



From Technologies to Solutions

jQuery 1.3 with PHP

Enhance your PHP applications by increasing their responsiveness through jQuery and its plugins.

Kae Verens

PACKT
PUBLISHING

jQuery 1.3 with PHP

Enhance your PHP applications by increasing their responsiveness through jQuery and its plugins

Kae Verens



BIRMINGHAM - MUMBAI

jQuery 1.3 with PHP

Copyright © 2009 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 2009

Production Reference: 1211009

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

ISBN 978-1-847196-98-9

www.packtpub.com

Cover Image by Louise Barr (lou@frogboxdesign.co.uk)

Credits

Author

Kae Verens

Reviewers

Akash Mehta

Ashok Modi

John K. Murphy

Acquisition Editor

Douglas Paterson

Development Editor

Darshana D. Shinde

Technical Editor

Ishita Dhabalia

Copy Editor

Sanchari Mukherjee

Editorial Team Leader

Gagandeep Singh

Project Team Leader

Lata Basantani

Project Coordinator

Srimoyee Ghoshal

Proofreader

Chris Smith

Indexer

Rekha Nair

Production Coordinator

Dolly Dasilva

Cover Work

Dolly Dasilva

About the Author

Kae Verens lives in Monaghan, Ireland with his wife Bronwyn and their two kids Jareth and Boann. He has been programming for twenty years, fifteen of which were as a professional.

Kae started writing in JavaScript in the mid nineties, and started working on the server-side languages a few years later. After writing CGI in C and Perl, Kae switched to PHP in 2000, and has worked with it since then.

Kae is currently the secretary of the Irish PHP Users' Group, <http://php.ie/>, is part-owner of the Irish web-development company Webworks.ie, <http://webworks.ie/>, and is the author of popular web-based file-manager KFM, <http://kfm.verens.com/>.

In his spare time, Kae plays the guitar, juggles, is learning to play the piano, and likes to occasionally dust the skateboard off and mess around on it.

I would like to thank Packt Publishing for considering me for this project; it's been a journey, and I've learned quite a lot from it. I would also like to thank Webworks.ie for hiring me way back in 2001, when I was a brash programmer, eager to show how things should be done. And finally, I'd like to thank Bronwyn and the kids for putting up with the days and hours that I've had to avoid them to concentrate on the book!

About the Reviewers

Akash Mehta is a web application developer and technical author based in Australia. His area of work covers e-learning solutions, information systems, and developer training. He regularly writes web development articles for Adobe, CNet, the APC Magazine, and other print and online publications. He is a regular speaker at IT conferences, user groups, and BarCamps. Currently, Akash provides various services like web development, technical writing, consulting, and training through his website, <http://bitmeta.org/>.

Hailing from an igloo somewhere up in Canada, **Ashok Modi** is currently a systems analyst for California Institute of the Arts (<http://calarts.edu/>). He was a systems (and games) programmer for zinc Roe Design (<http://www.zincroe.com/>), and has been working with Drupal for the last three years. Starting from Drupal 4.6, he is the author of the abuse module (and looking for co-maintainers!), and has helped many maintainers in updating their contributed modules from 4.7.x to 5.x to 6.x. He also helped organize DrupalCamp Toronto in 2007 and 2008 and recently presented multiple sessions at DrupalCamp LA in 2009. In his spare time, Ashok tries to help contributed module maintainers with features and bugs. He was a technical reviewer on another book, *Drupal 6 Social Networking*, also published by Packt Publishing.

I would like to thank my wife, Richa, for letting me obsess over technology and pushing me to freelance and try other open-source technologies in my spare time.

John K. Murphy is a graduate of the University of West Virginia and has been wrapped up in computers and software development since the 1980s. When he is not buried in a book or jumping out of an airplane, he works as an IT consultant.

John lives with his wife and two children in Pittsburgh, PA and is currently obsessing about the Internet of Things.

Table of Contents

Preface	1
Chapter 1: Introduction and Overview	7
Expected developer skills	8
Differences between PHP and JavaScript	9
What is jQuery?	10
Why jQuery?	12
How does jQuery fit in with PHP?	13
How to work with the examples	14
Projects that use PHP and jQuery	16
WordPress	16
RoundCube	16
KFM	17
Drupal	18
Summary	18
Chapter 2: Quick Tricks	19
Dynamic select boxes	21
Client-side code	22
How it works	23
Server-side code	25
Quick deletes	26
Client-side code	27
Server-side code	30
Contextual help	31
Inline editing	36
Client-side code	36
Server-side code	41
Summary	41

Chapter 3: Tabs and Accordions	43
Using a jQuery tab plugin	44
Client-side code	45
Server-side considerations	49
Using jQuery accordion plugins	49
Client-side code	50
Server-side management of accordions	53
Client-side code	54
Server-side code	55
Just the conversion	56
Now show the data with the accordions	57
Server-side management of tabs	58
Server-side code	58
Just the conversion	59
Now show the data with the tabs	60
Using special codes to define tabs	61
Client-side code	63
Server-side code	63
Loading accordion panels through Ajax	66
Client-side code	67
Server-side code	69
Loading tab panels through Ajax	69
Client-side code	69
Summary	71
Chapter 4: Forms and Form Validation	73
Using the jQuery validation plugin	74
Client-side code	74
Setting up jQuery validation from PHP	77
Server-side code (setup)	77
Client-side code	78
Server-side code	79
Remote validation	81
Server-side code (setup)	81
Client-side code	82
Server-side code	83
Large select boxes	84
Client-side code	85
Server-side code	86

Auto-suggestion	87
Client-side code	88
Server-side code	89
Client-side code	90
Summary	91
Chapter 5: File Management	93
Security	93
Choosing a directory	95
Client-side code	95
Server-side code	97
Creating, renaming, and deleting directories	100
Client-side code	100
Server-side code	104
Moving directories	105
Client-side code	105
Server-side code	108
File uploads	109
Client-side code	109
Server-side code	112
Renaming, deleting, and moving files	113
Client-side code	114
Server-side code	115
File downloads	116
Client-side code	116
Server-side code	117
Summary	118
Chapter 6: Calendars	119
Displaying the calendar	120
Creating an event	121
Client-side code	122
Server-side code	124
Loading events from the server	126
Client-side code	126
Server-side code	127
Moving and resizing events	128
Client-side code	128
Server-side code	129

Editing events	129
Client-side code	130
Server-side code	132
Deleting events	132
Client-side code	133
Server-side code	134
Walk-through of the calendar so far	134
Recurring events	136
Client-side code	136
Server-side code	139
Deleting recurring events	141
Client-side code	141
Server-side code	142
Summary	143
Chapter 7: Image Manipulation	145
Showing the list of images	146
Server-side code	146
Client-side code	148
Selecting an image	149
Client-side code	149
Server-side code	150
Preparing your PHP	151
Rotating	152
Client-side code	152
Server-side code	154
Caching or saving the image	155
Client-side code	156
Server-side code	157
Resizing	159
Client-side code	160
Server-side code	162
Cropping	163
Client-side code	163
Server side	166
Summary	167

Chapter 8: Drag and Drop	169
Sorting simple lists	169
Client-side code	170
Server-side code	173
Client-side code	173
Sorting trees	175
Client-side code	175
Server-side code	181
Connecting lists	182
Client-side code	183
Server-side code	185
Summary	185
Chapter 9: Data Tables	187
Setting up DataTables	188
Client-side code	188
Getting data from the server	191
Client-side code	192
Server-side code	193
Caching your database queries	195
Server-side code	196
Pagination through Ajax	197
Client-side code	198
Server-side code	198
Sorting by column	200
Filtering	201
Server-side code	201
Setting a delay on the filter	203
Summary	204
Chapter 10: Optimization	205
Optimizing the page load	206
Place scripts at the bottom	206
Aggregate and cache your scripts	207
Output HTML similar to the post-jQuery code	208
Optimizing development and maintenance	208
Writing your own plugins	209
Un-caching your JavaScript	211

Optimizing jQuery	212
Use Google's CDN	212
Caching jQuery objects	214
Use \$.load for non-visual code	216
JavaScript optimizations	216
Chaining	216
Speeding up selectors	217
Inject multiple elements at once	218
Using setTimeout	219
Use var for your variables	221
Summary	223
Index	225

Preface

Creating PHP applications that respond quickly, avoid unnecessary page reloads, and provide great user interfaces, often requires complex JavaScript techniques. Even then, if you get that far, the results might not work across different browsers! With jQuery, you can use one of the most popular JavaScript libraries, forget about cross-browser issues, and simplify the creation of very powerful and responsive interfaces – all with the minimum of code.

This is the first book in the market that will ease the server-side PHP coder into the client-side world of the popular jQuery JavaScript library.

This book will show you how to use jQuery to enhance your PHP applications, with many examples using jQuery's user interface library jQuery UI, and other examples using popular jQuery plugins. It will help you to add exciting user interface features to liven up your PHP applications without having to become a master of client-side JavaScript.

This book will teach you how to use jQuery to create some really stunning effects, but without you needing to have in-depth knowledge of how jQuery works. It provides you with everything you need to build practical user interfaces, for everything from graphics manipulation and drag-and-drop to data searching, and much more. The book also provides practical demonstrations of PHP and jQuery and explains these examples, rather than starting from how JavaScript works and how it is different from PHP.

By the end of this book, you should be able to take any PHP application you have written, and transform it into a responsive, user-friendly interface, with capabilities you would not have dreamed of being able to achieve, all in just a few lines of JavaScript.

What this book covers

Chapter 1, *Introduction and Overview*, introduces you to what jQuery is, why you would want to use it, and why is it useful to combine it with PHP. Also, it gives a list of projects illustrating uses for combining PHP and jQuery.

Chapter 2, *Quick Tricks*, looks at a few quick examples on how to interface PHP and jQuery and a few tricks, which demonstrate how to relieve the most obvious resource wastes on the server.

Chapter 3, *Tabs and Accordions*, walks through the creation of tabs and accordions using the jQuery UI project, managing tabs and accordions using a rich text editor and a bit of PHP, and using Ajax to populate your accordion and tab panels.

Chapter 4, *Forms and Form Validation*, explores form validation using jQuery and PHP and how to use the same PHP configuration to validate on both sides—the server and the client side. It also covers examples on optimization of large select boxes and building auto-suggest fields.

Chapter 5, *File Management*, teaches you to manage almost everything—creating, uploading, editing, moving, downloading, and deleting files and directories.

Chapter 6, *Calendars*, builds a weekly calendar for you, which has events that can be created, edited, moved around, and deleted. It also takes care of recurring events.

Chapter 7, *Image Manipulation*, discusses methods for manipulating images with jQuery and PHP, along with some ways to make the changes non-destructive, so that multiple manipulations, such as select, rotate, resize, and crop, can be made on the same image.

Chapter 8, *Drag and Drop*, demonstrates a few uses of drag and drop, including sorting lists, dragging between lists, and hierarchical sorting, which can be used to improve the usability of your content management system.

Chapter 9, *Data Tables*, builds a very large data table and discusses how to navigate, sort, search, paginate, and search it using jQuery and Ajax.

Chapter 10, *Optimization*, shows the ways to optimize jQuery and various other elements of the web development environment.

What you need for this book

To try out the examples in this book, all you need is a test server; we recommend using PHP 5.2 or up. You will also need to download the jQuery library from http://docs.jquery.com/Downloading_jQuery.

Who this book is for

This book is for PHP application developers who want to improve their user interfaces through jQuery's capabilities and responsiveness. Whether you are familiar with jQuery or have only dabbled a little with JavaScript, this book will provide you with numerous practical examples on how to improve your application.

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: "The URL string we're working with includes a number of `sSearch` parameters."

A block of code is set as follows:

```
<?php
    $dir='.'; // directory containing the scripts
    $d=0;
    foreach (new DirectoryIterator($dir) as $file) {
        $d+=$file->getMTime();
    }
?>
```


When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in **bold**:


```
$.getJSON('./calendar.php?action=get_event&id='+calEvent.id,
function(eventdata) {
    var controls='<a href="javascript:calendar_delete_entry('
        +eventdata.id+') ">'
        + ' [delete] </a>';
    if(+eventdata.recurring) controls+='<br />'
        +'<a href="javascript:calendar_delete_reurrences('
        +eventdata.id+') ">'
        +'  [stop recurring] </a>';
    $('<div id="calendar_edit_entry_form"
```


Any command-line input or output is written as follows:

```
yum install php-pecl-imagick
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "When you click on this ? icon, smaller help icons will be shown next to all the elements in the page."

 Warnings or 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 send an email to feedback@packtpub.com, and mention the book title via 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/6989_Code.zip to directly download the example code.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration, and help us to improve subsequent versions of this book. If you find any errata, please 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 any list of existing errata. Any 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 us with the location address or web site name immediately so that 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 any aspect of the book, and we will do our best to address it.

1

Introduction and Overview

Welcome!

This book is a general overview of how to take advantage of jQuery's ease-of-use to make your PHP applications more "snappy" on the client side.

The book is aimed at the PHP developers who usually don't write client-side JavaScript, and would like an example-based introduction demonstrating a wide variety of integrations between server-side and client-side code.

For years, JavaScript (the language that jQuery is written in) has been seen as a "toy" language. However, the pervasiveness of Ajax has allowed it to break out of the client side and become an essential tool for all serious web application developers.

Ajax is a tool used by client-side code to contact the server and get or send data without needing to reload the entire page. It's used to update part of the page, or to retrieve data for a JavaScript application, or to send data to the server. Ajax stands for **Asynchronous JavaScript And XML**. However, the XML part is optional, and is often replaced with plain text or another data format such as **JSON** (explained later in this chapter).

This book will demonstrate how combining server-side PHP with client-side jQuery can make JavaScript much more useful than just for "form checking".

It is assumed that you are a server-side web application developer who wants to take a step into the client side and learn how to use JavaScript to make your applications more efficient and exciting for the client.

To be very clear about it, this book is not about PHP, and it is not about jQuery — it is about how to work with *both* PHP and jQuery. This book will teach you how to use jQuery to create some really cool effects, but without you needing to have in-depth knowledge of how jQuery works.

With this book, we are more interested in providing some practical demonstrations of PHP and jQuery with explanations, than in getting right down into the guts of how JavaScript works and how it is different from PHP.

After reading this book, you will have seen and built a wide variety of example applications. This would certainly be enough to immensely change how your current PHP applications work.

Each chapter in this book is dedicated to a specific subject, and will usually involve building a practical example, which you can use in your day-to-day work.

Expected developer skills

To read and work with this book's contents, it is expected that you are already a PHP developer with a basic understanding of JavaScript, and experience of HTML and CSS.

The book will not cover jQuery in depth—there are other books available that are dedicated to it. This book is designed to help a PHP developer write some immediately-useful client-side applications without needing weeks of study.

You need to understand how CSS selectors work. A CSS selector is the part of CSS that goes to the left of the { character. jQuery selects elements by using CSS selectors, and it's an extremely powerful way of choosing the elements that you wish to work on.

HTML is a must-have for every web developer. I think we can safely say that your HTML skills are not lacking, especially because that's the environment that PHP is usually working in.

Your PHP is expected to be good. You don't need to know everything about PHP, but it is beneficial if you have already written a few full PHP projects yourself and are comfortable with reading code and quickly understanding what it does.

You are not expected to be a good JavaScript writer. It will be useful, but the book is written such that beginners should not have a problem understanding how it all fits together. As a PHP developer, you will find that JavaScript is similar to PHP anyway, so you should not have a problem reading the examples and understanding them.

Differences between PHP and JavaScript

Syntactically, JavaScript and PHP are very similar. They're both **loosely typed** languages, and you can choose whether to write in an object-oriented or functional way.

However, there are some interesting differences, which a PHP developer may not have encountered before.

One example is asynchronous events. In PHP, everything happens linearly – when you call a function, the result is returned before the next line runs. However, in JavaScript, you can call some functions and have them return their results to a "callback" function a few seconds later.

The most obvious example is Ajax: when you request information from the server, it is bad design to have everything freeze while you're waiting for the data to return. Instead, you write your script such that the data is sent, and you carry on with other stuff. When the data returns, you handle it with another function.

This poses some interesting problems, such as race conditions, and how to pass variables through to remind the script what it was supposed to be doing when the Ajax request was fired.

A race condition occurs when one resource is accessed by two separate operations at the same time and one of them changes the value. This is because computers always do things one after the other; the change to the value will happen either before or after the other operation. This uncertainty means that in asynchronous systems, such as networks, care must be taken to make sure that things happen in the right order.

Race conditions are not a language-specific problem. They can happen anytime – say, when you open a file, work on it, and close it, only to find that at the time you were working on it, someone else came in and changed the name of the file. There are many ways of solving these problems depending on the nature of the case. The most common solution is to use a "lock", where if an operation is to write into a value, it will first "lock" it (usually by setting a flag variable or creating a `.lock` file) before it reads it, and then will "unlock" it (remove the file or unset the flag) after it has completed its work.

When an asynchronous function is called and its return value is sent to a callback function, that callback function must be reminded what the caller function was doing, so that it can carry on with it. This can be done using **closures**. With closures, the callback function is generated by the caller function, and has a copy of the caller's environment, including what variables were set, among other things. Later in the book, you will see some examples of this.

What is jQuery?

jQuery is a JavaScript library that makes it easy to work with the **Document Object Model (DOM)** of a website. jQuery is not a replacement for JavaScript. It is a JavaScript library, which gives some extra functionality that is not available natively in JavaScript itself.

jQuery is designed to make it easy to create, manipulate, or destroy elements in the document. Manipulation includes animation, CSS effects such as fades, resizes, and so on.

jQuery also makes it very easy to add behaviors to elements. So, you can do things like drag boxes around, have things happen when you hover a mouse cursor over something, have scripts run when a select box is changed, and so on.

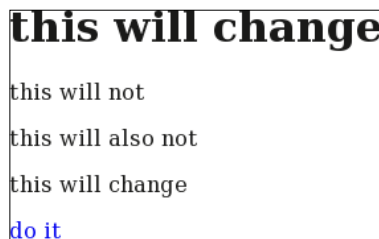
All of this can be accomplished in plain JavaScript if you feel the need to write it yourself, but there is no point in re-inventing the wheel. If there is a tool available that makes things easier for you, then you should not do it the hard way.

Besides, handwritten JavaScript tends to be much more verbose than it could be if you used a library, such as jQuery.

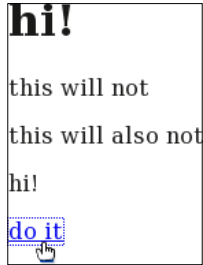
As an example, let's say that you want to get all of the `` elements in a page that are contained in an element with the `testme` class, and change their contents to the word **hi!**. Here is the HTML of the example:

```
<html>
  <head>
  </head>
  <body>
    <h1 class="testme"><span>this will change</span></h1>
    <p>this will not</p>
    <p class="testme">this will also not</p>
    <p class="testme"><span>this will change</span></p>
    <a href="javascript:run_test()">do it</a>
  </body>
</html>
```

When displayed in a browser, it will look like the following:



When the **do it** link is clicked, we want the view to change to this:



For the first test, here is how to do it in plain JavaScript. Place this code in the <head> section of the previous HTML code.

```
<script>
  function run_test(){
    var i,j,els,els2;
    els=document.getElementsByTagName('*');
    for(i in els){
      if(!/(^| )testme( |$)/.test(els[i].className))continue;
      els2=els[i].getElementsByTagName('span');
      for(j in els2){
        els2[j].innerHTML='hi!';
      }
    }
  }
</script>
```

It would be difficult to write this more compactly in pure JavaScript, but it's still too complex to be maintainable. Really, if you saw that for the first time, would you know straight away what it was trying to do? And, can you be sure that it will work in all browsers?

With jQuery, you can write a much more readable piece of code. Replace the above JavaScript with this:

```
<script src="../../jquery.min.js"></script>
<script>
  function run_test(){
    $(' .testme span').html('hi!');
  }
</script>
```


I know what you're thinking – where's the rest of it? Well, that's the whole thing. It very concisely strips away the confusion and all that you end up with is a very clear piece of code, which can be understood immediately. The problems of cross-browser compatibility are also solved by using the jQuery library. You can be sure that the new script will work as planned in all major browsers.

The example links to the jQuery library file, `jquery.min.js`, which you can download from <http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js>.

Why jQuery?

There are many libraries and frameworks available for JavaScript, including MooTools, Ext, Dojo, and Prototype. So, why use jQuery and not others? Here are some of the reasons:

- jQuery has a huge number of plugins available for everything you could imagine wanting to do online
- The information on the jQuery site is extremely well documented, with many examples
- jQuery does not extend the elements that it works on, which means that JavaScript such as `for(i in el){...}` will still work
- jQuery's CSS selector engine, **Sizzle**, is arguably the most complete and the quickest available

jQuery is available at Google's Ajax Libs CDN (<http://code.google.com/apis/ajaxlibs/>), so probably you already have it in your browser's cache.

There are benefits and detractors to everything. So in the end, it's a matter of taste.

In my case, I was using MooTools before I turned to jQuery, but MooTools' habit of extending every element it touches was interfering with my own code, so I had to replace it.

I chose jQuery because the body of knowledge on it was very large (every second article on JavaScript blogs appears to mention jQuery), it's impressively clean to work with, and there are a massive number of plugins available.

Also, because jQuery is used in so many large projects (a few of which are mentioned in the *Projects that use PHP and jQuery* section of this chapter). If you have ever worked on any of those projects, you will find that the skills you learned on them are transferable to other projects using jQuery.