

C o m m u n i t y   E x p e r i e n c e   D i s t i l l e d

# Programming with CodeIgniter MVC

Build feature-rich web applications using the CodeIgniter  
MVC framework

Eli Orr    Yehuda Zadik

**[PACKT]**  
PUBLISHING

# Programming with CodeIgniter MVC

Build feature-rich web applications using  
the CodeIgniter MVC framework

**Eli Orr**

**Yehuda Zadik**



BIRMINGHAM - MUMBAI

# Programming with CodeIgniter MVC

Copyright © 2013 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 authors nor Packt Publishing, and its dealers and 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 of 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: September 2013

Production Reference: 1160913

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-84969-470-4

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Abhishek Pandey ([abhishek.pandey1210@gmail.com](mailto:abhishek.pandey1210@gmail.com))

# Credits

**Authors**

Eli Orr

Yehuda Zadik

**Reviewers**

Jason Hamilton-Mascioli

Muhammad Faisal Shabbir

**Acquisition Editor**

James Jones

**Commissioning Editor**

Shreerang Deshpande

**Technical Editors**

Dylan Fernandes

Dipika Gaonkar

Kapil Hemnani

**Copy Editor**

Brandt D'Mello

Aditya Nair

Alfida Paiva

Laxmi Subramanian

**Project Coordinator**

Deenar Satam

**Proofreader**

Maria Gould

**Indexer**

Tejal Soni

**Graphics**

Ronak Dhruv

**Production Coordinator**

Melwyn D'sa

**Cover Work**

Melwyn D'sa

# About the Authors

**Eli Orr** brings over 30 years of experience in the high tech industry, mainly in software product development. Eli published articles in several magazines such as Wireless Systems Design, Telephony-Online, CommsDesign, EE Times, and for various emerging technologies. Currently, Eli is a PHP Advanced Web Application Developer, focused on CodeIgniter based projects for the last two years for rich functionality heavy-duty web applications.

Prior to that, as an entrepreneur, Eli established LogoDial Zappix Ltd., and had the role of CTO and server-side developer. Zappix enables dynamic visualization of call center voice menus, which are currently available in USA and Israel on iOS and Android-enabled smartphones.

Prior to that, Eli developed telecom solutions for Unified Communications with AT&T. Prior to that, Eli developed VoIP developer toolkits as a product manager with Radvision Ltd. During that period, he was an active member with the ITU-T Signaling Group 16 for defining VoIP protocol standardization, mainly the H.323.

Prior to that, Eli led development teams with IAI (Israel Aircraft Industries) ELTA based C and C++ programming languages. Eli can be contacted through his website, <http://EliOrr.com>.

---

Wring a book about the CodeIgniter Framework that I use daily and love was a great, thrilling challenge for me.

I would like to thank Yehuda Zadik who assisted me in writing this book as well as the Packt Publishing team for advising and assisting me through the entire book-writing process. In addition, I would like to thank Asher Efrati who is a strong CodeIgniter supporter, who assisted me by reviewing the book drafts and commenting on them. Finally, I would like to thank my daughter Hila Orr who supported me in my effort of writing this book.

---

**Yehuda Zadik** has over 20 years of experience in the IT industry, where he mainly specialized in software development based **object-oriented programming (OOP)** technologies.

Yehuda has over 8 years of experience developing with PHP OOP and open source Linux environments for developing web-based applications. Yehuda used state-of-the-art technologies for building dynamic web-based applications that were e-commerce enabled as well as social network integrated. Yehuda has a vast knowledge for integrating third-party plugins for network, mobile, and social environments' integration. Among the environments, Yehuda integrated with Facebook API, LinkedIn API, and various others.

Yehuda is an enthusiastic CodeIgniter developer who has been developing rich functionality and heavy traffic web-based applications over the last two years. Among Yehuda's clients are several major academic institutes. Yehuda can be contacted through his website, <http://yudazdk.co.il>.

---

Writing this book has been a challenging experience for me. My purpose was to write a practical book for developers that includes many examples.

First of all, I would like to thank my family members, my wife, Elana, and my son, Avishay, for their understanding and support during the writing of this book.

I would like to thank Eli Orr a lot for assisting me in writing the book and his helpful and fruitful feedback.

I would like to thank my clients: Omer Weissbein, CEO of Ontxt, for his advice and support; and Merav Babai, CEO of Pro Man and a LinkedIn expert, for her LinkedIn tips.

Finally, I would like to thank the Packt Publishing team for advising and assisting me throughout the book-writing process.

---

# About the Reviewers

**Jason Hamilton-Mascioli** leads the growth of `77robots.com`, the Canadian-based web development company he founded in 2005. Jason's role is to find and work with entrepreneurs to produce sustainable online businesses including solutions that aid early-stage startups.

With over 15 years as a senior web developer, Jason has worked with over 100 online startups globally, in addition to providing consulting services and mentorship to early-stage startups and entrepreneurs. For over 6 years, Jason has taught the Building Database-Driven Websites course at McMaster University Continuing Ed based in Hamilton, Ontario, Canada.

**Muhammad Faisal Shabbir** works as a senior software engineer at Strategic Systems International. Faisal has more than 6 years of extensive experience in software architecture, design, agile development, and deployment. Faisal completed his BS (Information Technology) from Virtual University. He can be reached at `faisal215@gmail.com`.

---

Special thanks to my mother, wife, and kids, who pushed me up to do such activities.

---

# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.





# Table of Contents

<b>Preface</b>	<b>1</b>
<b>Chapter 1: Getting Started</b>	<b>7</b>
Installing CodeIgniter	8
<b>Folders overview</b>	<b>9</b>
Mandatory components	9
<b>Example 1 – hello world</b>	<b>10</b>
The controller file	11
The view file	12
<b>Example 2 – passing the complex parameters to a view</b>	<b>12</b>
The controller file	13
The view file	14
<b>Example 3 – the database query by a model rendering results to a view</b>	<b>14</b>
The controller file	15
The model file	16
The view file	16
<b>Example 4 – interactive contact forms</b>	<b>17</b>
The controller file	18
The view file	19
<b>Summary</b>	<b>20</b>
<b>Chapter 2: Configurations and Naming Conventions</b>	<b>21</b>
<b>CI directory tree</b>	<b>22</b>
config.php	22
database.php	27
routes.php	29
<b>Defining and using your own configurations</b>	<b>29</b>
<b>Understanding and using CI naming conventions</b>	<b>30</b>
The main resource type naming rules	30
Controller definition naming rules	31

Example 1 – calling the controller index method	33
Example 2 – calling the controller and calc method without arguments	34
Example 3 – calling the controller and calc method with arguments	34
Example 4 – calling AJAX to an AJAX-oriented method with arguments	36
Loading libraries, models, and helpers	37
<b>Miscellaneous naming conventions</b>	<b>39</b>
<b>Summary</b>	<b>39</b>
<b>Chapter 3: Controller Usage and Scope</b>	<b>41</b>
<b>Scope of the CI controller</b>	<b>43</b>
<b>The user-defined CI controller</b>	<b>44</b>
Extending the CI controller	45
<b>CI controller use cases</b>	<b>46</b>
<b>Example 1 – the default homepage controller</b>	<b>49</b>
The controller file	49
The view file	50
The configuration file	51
<b>Example 2 – sending e-mails with attachments</b>	<b>52</b>
The controller file	52
<b>Example 3 – admin and regular user log in</b>	<b>55</b>
The controller file	56
The model file	59
The database file to upload for this example	60
The login_view view file	60
The login_in_view view file	62
<b>Summary</b>	<b>63</b>
<b>Chapter 4: Libraries</b>	<b>65</b>
<b>The CI libraries' scope and usage</b>	<b>67</b>
Available CI libraries	68
<b>Example 1 – using the built-in libraries</b>	<b>69</b>
The controller file	70
The view file	71
<b>Example 2 – using third-party libraries such as the Google Maps CI library wrapper</b>	<b>71</b>
The controller file	72
The view file	75
<b>Example 3 – building a library such as the Flickr API wrapper</b>	<b>76</b>
The flickr_wrapper.php library file	77
The flickr_recent.php controller file	82
The flickr_recent_view.php view file	84

---

<b>Example 4 – the LinkedIn API wrapper</b>	<b>85</b>
Requirements	86
Authentication flowchart	86
The linkedin_handler.php library file	88
The linkedinfo.php controller file	96
The linkedin-me.php view file	101
<b>Summary</b>	<b>105</b>
<b>Chapter 5: Helpers</b>	<b>107</b>
<b>CI helpers' scope and usage</b>	<b>108</b>
Available CI helpers	109
CI system helpers	109
CI third-party helpers	110
<b>Example 1 – using built-in helpers</b>	<b>110</b>
The controller file	110
The view file	111
<b>Example 2 – SSL helper</b>	<b>112</b>
The helper file	112
The controller file	113
The view file	114
<b>Example 3 – building your own helper</b>	<b>114</b>
The helper file	115
The controller file	117
The view file	118
<b>Summary</b>	<b>118</b>
<b>Chapter 6: Models</b>	<b>119</b>
<b>Scope of the CI model</b>	<b>120</b>
The model resource path	120
Loading a model	120
Using model methods	121
Connecting to a database	121
Business logic	122
<b>Object Relational Mapping (ORM)</b>	<b>122</b>
ORM simple operations example	124
<b>Example 1 – a CRUD example</b>	<b>125</b>
The controller file	126
The model file	129
The view file	131

<b>Example 2 – a business logic example</b>	<b>132</b>
The controller file	132
The model file	133
The view file	136
<b>Example 3 – retrieving data from Facebook</b>	<b>136</b>
The controller file	137
The model file	139
The view file	140
<b>Summary</b>	<b>141</b>
<b>Chapter 7: Views</b>	<b>143</b>
<b>Scope of the CI view</b>	<b>144</b>
The CI view resources path	145
The rendering flow	146
View flexibility	147
<b>Accessing the libraries/helpers</b>	<b>148</b>
Forms	148
AJAX	149
Parser configuration issues	152
Integrating jQuery or other client-side libraries	152
Plugins for rendering view	153
<b>Example 1 – HTML5 location powered by Google Maps</b>	<b>156</b>
The controller file	156
The view file	159
<b>Example 2 – user feedback powered by AJAX and the jQuery UI</b>	<b>161</b>
The ajax_handler.php controller file	162
The users_model.php model file	163
The logged_in_view.php view file	165
<b>Summary</b>	<b>168</b>
<b>Appendix</b>	<b>169</b>
<b>Index</b>	<b>171</b>

---

# Preface

This book aims to teach you how to develop web applications efficiently with the Ellis Labs CodeIgniter platform. The CodeIgniter platform is an object-oriented Model-View-Controller development platform. For more on MVC, please refer to <http://en.wikipedia.org/wiki/Model-view-controller>. The reader of this book is expected to be familiar with at least the PHP programming language, specifically with PHP OOP (object-oriented programming) and its usage, as well as with MySQL.

CodeIgniter (referred to as CI in this book) is an Application Development Framework, a toolkit for people who build websites and web applications using PHP. CodeIgniter is a smart application development skeleton framework, with flexible and expandable core powered high performance and low footprint. The CodeIgniter framework (OSL 3.0 open source license), developed and maintained by Ellis Labs, powers an echo system of developers across the globe. The first public version of CodeIgniter was released on February 28, 2006. It got very good feedback from web application professional developers. During November 2010, the CodeIgniter development project was added to the well-known GitHub community projects, and got increasing interest and usage by developers worldwide, as well as more and more third parties providing more add-ons with a better maturity and functionality set.

There is a rising trend of web applications based on **OOP (object-oriented programming)** frameworks using **MVC (Model-View-Controller)** development patterns, described in the next section, for developing advanced web applications. CodeIgniter is such a framework. It seems that CodeIgniter is continuously increasing its popularity as it has a simple yet high quality OOP core that enables great creativity, reusability, and code clarity naming conventions, which are easy to expand (user class extends CI class), while more third-party application plugins (including views/controllers/models/libraries/helpers providing application-oriented solutions such as CMS, shopping carts, or table grid navigators) and add-ons of libraries/helpers are becoming available.

The MVC concept is a development pattern or an application framework for a computer user interface that separates the representation of information from the user interacting with it. MVC has been adopted as a successful architecture for web application developments. The model consists of application data and provides services to manipulate them. The controller handles business rules and executes requests to the models and views. The controller mediates between the input, mostly received from a user interacting with a web browser that executed the rendered view. The browser runs a received rendered view by the controller through an HTTP protocol. The controller is the heart of the application. It performs model/database updates, business logic calculations, renders views to the user, and responds to an asynchronous **AJAX (Asynchronous JavaScript and XML)** request sent from the client side. The view code defines the presentation and user input logic to be rendered by the controller as HTML and JavaScript to the browser. The browser receives the rendered view via the HTTP response to be executed locally. The browser executing that content can present data, such as a mix of text, charts, diagrams, and images.

There are legacy **CMS (Content Management System)** web development platforms focused on CMS functionality and maintenance, such as a mature platform named DRUPAL. It might be very useful for content-oriented projects, but less appealing if the project aims to develop a new rich set of functionality, that is, web apps with many inputs and customized UI operations. If the project's requirements involve a low footprint and fast response/high performance, CodeIgniter is found to have excellent results.

To sum up in terms of flexibility, code reusability, light infrastructure, enabling developer creativity, code clarity, highest performance, minimal footprint, and fast learning curve, CodeIgniter seems to be the best choice. Furthermore, it is part of a proactive improvement process thanks to the growing developer's community worldwide.

## What this book covers

*Chapter 1, Getting Started*, introduces the CodeIgniter framework, while initially getting started with web-based applications.

*Chapter 2, Configurations and Naming Conventions*, reviews the CI naming convention rules, style guide, and spirit as well as the mandatory and optional configurations and usage within a CI project, with several examples. The practice of user-defined configurations will be reviewed as well.

*Chapter 3, Controllers*, reviews the CI controller and the user-defined controllers extending the CI controller. The CI controller class services, role, definition, usage, and scope will be reviewed with several examples to clarify.

*Chapter 4, Libraries*, reviews the user-defined libraries in a CI framework. Their services, role, definition, usage, and scope will be reviewed with several examples to clarify. Several examples for defining libraries and using them will be provided.

*Chapter 5, Helpers*, introduces you to the CI helpers and user-defined helpers' reusability value, definition rules, scope, and usage. Several examples for defining helpers and using them will be provided.

*Chapter 6, Models*, covers CI models and user-defined models' reusability value, definition rules, scope, and usage. Several examples for defining the models, extending the CI model, and using them will be provided.

*Chapter 7, Views*, explains the CI views concept as the generators for the client-side visualization and user interaction provided via HTTP. The view of the PHP part and scope, visual content (HTML/CSS), and program (JavaScript/AJAX/jQuery) for the client browser will be reviewed. The view's scope, definition, and controller rendering guidelines with practical practice and tips and tricks will be covered in this chapter.

*Appendix, Appendix References*, refers to recommended external resources related to CodeIgniter's formal resources as well as the ECHO system of the developer's community.

## What you need for this book

In order to understand this book's content, the user is required to at least have PHP programming language experience with some PHP OOP (object-oriented programming) and MySQL knowledge.

## Who this book is for

This book is for PHP web application developers who are interested in developing applications using OOP MVC concepts and specifically the CodeIgniter platform.

## 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: "We can include other contexts through the use of the `include` directive."

A block of code is set as follows:

```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,VoiceMail(u100)
exten => s,102,VoiceMail(b100)
exten => i,1,VoiceMail(s0)
```


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


```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,VoiceMail(u100)
exten => s,102,VoiceMail(b100)
exten => i,1,VoiceMail(s0)
```

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

```
# cp /usr/src/asterisk-addons/configs/cdr_mysql.conf.sample
   /etc/asterisk/cdr_mysql.conf
```

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: "clicking the **Next** button moves you to the next screen".

[  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 e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title through the subject of your message.

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](http://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

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

## 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 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 **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the Errata section of that title.

## 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 website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto: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](mailto: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

## Getting Started

This chapter covers the basics of the CI development framework and its usage by reviewing some fundamental web application examples. We will start with a basic hello world example and move to an interactive contact-form integration with a database. We will construct the CI applications by following a step-by-step method. Throughout this chapter, we need to remember that the CI development framework is an MVC-based development architecture (for more information, refer to the Wikipedia definition at <http://en.wikipedia.org/wiki/Model-view-controller>).

This chapter will primarily focus on the following topics:

- The CI project directory tree framework
- Configurations (routing and autoloading are covered in this chapter, while the other issues are covered in *Chapter 2, Configurations and Naming Conventions*)
- Example 1: hello world
- Example 2: passing parameters to a view
- Example 3: the database query by a model rendering results to a view
- Example 4: interactive contact forms

By reviewing these examples, we will get the basics of using CI resources. We will begin by briefly reviewing the CI resources used. Then we will review a web application code that loads a static view page. Next we will use the model to retrieve data from a database and show it in a view. Finally, we'll add a view with a contact form to enter input and save it by calling a controller method into the database.