



F r o m T e c h n o l o g i e s t o S o l u t i o n s

RESTful PHP

Web Services

Learn the basic architectural concepts and steps through examples of consuming and creating RESTful web services in PHP

Samisa Abeysinghe

[PACKT]
PUBLISHING

RESTful PHP Web Services

Learn the basic architectural concepts and steps through examples of consuming and creating RESTful web services in PHP

Samisa Abeysinghe



BIRMINGHAM - MUMBAI

RESTful PHP Web Services

Copyright © 2008 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, Packt Publishing, nor its dealers or distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: October 2008

Production Reference: 1201008

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 978-1-847195-52-4

www.packtpub.com

Cover Image by Nilesh Mohite (nilpreet2000@yahoo.co.in)

Credits

Author

Samisa Abeysinghe

Project Manager

Abhijeet Deobhakta

Reviewers

Md Emran Hasan

Suhreed Sarkar

Deepak Vohra

Project Coordinator

Rajashree Hamine

Indexer

Rekha Nair

Senior Acquisition Editor

Douglas Paterson

Proofreader

Laura Booth

Development Editor

Swapna V. Verlekar

Production Coordinator

Aparna Bhagat

Technical Editor

Siddharth .D. Mangarole

Cover Work

Aparna Bhagat

Editorial Team Leader

Mithil Kulkarni

About the Author

Samisa Abeysinghe is Director, Engineering at WSO2. Samisa pioneered the Apache Axis2/C effort and architected the core of the Apache Axis2/C Web services engine. He continues to be an active contributor in the Apache Axis2/C project. His involvement in open-source projects began in 2004 when he started working with the Apache Axis C/C++ project.

Prior to his current role, Samisa played the project lead role for the WSO2 Web Services Framework for PHP project, which provides comprehensive support for building both SOAP and REST services and clients.

"No man is an island" – John Donne

As human beings, we do not thrive when isolated from others. This book was no exception. Many people contributed to the successful completion of this book, and I would like to acknowledge all those who contributed.

First, I must thank Douglas Paterson. Douglas, Senior Acquisition Editor of Packt Publishing Ltd., is the one who initially proposed to me that I write this book. And thanks to him, this book was born.

Next, my gratitude goes to Sanjiva Weerawarana, Founder, Chairman, and CEO of WSO2, Inc. When I first consulted Sanjiva on his thoughts on whether I should be writing this book, he encouraged me and even offered to help.

Speaking about encouragement, I must thank my mother, who checked, on a weekly basis, if I was continuing with my work on the book and the progress that I was making on that front.

The staff at Packt Publishing Ltd. helped a great deal to make this book a reality. I would like to thank Rajashree Hamine the project coordinator, Swapna Verlekar the development editor, and Siddharth Mangarole the technical editor. I would also like to thank all others from Packt Publishing Ltd. who contributed to this book in many ways.

I would also like to thank some of my WSO2 colleagues, who worked with me closely on the scripting projects, specially WSO2 WSF/PHP. I would like to mention Nandika, Dimuthu, Chinthana, and Buddhika. Though they did not work on this book directly, they helped me a lot to understand PHP while working on WSF/PHP.

About the Reviewers

Md. Emran Hasan is a web application developer, usability consultant, and a successful entrepreneur from Bangladesh. He has a Bachelor in Business Administration with MIS major and is currently pursuing his MBA. In his early days with programming, he developed a number of desktop-based business applications for clients all over the globe including US, UK, Canada, Australia, Malaysia, and Spain. Later he switched to the Web and started programming in PHP.

He developed the largest social community blogging platform in Bangladesh called "Badh Bhangar Awaj" (<http://www.somewhereinblog.net>), while working at Somewhere In Net Ltd. He then worked in Pageflakes (<http://www.pageflakes.com>) – two times "Web 2.0 Award" Winner for Start page and Trippert Labs, Inc. (<http://www.trippertlabs.com>) – which develops social software for large companies such as Electronic Arts.

Emran's work toolbox includes CodeIgniter and Zend Framework for rapid application development in PHP, MySQL, and SQLite for efficient data storage, jQuery & Dojo for feature rich UI, W3C valid XHTML and CSS for standard-compliant site layout, and strong Usability and Accessibility sense for pleasant user experience.

Currently Emran is leading his own web development company, Right Brain Solution Ltd. (<http://www.rightbrainsolution.com>), as the Chief Technical Officer. He is the technical lead and helps the company deliver industry-standard web solutions. When he is not working in his job or browsing around or replying to threads in the phpXperts group (<http://tech.groups.yahoo.com/group/phpexperts>), he listens to music, reads book, and writes in his technical blog at <http://www.phpfour.com>.

Suhreed Sarkar is an IT consultant, trainer, and technical writer. He studied Marine engineering, served on board a ship for two years, and then started his journey in to the IT world with MCSE in Windows NT 4.0 track. Later he studied business administration and earned his MBA from the University of Dhaka. He has a bunch of BrainBench certifications on various topics including PHP4, Project Management, RDBMS Concepts, E-commerce, Web Server Administration, Internet Security, Training Development, Training Delivery and Evaluation, and Technical Writing.

As a trainer, he taught courses on system administration, web development, e-commerce, and MIS. He has consulted for several national and international organizations including the United Nations, and helped clients building and adopting their enterprise portals, large scale databases, and management information systems. He is a renowned technical author in Bengali—having a dozen of books published on subjects covering web development, LAMP, networking, and system administration. He authored *Zen Cart: E-commerce Application Development*, published by Packt Publishing.

While not busy with hacking some apps, blogging on his blog (www.suhreedsarkar.com), reading the philosophy of Bertrand Russel or the management thought of Peter F. Drucker, he likes to spend some special moments with his family. Suhreed lives in Dhaka, Bangladesh and can be reached at suhreedsarkar@gmail.com.

I would like to thank the team at Packt who provided excellent support to work on this book, especially Swapna Verleker and Rajashree Hamine. I am also grateful to my family and friends for allowing me to work on this.

Deepak Vohra is a consultant and a principal member of the NuBean.com software company. Deepak is a Sun Certified Java Programmer and Web Component Developer, and has worked in the fields of XML and Java programming and J2EE for over five years. Deepak is the co-author of the Apress book *Pro XML Development with Java Technology* and was the technical reviewer for the O'Reilly book *WebLogic: The Definitive Guide*. Deepak was also the technical reviewer for the Course Technology PTR book *Ruby Programming for the Absolute Beginner*, and the technical editor for the Manning Publications book *Prototype and Scriptaculous in Action*. Deepak is also the author of the Packt Publishing book *JDBC 4.0 and Oracle JDeveloper for J2EE Development*.

Table of Contents

Preface	1
Chapter 1: Introduction to REST	7
Programmable Web	8
HTTP and Web Services	11
What is REST?	13
HTTP Methods	14
The Need for RESTful Web Services	16
REST Tools and Frameworks in PHP	17
XML Parsers	17
Tools for Accessing Services	18
Providing Services	18
PHP REST Frameworks	19
Tonic	19
Konstrukt	19
Zend Framework	20
WSO2 WSF/PHP	20
Madeam	20
dbscript	20
What Framework to Use	20
Summary	21
Chapter 2: REST with PHP—A First Look	23
HTTP with PHP	23
CURL	24
HTTP GET	26
HTTP POST	28
HTTP PUT	30
HTTP DELETE	31

Building the Request with XML Tools	32
SimpleXML	33
DOM	34
Processing the Response	35
SimpleXML	35
DOM	36
Consuming Flickr	37
Photo Search	39
Photo Search with Information	43
Summary	49
Chapter 3: REST in the Real World	51
Types of Services Available	51
Consuming Real-World Services	52
Cresting our Utility Code—RESTUtil.php	54
Consuming an RSS Feed—BBC News Feed	54
BBC News Feed with Yahoo News Search	58
Yahoo Maps and Local Search	61
Earthquakes and Yahoo Maps	70
Mashups	74
Summary	74
Chapter 4: Resource-Oriented Services	77
Designing Services	77
Simplified Library System	78
Resource Design	79
PUT vs POST	80
URI Design	81
URI and HTTP Verb Mapping	82
System Implementation	83
Library Database	83
Web Page from Data	85
Retrieve Operation	87
Create Operation	92
Handling Multiple Path Parameters	95
Summary	102
Chapter 5: Resource-Oriented Clients	103
Designing Clients	103
Resource Design	104
System Implementation	105
Retrieving Resource Information	105
Creating Resources	108

Deleting Resources	110
Putting it All Together	111
Implementing a Form-based Application	112
Summary	124
Chapter 6: Resource-Oriented Clients and Services with Zend Framework	125
Installing Zend Framework	125
Services with Zend_Rest_Server	126
Clients with Zend_Rest_Client	127
Library System with Zend REST classes	129
Library Service	129
Controllers for Book and Member Resources	131
Models for Book and Member Resources	131
Application Configuration and Initialization	132
Book Controller	134
Member Controller	138
Library Clients	141
List Books with GET	141
Add a Book with POST	143
List Members with GET	146
Add a Member with POST	148
Complete Client Application Controller	149
Summary	151
Chapter 7: Debugging REST Web Services	153
Message Tracing	153
Errors in Building XML	164
Errors in Parsing XML	166
Best Practices	167
Summary	169
Appendix A: WSO2 Web Services Framework for PHP	171
Installing WSF/PHP	172
Implementing Services	172
Implementing Clients	176
SOAP Service and Client	178
Summary	183
Appendix B: RESTClient class	185
get Method	186
post Method	187
put Method	188
delete Method	189

Table of Contents

Complete RESTClient Class	190
get Example	195
post Example	196
Yahoo Search Client Example	197
Summary	197
Index	199

Preface

This book discusses the use of PHP to implement web applications based on REST architectural principles. Web services are a popular breed of web application technologies in today's programmable Web, and REST is the most popular style used in there. This book uses real-world examples as well as step-by-step guidelines to explain how to design REST-style services and clients from the ground up and how to use PHP programming constructs and frameworks to implement those services and clients.

What This Book Covers

Chapter 1 introduces the concepts related to the programmable Web, shows how HTTP and web services are related to each other, introduces the principles behind REST, explains how HTTP verbs are used in REST applications, explains the need for RESTful web services while building PHP web applications, and introduces some frameworks and tools that can be used to work with REST in PHP.

Chapter 2 takes a first look at REST with PHP. While providing and consuming REST-style web services, the primary pre-requisites are an HTTP server or an HTTP client library and an XML parser library. In this chapter, we will see how to use the PHP CURL API to consume web services using various HTTP verbs such as HTTP GET, POST, PUT, and DELETE. The DOM API and SimpleXML API for building XML object structures and parsing XML streams are also discussed. We will discuss in detail how to build XML request payloads and also how to parse XML response payloads. The final section of this chapter demonstrates how to use the HTTP client features and XML parser features to invoke the Flickr REST API.

Chapter 3 looks into some real-world applications and discusses how to combine multiple service interfaces to build value-added custom applications. In this chapter, we will see how to use RSS or ATOM feeds, Yahoo search API, and Yahoo maps API. With the know-how you gain in this chapter and the previous chapters, you could build very powerful value-added applications like mashups using publicly available REST-style services.

Chapter 4 covers the steps that you would have to follow in designing and implementing a resource-oriented service in detail. Identifying resources and business operations for a given problem statement, designing the URI patterns, selecting the correct HTTP verbs, mapping URI and HTTP verbs to business operations are covered using the library example. Implementing the services and business operations using PHP is explained in detail, step by step.

Chapter 5 covers the steps that you would have to follow in designing and implementing resource-oriented clients in detail. The design of the clients is governed by the design of the service. And the client programmer needs to understand the semantics of the service, which is usually communicated through service API documentation. In the examples of this chapter, we will use the library service API designed in Chapter 4 to explain how we could use an existing API while designing PHP applications.

Chapter 6 uses the REST classes provided with the Zend Framework to implement the sample library system. The design of the service and client are covered, along with the MVC concepts supported by the Zend Framework. We will discuss how resources map to the model in MVC, and how HTTP verbs when combined with resource URIs map to the controller in MVC. We will explore how to combine `Zend_Rest_Server` with `Zend_Controller` to implement the business operations of the service and how to use `Zend_Rest_Client` class to consume the services.

Chapter 7 looks into the use of tools to trace and look into the messages to figure out possible problems with request and response pairs passed between clients and services. That helps with debugging and troubleshooting of services and clients. We will also look into how we could look at the XML documents to figure out possible problems in building XML in this chapter, and discuss how we can locate problems in parsing incoming XML messages.

Appendix A introduces the WSO2 Web Services Framework for PHP (WSO2 WSF/PHP) and discusses how to use the WSF/PHP service API to implement the sample Library system as a REST service and implement a REST client to consume it. We will also look into using the SOAP features provided in the frameworks to implement a SOAP client to consume the same service using SOAP-style messages. This chapter also discusses the differences between REST and SOAP message styles, in brief.

Appendix B introduces a PHP class named `RESTClient` that can be used to consume REST-style services. This class supports all key HTTP verbs, `GET`, `POST`, `PUT`, and `DELETE`. The advantage of using such a class is that it minimizes the complexity of your client code. At the same time, you can re-use this class for all your REST-style client implementations. This PHP class is sufficient for most simple REST-style client programs, and requires no third-party libraries. However, if you want to implement services and also want advanced clients, it is advised to use a more established framework such as Zend Framework or WSO2 WSF/PHP introduced in Chapter 6 and Appendix A of this book.

What You Need for This Book

You need PHP5 installed with Apache httpd server to try out the samples of this book. You would require a MySQL installation to try out the library sample discussed in the book.

You also need to install Zend Framework and WSO2 WSF/PHP to try out the samples based on those frameworks.

Who This Book is For

This book is for PHP programmers who are interested in using Web Services in their applications. Sometimes, you would be interested in using the publicly available REST-style services in your own applications, in which case, the REST client concepts discussed in this book would be very useful. You might also be involved with the implementation of PHP applications where you want to expose some aspects of the application as services to the outside world, in which case, you can benefit from the REST service concepts covered in this book. In addition, if you are a software developer looking for a hands-on text that will help you understand REST principles, from the ground up, this book would be a very good guide for you.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: ISBN0001

A block of code will be set as follows:

```
<books>
  <book>
    <id>1</id>
    <name>Book1</name>
    <author>Auth1</author>
    <isbn>ISBN0001</isbn>
  </book>
</books>
```

New terms and **important words** are introduced in a bold-type font. Words that you see on the screen, in menus or dialog boxes for example, appear in our text like this: "clicking the **Next** button moves you to the next screen".



Important notes appear in a box like this.



Tips and tricks appear like this.

Reader Feedback

Feedback from our readers is always welcome. Let us know what you think about this book, what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply drop an email to feedback@packtpub.com, making sure to mention the book title in the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or email suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer Support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the Example Code for the Book

Visit http://www.packtpub.com/files/code/5524_Code.zip to directly download the example code.

The downloadable files contain instructions on how to use them.

Errata

Although we have taken every care to ensure the accuracy of our contents, mistakes do happen. If you find a mistake in one of our books — maybe a mistake in text or code — we would be grateful if you would report this to us. By doing this you can save other readers from frustration, and help to improve subsequent versions of this book. If you find any errata, report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **let us know** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata added to the list of existing errata. The existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide the location address or website name immediately so we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with some aspect of the book, and we will do our best to address it.

1

Introduction to REST

When we look around the Web today, we can see a whole new breed of web applications compared to those available a few years back. It is a whole new Web, and some even call it as Web 2.0. What makes Web 2.0 possible? Web services are one of the key technologies that make the Web as powerful as we can see it is today.

Web services allow heterogeneous systems to communicate with each other using messages. Because the systems could be heterogeneous, the need for interoperability arises. Hence XML is often used to format the messages. Because XML is in text format, almost all systems can understand the messages and work with each other. Messages are used when it comes to communicating between applications that run on different machines. As an example, in a chat application, the text typed in by the users are wrapped in messages, along with the data that would explain where the message should go and how that should be interpreted and passed between the server applications.

There are various technologies that could be used to implement web services. **Representational State Transfer** or **REST** has over time become the preferred technology for web services used in web applications. SOAP web services, also known as **WS-* Stack**, is also a popular alternative. However, there are criticisms against SOAP style services, especially related to the complexity and bulkiness of messages, when it comes to using the services for web applications. Due to the simplicity, ease of use, and the extensive use of web-based technologies such as HTTP that the Web developers are already familiar with, REST has become more popular among web application developers.

This chapter will introduce REST and the concepts related to REST. As a preview, here are the key REST principles to be discussed in this chapter:

- The concept of resource (for example, a document is a resource)
- Every resource given a unique ID (for example, document URL)
- Resources can be related (for example, one document linking to another)
- Use of a standard (HTTP, HTML, XML)

- Resources can have multiple forms (for example, status of a document, updated, validated, deleted)
- Communication in a stateless fashion using HTTP (for example, subsequent requests not related to each other)

Programmable Web

The initial intended use of the Web was to share information among the members of academic research teams. The academicians wanted an easy way to set up and maintain infrastructure to share their findings. They often wanted to link their documents to that of others and previous related work, so they used hyperlinks to site relevant documents.

A useful abstraction of this principle is a document-based hypermedia model that provides content to the users. In the 1990s, Web was used as a platform for distributing information, and it experienced an explosion of users due to the visual appeal of the hypermedia model.

In sync with the ever-increasing number of users, the number of web-based applications too kept up with the pace. With the large number of applications, the volume of data available on the Web has grown tremendously.

With the data available, apart from web applications that could be accessed by users with a web browser, developers built services that could be used by other applications. The programmable Web is the set of enabling technologies that helps developers build services for the Web.

As an example, think of a weather service. More often than not, people who are travelling are interested in weather. So, a travel-related web application could benefit by presenting the users with weather data on the travel website itself. A developer implementing the travel application could consume a weather service to access the weather information and integrate it with the travel application.

RSS (Really Simple Syndication) is a family of Web feed formats used to publish frequently updated contents such as weather information. An RSS document is essentially an XML document. The Yahoo weather service, located at <http://developer.yahoo.com/weather/> provides you with an RSS feed of weather. The following code shows how you can access this code using a few lines of PHP:

```
<?php
    $url = 'http://weather.yahooapis.com/forecastrss?p=USNY0996';
    $xml = file_get_contents($url);
    echo $xml;
?>
```

This piece of code would fetch the RSS feed that contains the New York weather information. In the above example, the returned XML is just echoed. With a bit of XML processing, you can extract the weather information from the RSS feed returned.

Following is a sample response returned from the service.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<rss version="2.0" xmlns:yweather="http://xml.weather.yahoo.com/ns/
rss/1.0" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
<channel>

<title>Yahoo! Weather - New York, NY</title>
<link>http://us.rd.yahoo.com/dailynews/rss/weather/New_York__NY/
*http://weather.yahoo.com/forecast/USNY0996_f.html</link>
<description>Yahoo! Weather for New York, NY</description>
<language>en-us</language>
<lastBuildDate>Sat, 16 Aug 2008 8:51 am EDT</lastBuildDate>
<ttl>60</ttl>
<yweather:location city="New York" region="NY" country="US"/>
<yweather:units temperature="F" distance="mi" pressure="in"
speed="mph"/>
<yweather:wind chill="66" direction="0" speed="3" />
<yweather:atmosphere humidity="87" visibility="7" pressure="30.02"
rising="1" />
<yweather:astronomy sunrise="6:08 am" sunset="7:52 pm"/>
<image>
<title>Yahoo! Weather</title>
<width>142</width>
<height>18</height>
<link>http://weather.yahoo.com</link>
<url>http://l.yimg.com/us.yimg.com/i/us/nws/th/main_142b.gif</url>
</image>
<item>
<title>Conditions for New York, NY at 8:51 am EDT</title>
<geo:lat>40.67</geo:lat>
<geo:long>-73.94</geo:long>
<link>http://us.rd.yahoo.com/dailynews/rss/weather/New_York__NY/
*http://weather.yahoo.com/forecast/USNY0996_f.html</link>
<pubDate>Sat, 16 Aug 2008 8:51 am EDT</pubDate>
<yweather:condition text="Fair" code="34" temp="66" date="Sat, 16
Aug 2008 8:51 am EDT" />
<description><![CDATA[
<br />
<b>Current Conditions:</b><br />
Fair, 66 F<br />
```