83
■ Chapter 7: Windows Phone 7: Pushing Changes Back to the Server .....	131
■ Chapter 8: Six Bookmarks on Windows Mobile .....	163
■ Chapter 9: BlackBerry: Installing the Toolset .....	201
■ Chapter 10: BlackBerry: Building the Logon Form and Consuming REST Services .....	217
■ Chapter 11: BlackBerry: An ORM Layer on SQLite .....	243
■ Chapter 12: BlackBerry: Pushing Changes to the Server .....	297
■ Index .....	339

# Contents

■ <b>About the Author</b> .....	<b>xi</b>
■ <b>About the Technical Reviewer</b> .....	<b>xii</b>
■ <b>Acknowledgments</b> .....	<b>xiii</b>
■ <b>Chapter 1: Introduction</b> .....	<b>1</b>
What's the Purpose of This Book? .....	2
How Is This Book Structured? .....	3
Where Can You Get Help and Support? .....	5
Conclusion .....	5
■ <b>Chapter 2: The Six Bookmarks Server Service</b> .....	<b>7</b>
Creating an API Account .....	7
Creating a User .....	8
The Users Service .....	10
RESTful Web Services .....	10
Testing the Calls .....	10
Examining Logon Operations .....	11
The Bookmarks Service .....	14
Adding Some Test Data .....	14
Working with OData.....	15
OData Queries.....	19
Issuing Updates over OData .....	20
Constraining Data to the Logged-On User .....	20
Conclusion .....	20

■ <b>Chapter 3: Application Architecture and Functional Specification .....</b>	<b>21</b>
A Word About Tablets .....	21
Functional Specification .....	21
Logging On .....	22
Synchronizing .....	23
Navigator .....	23
Configuring Bookmarks .....	24
Configuring a Single Bookmark (“Configure Singleton”) .....	25
Missing Functionality .....	26
Application Architecture and Technical Specification .....	27
Approach .....	27
Object-Relational Mapping .....	28
Server Communication .....	32
Technical Approach Broken Down by Platform .....	34
Conclusion .....	37
■ <b>Chapter 4: Windows Phone 7: Installing the Toolset .....</b>	<b>39</b>
Silverlight vs. XNA .....	39
Visual Studio 2010 .....	39
Installing the Windows Phone Developer Tools .....	39
Creating a Test Application .....	40
Conclusion .....	47
■ <b>Chapter 5: Windows Phone 7: Building the Logon Form and Consuming REST Services .....</b>	<b>49</b>
Creating the Project .....	49
Building the Logon Form .....	50
Running the Logon Form .....	53
Calling the Services .....	55
Testing HttpHelper .....	59

Special Note on MessageBox .....	63
<b>Building Out the API and User Service Proxies .....</b>	<b>63</b>
Implementing the Authenticate Method .....	70
Making the Logon Button Work .....	73
Implementing “Remember Me” .....	77
<b>Conclusion .....</b>	<b>82</b>
<b>■ Chapter 6: Windows Phone 7: Persisting Bookmarks Locally .....</b>	<b>83</b>
Approach .....	83
Entities .....	83
The EntityType Class .....	84
The Entity Class .....	88
Setting Values in an Entity .....	89
Building Bookmark .....	94
Creating SBEEntityType Instances .....	96
Displaying Some Fake Bookmarks .....	97
Creating Fake Bookmarks .....	99
Showing and Populating the View .....	100
Implementing Button Clicks .....	104
Tombstoning—A Brief Word .....	107
Storing Entities in the Data Box .....	107
Data Box Basics .....	107
Creating the Entity Change Processor .....	111
Building the Sync Class .....	113
Selecting Entities .....	119
Implementing Delete All on the Sync .....	126
Using the ORM Subsystem to Support Tombstoning .....	126
<b>Conclusion .....</b>	<b>130</b>



■ <b>Chapter 7: Windows Phone 7: Pushing Changes Back to the Server</b> .....	<b>131</b>
The Configuration User Interface .....	131
Building ConfigurePage.xaml .....	131
Binding the List.....	134
Configuring the Application Bar.....	137
The ConfigureSingletonPage Page .....	142
Sending Changes to the Server .....	149
Building the PushUpdates Method .....	149
Issuing Server Requests to Insert, Update, and Delete .....	154
Building PushNextWorkItem .....	154
Update via HTTP MERGE and Insert via HTTP POST .....	155
Marking Fields As Being Available on the Server .....	155
Conclusion .....	162
■ <b>Chapter 8: Six Bookmarks on Windows Mobile</b> .....	<b>163</b>
Migrating/Porting.....	163
The Toolset .....	164
Hello, World .....	164
Six Bookmarks.....	164
How the Windows Mobile Application Was Built .....	165
Logging On and Calling RESTful Services.....	165
Synchronization 1: Downloading Bookmarks .....	171
Using SQL CE .....	176
Navigator .....	191
Configuring Bookmarks .....	193
Synchronization 2: Uploading Changes .....	195
Downloading Assemblies/Custom Bootstrappers.....	199
Conclusion .....	199

■ <b>Chapter 9: BlackBerry: Installing the Toolset .....</b>	<b>201</b>
A Word About Versions .....	201
Installing the Toolset .....	201
Installing the Java SDK.....	201
Installing Eclipse.....	201
Installing the BlackBerry Plug-In .....	202
Installing the BlackBerry Device Simulator .....	202
Installing the BlackBerry Email and MDS Simulator .....	204
“Hello, World” .....	206
Conclusion .....	215
■ <b>Chapter 10: BlackBerry: Building the Logon Form and Consuming REST Services .....</b>	<b>217</b>
How This Project Was Built and Significant Issues with BlackBerry Development .....	217
Creating the Project.....	218
Conventions for Presenting Code.....	219
Calling RESTful Services.....	220
Issuing Web Requests .....	220
Authenticating Our API Account.....	224
Authenticating the User via UsersService .....	235
Creating the Logon Form .....	236
Logging On .....	239
Conclusion .....	242
■ <b>Chapter 11: BlackBerry: An ORM Layer on SQLite .....</b>	<b>243</b>
SQLite on BlackBerry.....	243
Entities.....	243
EntityType.....	244
Creating the Basic Entity Class.....	250
Setting Values in an Entity.....	252
Building Bookmark .....	257

Registering the EntityType.....	259
Displaying Some Fake Bookmarks .....	261
Creating the Form.....	261
Showing a Web Page.....	264
Creating Fake Bookmarks .....	265
Building the Sync Class .....	269
Calling the Server's Bookmarks OData Service.....	269
Managing the Database.....	277
The SqlStatement Class and ISqlStatementSource Interface.....	278
Creating Tables.....	280
Examining the Database with Sqliteman.....	283
Writing Bookmarks to the Database .....	285
Reading Bookmarks and Displaying Them on the Navigator.....	291
Conclusion .....	296
<b>■ Chapter 12: BlackBerry: Pushing Changes to the Server .....</b>	<b>297</b>
Capturing Local Changes.....	297
Constraining SQL Filters .....	297
Excluding Deleted Entities from the View.....	302
Getting a Bookmark by Ordinal.....	303
Building the Configuration Form.....	304
Configuring Singletons .....	313
Pushing Changes to the Server .....	320
My "Facepalm" Moment.....	320
Detecting Local Changes .....	321
Issuing Server Requests to Insert, Update, and Delete .....	325
Update via HTTP MERGE and Insert via HTTP POST .....	326
Marking Fields As Being Available on the Server .....	326
Building XmlSerializer .....	331
Conclusion .....	337
<b>■ Index .....</b>	<b>339</b>

# About the Author

■ **Matthew Baxter-Reynolds** is an independent software development consultant, trainer, and author based in the UK, specializing in mobile technology solutions. He can be contacted via LinkedIn at [www.linkedin.com/in/mbrit](http://www.linkedin.com/in/mbrit).

# About the Technical Reviewer

In 2004, **Matthew Fitchett**, with experience in VB.Net, joined a small e-commerce team to trial C# within a (then) small DVD- and CD-focused e-commerce company.

Play.com went on to become one of Europe's largest e-commerce companies, with Matthew playing a major role as one of a handful of senior software developers. After six and a half enjoyable years, Matthew decided to move on to specialize in mobile technology, which he sees as a significant growth area for software developers and enterprises.

Working alongside Matthew Baxter-Reynolds, Matthew produced prototypes on a variety of technology platforms (Android, iPhone, Windows Phone 7, to name three) for a leading company in the mobile survey software market.

Matthew and his beautiful wife, Sarah, have two young boys, Isaac and Harry, and live in the beautiful town of Bury St. Edmunds. He enjoys films, games, music, and eating good food while drinking good beer, and he regularly practices Muay Thai.

His blog at [www.mattfitchett.com](http://www.mattfitchett.com) covers all of the above, along with more mobile technology discussion.



# Acknowledgments

With much thanks and appreciation to my wife, Andy, for the patience and support she has shown during writing and development of this book, Matt Fitchett for his excellent suggestions and review work, and Jonathan Hassell, Anita Castro, and the others at the Apress team for their sterling work in turning this book into reality.



# Introduction

For me, this book has become all about change. In the time that I have been watching the mobile computing marketplace and developing software solutions for it, there has never been a time when there has been a more rapid series of shifts and changes. A good friend of mine tells me that this is because of market consolidation. As of the time of writing (March 2011), we're looking at the time when the people who will be leaders in this space for the next 20 years jostle for position. There is a ton of money out there being spent, which is fantastic news for the typical reader of this book. Position yourself correctly, and you could earn a seriously good living out of it all.

To illustrate this point about change, I proposed this book to Apress in February 2010, and in the time between then and March 2011, a massive amount of changes have happened.

In a normal year, in a normal market, just a few of these things would be big news.

- Microsoft was still developing and building Windows Mobile 6.5. Windows Phone 7 had not been announced. No one really knows what sort of impact Windows Phone 7 will have.
- The iPad had not been announced, let alone sold the millions and millions of units that it has, and, of course, this has now been followed up with iPad 2. (For me, this is perhaps the biggest change of all—the world will never be the same now that this class of device has been introduced.)
- There was no sign of Android running on tablets. Now we're looking forward to Gingerbread making for a fabulous tablet experience.
- The Pre/webOS was included in the original proposal. HP has now bought Pre, and in the last week or so HP has announced that it intends to include webOS on all of its shipped PCs.
- Android has been growing, and growing, and growing. IDC has this week announced it is the fastest growing OS of all time.
- Canalys has also recently announced that 50 percent of BlackBerry users are looking to defect to iOS or Android.
- The image of Flash hadn't been damaged by Apple's insistence that it had no place on its platform. Although there was a short resurgence, Mozilla has come out saying that it sees little future in it as a platform.
- iPhone 4 had not been announced or released, and "Antennagate" had not happened. Now the rumor mill is talking about iPhone 5.
- You couldn't multitask on an iPhone.
- iOS was still a trademark owned by Cisco.

- Gartner had not come out and likened Symbian to “re-arranging the deck chairs on the Titanic” in the face of the Android threat. Symbian then seemed to spend the next few months dying, not with a bang, but with a whimper. Now, Microsoft and Nokia are moving to become strange bedfellows, effectively moving to Windows Phone 7 as its primary platform.
- No one knew anything about QNX and the BlackBerry PlayBook. It looks like now the PlayBook will even run Android apps.
- Steve Ballmer hadn’t said that Apple had sold more iPads than he would have liked and that “Microsoft-powered tablets are ‘job one’ urgency.” Microsoft still won’t look at rolling out the Windows Phone 7 platform onto tablet devices, insisting that Windows 8 is the platform of choice.
- There was no Amazon Appstore, and no one was doing anything as cool as firing up a Dalvik VM in the cloud to try the app before you buy. (How cool!)
- We didn’t know that Google could remote uninstall applications from any Android phone using a “kill switch.” It has recently used this to kill off a score of applications that were causing problems with users.
- Amazon hadn’t announced its Android app store, although even today the details of it are sketchy.
- The United Arab Emirates had not turned off BlackBerry Enterprise Services within the country.
- Motorola was looking very sick indeed, but it is now looking much healthier thanks to the Droid, Droid X, and Xoom.
- MeeGo had not been announced (and as of the time of writing is not substantial enough to include in this book). My prediction, for what it’s worth, is that this will get traction in spaces like automotive as opposed to slate or phone factors.
- Microsoft announced, launched, and killed a device called “Kin.” To give you some idea of how much money is being thrown around, Microsoft attributes US\$240 million of written-off monies to Kin. That’s not small change.

In fact, this book has been difficult to write because of the velocity of all of this change. I’ll be forever grateful to the team at Apress for managing to corral it into the place where, I hope, it’s helpful and relevant to you, in spite of this almost constant upheaval in the market.

## What’s the Purpose of This Book?

In 2001, I set up a web site called .NET 247 ([www.dotnet247.com/](http://www.dotnet247.com/)) that at the time achieved some success in the community that had sprung up around Microsoft’s new software development toolset. The premise of the site was to help me as a developer migrate my knowledge from pre-.NET technologies (Win32, MFC, classic ASP, etc.) over to .NET. I found it frustrating that spinning up a thread or opening a file would be a few seconds’ work prior to .NET, but in .NET it took hours of research.

With this book, I’ve looked to do a similar thing—answer the common questions and give you a leg up into understanding the platform so that you can get on and do the clever thing that only you’ve thought of. The idea of this book is not to go into masses of detail on every little thing; however, if you



work through all of the different platforms in this book and its companion, you'll know enough to be proficient on any platform that you turn your hand to.

Specifically, what I've tried to concentrate on is the following:

- Getting to a point where you can compile and run an application on the emulator or device
- Showing how to build a user interface—specifically move between forms, handle events, get data on the screen, and capture input
- Showing how to connect to HTTP-based resources so that you can talk to services in the cloud
- Showing how to store and cache data locally for performance and for offline support
- Showing how to build a simple but real application that works end to end

## How Is This Book Structured?

This book is split into three sections. There's an introduction section, which takes you through the background of the two applications that we're going to build. There is then a section on Windows Phone 7 and another section on BlackBerry. There is also a bonus chapter on Windows Mobile.

In addition, this book has a sister book, which is structured similarly and takes you through building the same application that we're going to build in this book. The book's title—*Multimobile Development: Building Applications for Android and iPhone*—should tell you what you need to know.

Each section starts with instructions on how to install the toolset that you are supposed to use with the platform. Some toolsets are very easy to install, while some have gotchas; thus the aim of the toolset installation chapter is mainly to cover the gotchas.

The next three chapters in each section take you through building what's called the “Six Bookmarks” application. This is a very simple application that is designed to show six buttons on the screen, and each button can be configured with a URL that invokes the device's default browser. The purpose of the application is not to be a fantastic piece of UI—it's designed to be a “carrier” to help you understand how to build all of the back-end bits and pieces that you need to make an application functional. Figure 1-1 shows an example.



*Figure 1-1. The Six Bookmarks application running on an iPhone*

Each volume contains two chapters that are *essential* to following the work in the book, and I strongly recommend that you read them first.

To reduce the amount of work required to build the application, Six Bookmarks works on the assumption that there is a cloud-based service that holds the user's bookmarks. In order to use the software on a device, the user needs an account on this service. (This model will seem familiar to all readers of this book, I hope.) Chapter 2 discusses the structure of this service and familiarizes you with the service calls that make the application work.

The second important chapter is Chapter 3, which discusses the functional specification of the Six Bookmarks application and the technical architecture. Again, it's important that you read this in order to understand what it is that you are trying to build.

## Where Can You Get Help and Support?

This book has a companion web site, located at [www.multimobileddevelopment.com/](http://www.multimobileddevelopment.com/), which hosts important resources that will support you in getting the most out of this book. Specifically, you will find the following:

- Downloads of all of the code for all of the platforms
- The Six Bookmarks cloud service implementation that you need to use to make the applications work
- A hosted version of the Six Bookmark HTML application (discussed in detail in Volume 2)
- Support forums (I'll be monitoring and contributing to these, so if you have a question or a problem, this is the best place to try.)

Finally, going back to my earlier point about the amount of flux in the market at the moment, I'll be updating the web site to keep it up-to-date with changes in the toolsets and other movements within the industry.

## Conclusion

Thanks for purchasing this book. Remember that if you do need help or support, then please visit the web site's discussion forums; but if you would like to contact me personally, you can find me at [www.linkedin.com/in/mbrit/](http://www.linkedin.com/in/mbrit/).

Matthew Baxter-Reynolds, April 2011



# The Six Bookmarks Server Service

We're going to talk more about the architecture and specification of the Six Bookmarks application in Chapter 3. In this chapter, we're going to look at the Six Bookmarks service. To support this book, I have set up a server with REST-based (a.k.a. "RESTful") services that allow the application to log on, retrieve bookmarks over the OData protocol, and post updates back, again using the OData protocol. (We'll talk more about OData later on.)

As discussed previously, Six Bookmarks is a commercial product provided in two ways—once as a commercial product and once as an open source product. In this book, we're going to be accessing a service based on the open source version of the code. Both applications communicate with a publically accessible server. The open source server operates a sandbox, and in order to complete the work in this book, you'll need your own account.

---

■ **Note** It's currently very popular to talk about the "cloud" and storing things "in the cloud." The Six Bookmarks server service is one of these "cloud" services—I've provided a server hosted on the public Internet that allows you to store bookmarks "in the cloud" and retrieve bookmarks "from the cloud."

---

We will not be covering how to build this service in the book; however, the source code for the service can be downloaded from the source repository at <http://code.multimobileddevelopment.com/>. This code and all of the other code downloads are distributed under the Mozilla Public License 1.1. More information on this can be found here: [www.mozilla.org/MPL/MPL-1.1-annotated.html](http://www.mozilla.org/MPL/MPL-1.1-annotated.html).

## Creating an API Account

To create an API account, visit the services web site at <http://services.multimobileddevelopment.com/>. You will find a link on that page entitled "Register a new API account". Click this to access a standard registration form, as shown in Figure 2-1.

AMX Mobile Six Bookmarks Service - Test Platform - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://services.amxmobile.com/registr... Google

**AMX Mobile Six Bookmarks Service Test Platform**

Home

## Register

Use this form to register for a new API account.

Username  \*

Password  \*

Confirm password  \*

First name  \*

Last name  \*

Company  \*

Address 1  \*

Address 2

Address 3

City/town  \*

Region/country

Postal code/postcode  \*

Country  \*

Email  \*

☐ Send me occasional news and updates

Done

*Figure 2-1. The service registration form*

---

■ **Note** The site at <http://services.multimobiledvelopment.com> is a live work in progress. Some of the screenshots presented here may differ from the current reality of the site as you see it today. Also, the site you are using is not secured when accessed over HTTPS, as this is a test site not intended for production use. Were you to build a similar thing for production applications, it would be essential that you secure the site using HTTPS.

---

Go ahead and create your account. Please provide a valid e-mail address, as you will need this should you need to reset your password in the future. (You will not get spammed.)

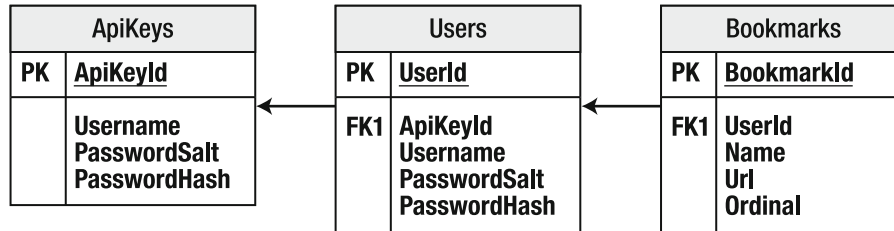
Registering your account will automatically log you on.

## Creating a User

The purpose of registering for an account is to partition off a private section of the database for you to keep your own data in. A single SQL Server database exists on the server, and everyone's users and

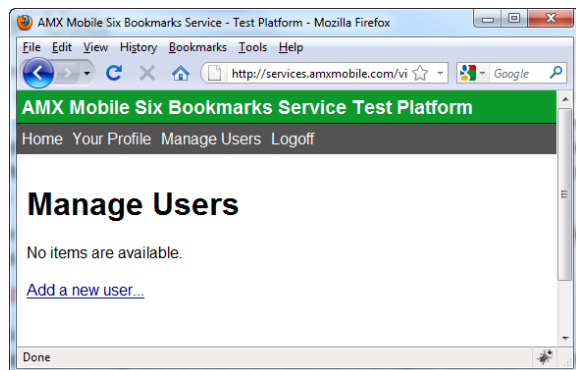
bookmarks are contained within this. This is likely to be slightly different for your own applications. For this book, we need to provide you with a sandbox service that makes it easier for you to work with the chapters on the actual application creation on the devices; however, in production applications, you typically do not need this. I have to hive off individual readers' data into separate "virtual databases" to prevent corruption of data and weird behavior, and with potentially tens of thousands of you out there doing this, it's impractical to create physically separate databases.

Under the covers, you're going to be working with three tables: *ApiKeys*, *Users*, and *Bookmarks*. The entity-relationship diagram (ERD) shown in Figure 2-2 illustrates.



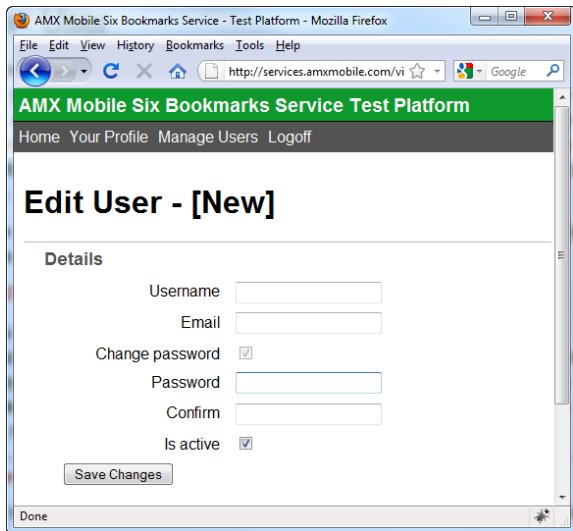
*Figure 2-2. ERD showing relationship between the *ApiKeys*, *Users*, and *Bookmarks* tables*

When you register for an API account, you do not get any users created for you. A user in this context relates to someone who would use an instance of the various mobile Six Bookmarks applications targeted for separate device platforms. To create a user, click the *Manage Users* link. You will be presented with a message that indicates no users are available, as per Figure 2-3.



*Figure 2-3. The *Manage Users* page showing no available users*

Click the "Add a new user" link to enter a new user. Figure 2-4 illustrates.



*Figure 2-4. The Edit User page*

You'll need to create at least one user in order to proceed to the next section.

## The Users Service

The “Users” service is a RESTful web service that provides a capability to log on a user. (This book deals only with logging users on; however, the service is capable of other functions, including registering users.) It's important to familiarize yourself with how the service works, as it will aid in understanding the flow of the applications that we will build in later sections.

## RESTful Web Services

A “RESTful” web service is a service that is based on the principle of REST, which stands for “Representational State Transfer.” It is not a formal, standardized protocol, but rather a set of principles or constraints that describes the shape and operational usage of a service that you can get data from or provide data to. It is a very natural way of working with remote services, which is why they are so popular and prevalent. That naturalness translates into being very easy to build, and equally very easy to consume.

One common and straightforward way of structuring a RESTful web service is to request data using an HTTP GET request and retrieving results back as XML. The HTTP request can be a GET request, including parameters specified in the query string. Alternatively, the parameters can be made via a POST request that works by passing up XML.

Let's continue this by looking in more detail at the logon operation on the Users service.

## Testing the Calls

The API relies on sending up custom HTTP headers, and as such we can't test it using a regular web browser. Rather than asking you to build some custom code to call the service, you can download a test harness for trying the service. You can download this from the source repository at

<http://code.multimobiledvelopment.com/>. Look for a file in the Downloads section of the form `Amx.Services-<Version>-TestClient.zip`. This is a .NET application, and hence you'll need the .NET runtime installed on the machine you're looking to use.

If you download the utility and run it, you'll see you have an area to enter a URL and an area to enter two values: the API username header and the Token header. We'll talk about these values later, but essentially they provide additional information to the service to help guide the response.

## Examining Logon Operations

The first thing we can try to do with our Users service is log on a user. Ultimately, a successful logon will return a token that we can use in subsequent requests.

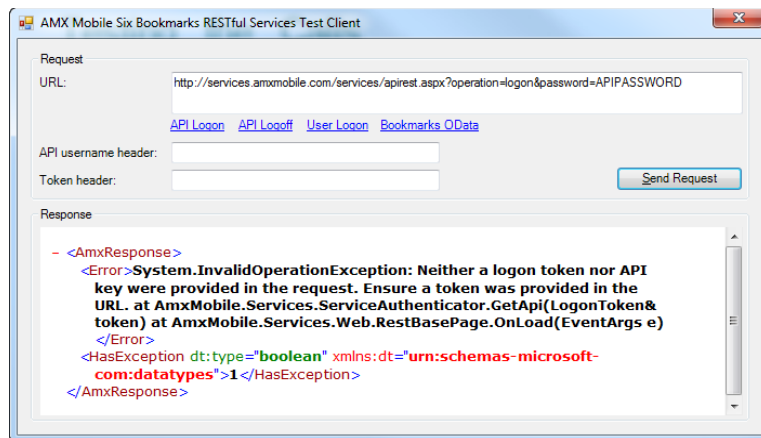
If you open the test harness, the URL will be given as follows:

---

<http://services.multimobiledvelopment.com/services/apirest.aspx?operation=logon&password=APIPASSWORD>

---

Click the Send Request button, and you'll see a result like Figure 2-5.



**Figure 2-5.** An example of a failed request to the API service

You can see in the response that an error has been returned.

The protocol for the REST services exposed by the service is that exceptions are returned back in the Error element, and the HasException element is set to true if an error has been returned. (The value shown in the XML is 1, but the datatypes schema is used to indicate that this is a Boolean value.)

---

■ **Note** This error notification and transmission are just how I have designed the service—it doesn't follow that all RESTful web services will use this approach. It's down to the owner of the service to design a protocol that is sensible and logical to use within the loose construct of what a RESTful service typically looks like.

---



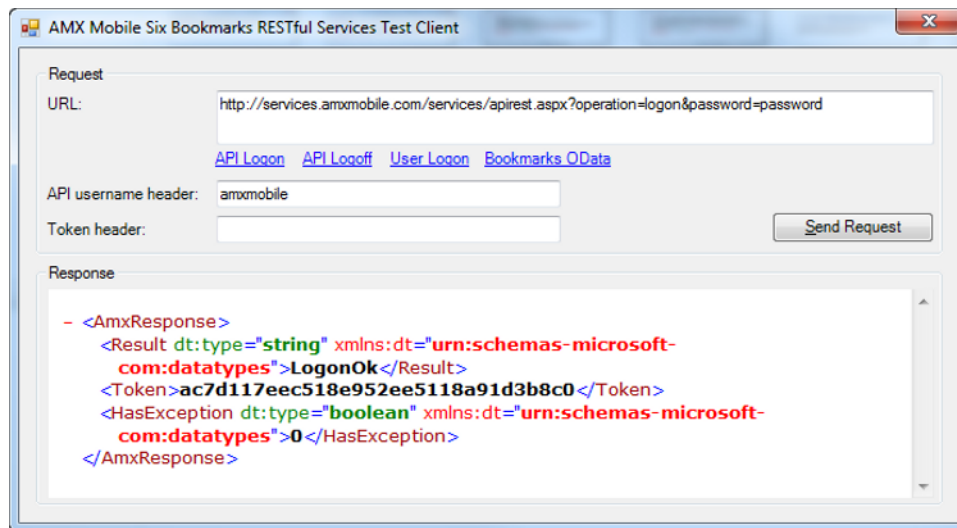
Referring back to Figure 2-5, we see the error indicates that “Neither a logon token nor API key were provided in this request.” What this is telling us is that the headers have not been provided to the server.

To call the operations on the server, we need a token. In order to get a token, we need to call the server, so we have a chicken and egg situation! However, one operation on the server does not need a token—this is the Logon operation on the API service, which is used solely to obtain a token for use with the other methods.

## Obtaining a Token

By default, when you start the harness, it will be set to connect to the API service and to call the Logon method. Firstly, into the API username header text box, enter the username of the account you created in the first part of the chapter. Secondly, modify the password value in the URL to be the username on your account.

If you click Send Request now and the details are correct, you’ll see something similar to the image shown in Figure 2-6.



*Figure 2-6. An example of the result of a successful request to the API service*

You’ll see in this case an error has not been returned. The Result element will be set to LogonOk or InvalidPassword. (Any other errors will result in an exception being returned.)

The most important element here is the Token value. This is the token that we’ll use in all other requests. Copy the token into the clipboard, and then paste it into the Token header field. We’ll use this later.

## Logging On the User

Now that we have obtained a token to use and authenticated the API, we can actually log on the user. We’ve used the API service so far—we’re now going to use the Users service.

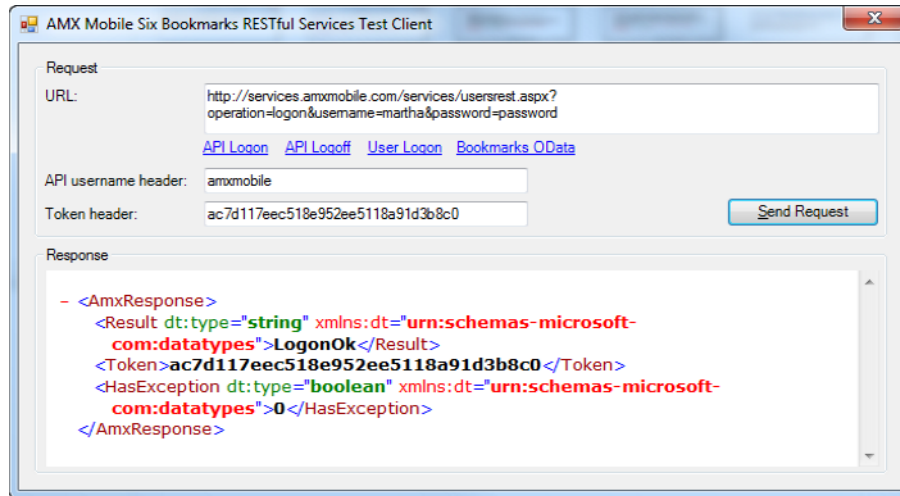
If you click the User Logon link on the test harness, the URL will be rewritten to the following:

---

```
http://services.multimobiledvelopment.com/services/usersrest.aspx?operation=logon&username=
USERNAME&password=PASSWORD
```

---

This URL is configured to call the Users REST service. If you replace the USERNAME and PASSWORD placeholders in that string, and assuming you have copied the token into the Token header field, and click Send Request, you'll get a response like Figure 2-7, which, apart from the URL, looks identical to Figure 2-6.



*Figure 2-7. An example of a response from a successful request to the Users service*

Assuming this works, you'll see another LogonOk response. What a LogonOk tells you here is that the token is now bound to the user you authenticated. (This is important—this means that you cannot use the same token with different users. This will never be a problem on a mobile device as these are typically solo-user devices and one's global state only ever refers to oneself, but in a web application, it is worth considering.) Other results you can get back from the service are InvalidUsername, InvalidPassword, or AccountInactive.

## Cleaning Up

To clean up the service, we have to log off of the API. This is done via the Logoff operation. Click the "API logoff" link on the harness, and the URL will be rewritten once again. Click the Send Request button, and you'll see a response much like in Figure 2-8.

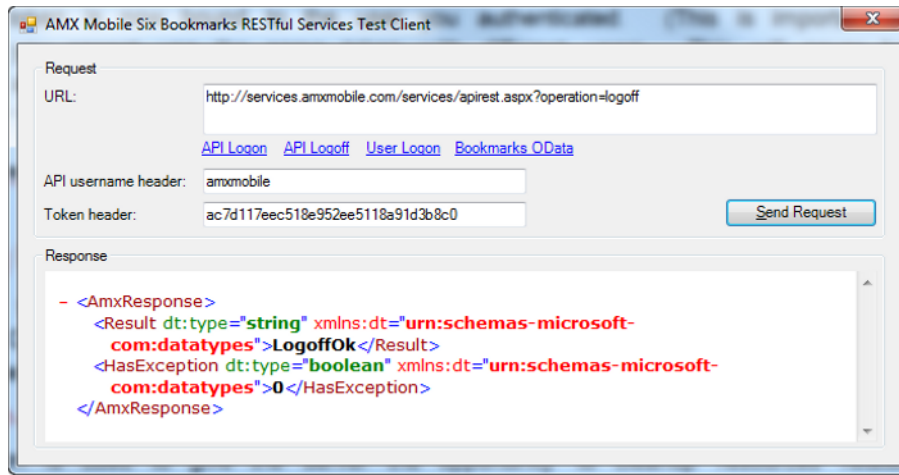


Figure 2-8. An example of a successful Logoff call to the API service

This operation is used to give the server the opportunity to clean up resources related to the token. (Specifically, it deletes a row in the database.) We'll look at token cleanup in more detail when we build the native device applications.

## The Bookmarks Service

The final service exposed from the server is the Bookmarks OData service. OData is an up-and-coming data format that is currently being pitched as the proposed *de facto* standard for data interchange in a Web 2.0 world. My opinion is that it is a decent standard with a good, practical working method, and hence I've chosen to use it in this book to bridge the gap between relational data stored in the cloud and data stored on the device.

---

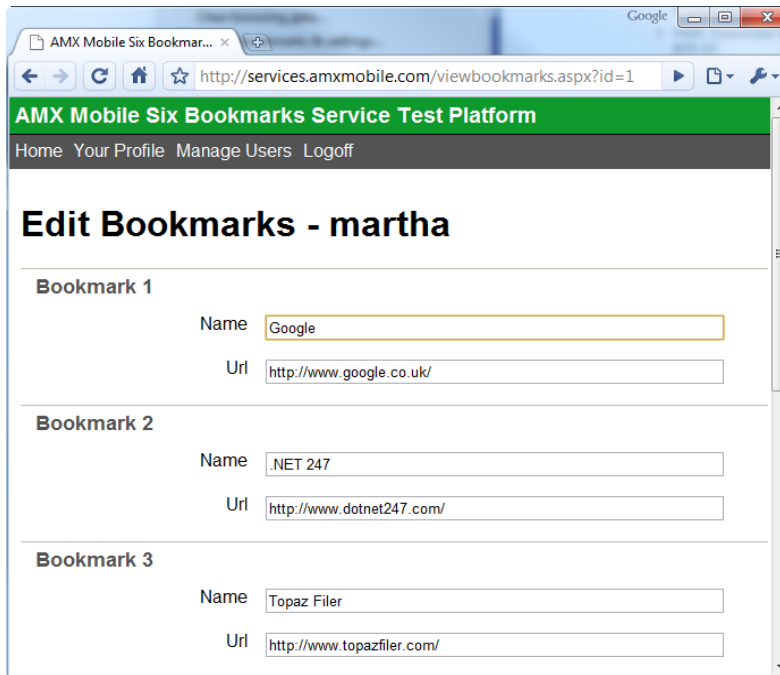
■ **Tip** You can find out more about OData at the official site: [www.odata.org/](http://www.odata.org/).

---

## Adding Some Test Data

In order to see how the OData service works, you're going to need some test data. There's an interface on the service that lets you maintain the bookmarks against a user.

Log on to [services.multimobileddevelopment.com](http://services.multimobileddevelopment.com), and find a user that you want to work with. Click the "Manage this user's bookmarks" link at the bottom of the page. You will see an interface that allows you to define bookmarks. Figure 2-9 illustrates.



*Figure 2-9. The Edit Bookmarks screen showing three bookmarks*

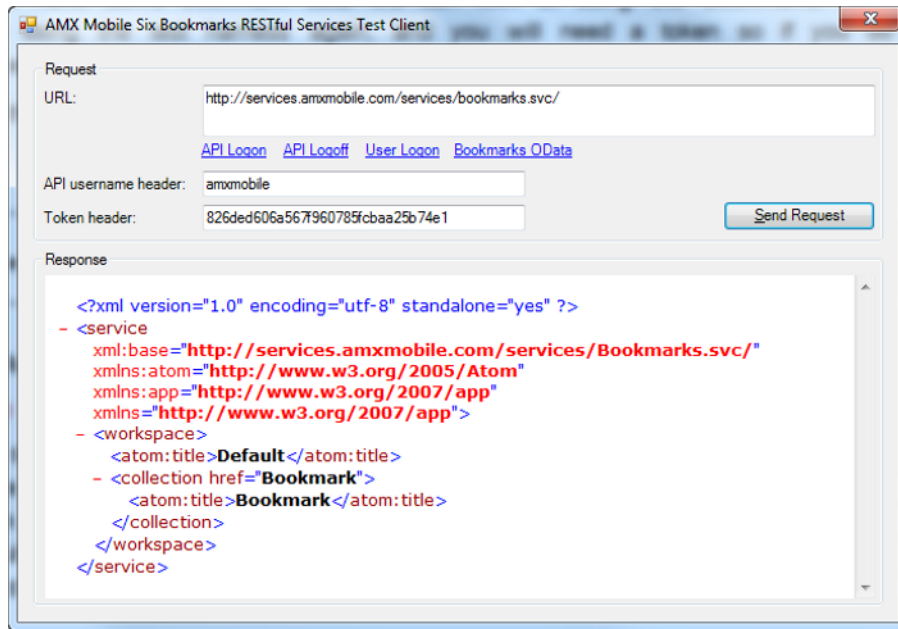
Add a number of bookmarks, and click Save Changes.

## Working with OData

Now that we have some bookmark data, we can look at using the Bookmarks service. We're going to be using the test harness again, and you will need a token—so if you do not currently have a token, go through the steps described previously to obtain one.

On the harness, if you click the Bookmarks OData link, you'll get a rewritten URL, like this one: `http://services.multimobiledvelopment.com/services/bookmarks.svc/`.

Click Send Request, and you'll get a response like Figure 2-10. You should note that the test harness continues to send up the special headers. The service call would be rejected should these headers be missing or incorrect.



*Figure 2-10. An example of a successful call to the Bookmarks OData service*

---

■ **Note** The OData standard allows for data to be returned either in Atom or JSON format. Atom format is the most relevant here—JSON is typically used when working with Ajax calls from a web page. The actual format of the data is not important—what is important is that OData is built on open standards. (Notably, Microsoft sees OData as a core data protocol going forward, starting with a full implementation in .NET 3.5 SP1 and support on the Azure platform.)

---

The preceding output is telling us that the Bookmarks service is about to return data of type Bookmark (look for the `//collection/atom:title` element in the XML). Thus, if we issue this URL, again using the test harness, we'll get back some bookmarks. Here's the URL:

---

```
http://services.multimobileddevelopment.com/services/bookmarks.svc/Bookmark
```

---

From this point, I'm going to show you the XML output as a listing, rather than screenshots. This will make it easier to follow the discussion.

In the following example, three bookmarks are returned from this call, and these are shown in the following listing. (Your output will vary depending on the bookmarks you've set up against the user that you've logged in as, obviously.) Here's the listing:

```

<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<feed xml:base="http://services.m multimobileddevelopment.com/services/Bookmarks.svc/" ↵
  xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" ↵
  xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata" ↵
  xmlns="http://www.w3.org/2005/Atom">
  <title type="text">Bookmark</title>
  <id>http://services.m multimobileddevelopment.com/services/bookmarks.svc/Bookmark</id>
  <updated>2010-04-18T10:54:32Z</updated>
  <link rel="self" title="Bookmark" href="Bookmark" />
  <entry>
    <id>http://services.m multimobileddevelopment.com/services/Bookmarks.svc/Bookmark(1002)</id>
    <title type="text"></title>
    <updated>2010-04-18T10:54:32Z</updated>
    <author>
      <name />
    </author>
    <link rel="edit" title="Bookmark" href="Bookmark(1002)" />
    <category term="AmxMobile.Services.Bookmark" ↵
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
    <content type="application/xml">
      <m:properties>
        <d:BookmarkId m:type="Edm.Int32">1002</d:BookmarkId>
        <d:UserId m:type="Edm.Int32">1001</d:UserId>
        <d:Name>.NET 247</d:Name>
        <d:Url>http://www.dotnet247.com/</d:Url>
        <d:Ordinal m:type="Edm.Int32">1</d:Ordinal>
      </m:properties>
    </content>
  </entry>
  <entry>
    <id>http://services.m multimobileddevelopment.com/services/Bookmarks.svc/Bookmark(1001)</id>
    <title type="text"></title>
    <updated>2010-04-18T10:54:32Z</updated>
    <author>
      <name />
    </author>
    <link rel="edit" title="Bookmark" href="Bookmark(1001)" />
    <category term="AmxMobile.Services.Bookmark" ↵
scheme="http://schemas.microsoft.com/ado/2007/08/dataservices/scheme" />
    <content type="application/xml">
      <m:properties>
        <d:BookmarkId m:type="Edm.Int32">1001</d:BookmarkId>
        <d:UserId m:type="Edm.Int32">1001</d:UserId>
        <d:Name>Google</d:Name>
        <d:Url>http://www.google.co.uk/</d:Url>
        <d:Ordinal m:type="Edm.Int32">0</d:Ordinal>
      </m:properties>
    </content>
  </entry>
  <entry>
    <id>http://services.m multimobileddevelopment.com/services/Bookmarks.svc/Bookmark(1003)</id>

```