

developerWorks  
中国

## 本文内容包括：

- MVC 框架如何提供帮助
- 安装和配置 CodeIgniter
- 您的第一个 CodeIgniter 项目
- 结束语
- 参考资料
- 关于作者
- 对本文的评价

## 相关链接：

- Web development 技术文档库
- Open source 技术文档库

developerWorks 中国 &gt; Web development | Open source &gt;

## CodeIgniter 入门

快速简单地创建 MVC 应用程序

级别：中级

[Thomas Myer](#), 作者、顾问和 Web 开发人员, Triple Dog Dare Media

2008 年 9 月 16 日

创建 CodeIgniter 应用程序要比您想象中的容易。我将引导您创建第一个项目：一个带有联系人表单的简单 Web 页面。

如果您是一位 PHP 开发人员，在使用 PHP 编程语言时不难发现：大型项目可能会变得复杂。

这并非 PHP 的缺点。是的，这种语言提供了丰富的特性，而且具有足够理想的弹性，可区分程序员之间的工作成果。在这种意义上，PHP 类似于 Perl，这是一些人喜欢它（而其他人鄙视它）的原因之一。任何曾经查看过遗留 PHP 项目的有经验的 PHP 开发人员都可以轻松判断出不同开发人员在项目不同阶段的工作 — 就像您是一位考古学家，正在凝视着深深的墓穴，见证着不同文化在各自时代的发展。

不管涉及了什么样式或使用了什么方法，超过几千行代码的 PHP 项目很容易在仓促之中变得凌乱不堪。这主要是因为它们在结构上不一致。一些程序员创建类来组织其工作，但似乎没有任何两个程序员对于如何编写类具有相同的看法。其他程序员构建大量充满函数的包含文件。还有其他人使用巨大、独立的库，比如 PEAR。

## MVC 框架如何提供帮助

在几年前，PHP 一直缺乏一个良好的、功能完善的模型-视图-控制器 (Model-View-Controller, MVC) 框架。MVC 框架允许程序员将其代码组织为三个不同的功能区：

- **模型** 包含与您的数据库和其他数据结构相关的所有代码。如果您具有一个名为 pages 的表，则您具有一个模型，其中具有用于从表中选择、创建、更新和删除记录的函数。
- **视图** 包含所有显示和 UI 元素 — JavaScript 代码、Cascading Style Sheets (CSS)、HTML 甚至 PHP。
- **控制器** 将一切联系在一起。控制器中的每个函数表示一个目的地或路线。如果您具有一个名为 /about 的目的地，则控制器将具有一个名为 about() 的函数。

如果以前没有使用过 MVC 框架，上述三点无法体现出这种组织模式的强大之处。一旦您开始用 MVC 思考，您对 PHP 开发的观点和态度将发生显著变化。

例如，不是在项目的每个可用角落中都塞入数据库查询代码，而是将一切都组织到模型中。为了从数据库表中选择页面，可以使用页面模型中的函数。

同样地，如果您需要更新特定页面的外观，可以使用视图，而不用与控制器打交道。与此类似，控制器是为您的应用程序添加目标和其他控制代码的位置；不必在模型中放入任何此类东西。

无论使用哪种 MVC 框架，在一天之内，您就会意识到您具有一个容易记住、可按需扩展的系统。如果客户在下周需要更改，没问题 — 您可以搞定。如果第二年有什么请求，同样如此。

## 约定优于配置 MVC

在所有 MVC 框架中，最著名的莫过于 Ruby on Rails。多年以前，它席卷 Web 开发领域，满足了所有人的想像。它并非纯粹的 MVC 框架，而是一种约定优于配置的 MVC 框架。

**约定优于配置** 意味着使用 Rails 时，您需要设置一些关键配置项（例如数据库的位置、特定用户名和路径），其他配置均由智能默认设置处理，您可在随后修改，也可不加修改。

结果不仅仅能得到组织良好的代码，而且还有速度极快、易于使用的 Web 开发环境。这都是 PHP 世界的梦想。经过一两年之后，许多类似于 Rails 的工具纷纷出现：CakePHP、Symfony 等等。

## 了解 CodeIgniter

最终，EllisLab 的工作人员发布了 CodeIgniter。许多企业尝试体验过所有 PHP MVC 框架之后，CodeIgniter 都成为赢家，主要是由于它为组织提供了足够的自由支持，允许开发人员更迅速地工作。

自由意味着使用 CodeIgniter 时，您不必以某种方式命名数据库表，也不必根据表命名模型。这使 CodeIgniter 成为重构遗留 PHP 应用程序的理想选择，在此类遗留应用程序中，可能存在需要移植的所有奇怪的结构。

CodeIgniter 不需要大量代码（1.6.2 版本仅为 2.8 MB，其中的 1.3 MB 是可以删除的用户文档），也不会要求您插入类似于 PEAR 的庞大的库。它在 PHP 4 和 PHP 5 中表现同样良好，允许您创建可移植的应用程序。最后，您不必使用模板引擎来创建视图 — 只需沿用旧式的 HTML 和 PHP 即可。

至此，我们已经提供了足够的介绍，下面来构建一个简单的项目，看看它的效果。

## 安装和配置 CodeIgniter

在构建任何 CodeIgniter 新项目时，第一步都是下载最新软件包（在本文撰写时，最新软件包是 1.6.2，请参见 [参考资料](#) 小节）。下载压缩存档文

## 文档选项

打印本页

将此页作为电子邮件发送

英文原文

件(.zip)并解压缩之后, 您就获得了一个 codeigniter\_<version\_number> 文件夹, 其中包括开始创建所必须的一切内容。

在进行一组必需的轻微的配置更改之前, 本节将为您介绍简单介绍 CodeIgniter, 使您熟悉它的基础知识。

## 文件夹结构

打开 CodeIgniter 文件夹时, 您会看到一个名为 system 的文件夹。所有 CodeIgniter 代码都将存放在这里。在此文件夹内还有一些文件夹, 其中有一个名为 application: 您要处理的 99.999% 的文件都将位于此文件夹内。该文件夹的命名十分贴切, 因为它包含您的应用程序及其所有组成部分 — system 文件夹的其他部分包括 CodeIgniter 核心代码和其他不应混淆的文件。

application 文件夹下又分为多个文件夹(参见图 1)。大多数文件夹易于理解。模型存放在 models 文件夹中、视图存放在 views 文件夹中、控制器存放在 controllers 文件夹中, 依此类推。还有一些文件夹用于存储 CodeIgniter 帮助程序和库的本地扩展, 这些内容不在本文讨论范围之内。

图 1. CodeIgniter 的文件夹结构



就目前而言, system/application 文件夹中最重要的文件夹就是 config。该文件夹内有两个需要关注的文件: config.php 和 database.php。

config.php 文件包含设置 CodeIgniter 所需的基本参数和自变量。database.php 文件包含连接数据库所需的基本参数和自变量。

就目前而言, 对于 config.php 文件, 您只需设置 base\_url 参数, 例如设置为 http://127.0.0.1/CodeIgniter/。根据您正在使用的服务器地址更改此设置:

```
$config['base_url'] = "http://www.example.com/";
```

务必牢记添加最后的斜杠, 即便是在子目录中设置 CodeIgniter 应用程序时也是如此。

接下来, 打开 database.php 文件, 为数据库服务器设置 connection 参数:

```
$db['default']['hostname'] = "your-db-host";  
$db['default']['username'] = "your-username";  
$db['default']['password'] = "your-password";  
$db['default']['database'] = "your-db-name";  
$db['default']['dbdriver'] = "mysql";
```

就是这样。您还可以进行其他一些设置(如自动加载首选项和特殊路径), 但只要 CodeIgniter 了解它位于何处且可连接其底层数据库, 您就可以放心开始编写代码了。

## 您的第一个 CodeIgniter 项目

现在, 您已经安装和配置了 CodeIgniter, 接下来即可构建项目, 这项工作至少要占用一个小时的时间。

这一次, 我们不会构建 Hello World 应用程序, 而是使用 CodeIgniter 创建一个简单的 Web 站点。该站点将有一个主页, 显示一些宣传文本和一个表单, 该表单将发布到数据库表中。无需为其外观费心 — 只需关注对应用程序有用的部分即可。换句话说, 让美工去关心外观 — 您只要确保一切可以正常工作、迅速完成即可。

按照 CodeIgniter 的术语, 可将这些需求转换为以下内容:

- 一个控制器, 仅包含少数功能(可使用默认的 Welcome 控制器)
- 一个模型(以及一个数据库表), 用于存储联系人信息
- 一个主视图, 包含一些支持

## 创建数据库表和模型

从模型入手可帮助您理解底层数据库表，之后再开始布设功能和 UI。如果对表将存储哪些内容认识不深，设计与表交互的表单将十分困难。

对于这个示例应用程序而言，您希望存储的是来自表单的联系人信息。那么需要的是哪些类型的联系人信息？目前而言，仅存储基本信息，要求提供姓名、电子邮件地址、电话号码和简短备注。您可能希望希望在后台存储时间戳和 IP 地址。

MySQL 表应如下所示：

```
CREATE TABLE `contacts` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(128) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `notes` text NOT NULL,  
  `stamp` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,  
  `ipaddress` varchar(32) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM;
```

现在我们已经有了表，接下来将创建第一个模型。在 `system/application/models` 文件夹内，创建一个 `mcontacts.php` 文件。为什么要将其命名为 `mcontacts`？这是一种速记形式 — 文件名中在模型名前加一个 `m` 可帮助您记住文件的组织方式，而不必使用更长的前缀或后缀，如 `model_` 等。

所有模型都采用相同的方式构造：

```
class MContacts extends Model{  
  
  function MContacts(){  
    parent::Model();  
  }  
}
```

请注意，类名与文件名匹配，必须为类提供一个构造函数。换句话说，一个模型就是一个 PHP 类。这也就意味着模型中的所有函数实际上都是该类的一个方法。

理解这一点之后，很快就能领悟到，需要具备一个函数，将数据安全地插入联系人数据库表中。提供此功能的函数如下：

```
function addContact(){  
  $now = date("Y-m-d H:i:s");  
  $data = array(  
    'name' => $this->input->xss_clean($this->input->post('name')),  
    'email' => $this->input->xss_clean($this->input->post('email')),  
    'notes' => $this->input->xss_clean($this->input->post('notes')),  
    'ipaddress' => $this->input->ip_address(),  
    'stamp' => $now  
  );  
  
  $this->db->insert('contacts', $data);  
}
```

注意，您获取了 POST 数组的输出、整理并将其存储在名为 `contacts` 的数据库表中。在此过程中，您使用了多个帮助程序来简化工作。

例如，`$this->input->xss_clean()` 整理表单字段的数据、`$this->input->post()` 简化对这些表单字段的访问、`$this->input->ip_address()` 从用户的浏览器获取 IP 地址、`$this->db->insert()` 向数据库表添加一条新记录。

在这种上下文中，`$this->input->xss_clean()` 的使用必不可少 — 您正在处理 Web 用户输入，那可以是任何内容。使用 `xss_clean()` 函数或许是最基本的应对方法，您可能还要考虑应用更加稳妥的措施。添加功能来将字段长度缩短到一定大小可能也是一种合理做法。但就目前而言，`xss_clean()` 例程即可为您提供足够的保护。

您只用了短短几分钟就创建了一个可重用的函数，允许在数据库中存储联系人信息。现在，我们将转而讨论控制器。

## 初始化控制器

在 CodeIgniter 中，控制器用于组织项目。设想每个函数都是站点或应用程序的一个页面或目标。如果使用主页，就需要一个 `index()` 函数。如果有一个 About 页面，就需要 `about()` 或 `about_us()` 函数 — 具体取决于您希望怎样构造 URL。

甚至可以将控制器组织到文件夹中，创建层次结构。例如，在 `system/application/controllers` 文件夹中，可能有一个 `admin` 文件夹，其中包含针对管理工具各主要部分的控制器。您可按照如下方法访问这些控制器（和函数）：`http://www.example.com/admin/controller-name/function-name/`。

目前只需使用默认控制器，即 `Welcome` 控制器。它存储在 `system/application/controllers/` 文件夹中，名为 `welcome.php`。打开时，应看到以下内容：

```
class Welcome extends Controller {  
  
  function Welcome(){
```

```

parent::Controller();
}

function index(){
    $this->load->view('welcome_message');
}
}

```

如您所见，类名反映了文件名。这里也有一个构造函数，调用 CodeIgniter 内核中的父 Controller 类。了解这些就够了。

接下来，注意名为 index() 的启动函数，它将加载 welcome\_message 视图。在删除此函数并编写您自己的函数之前，有必要注意，此原型 index() 函数很好地满足了为应用程序的最终用户显示信息的最低要求。

我们继续构建一个新的 index() 函数。首先需要加载有用的 Form — 它能帮助您完成创建联系人表单的繁琐任务。

下面，设置可在视图内部使用的多个变量 — 通过这种方法，即可更好地组织应用程序。例如，您可能希望在控制器中设置标题和标题栏。如果要这样做，就必须将变量载入视图。所载入的变量之一就是所包含视图的名称。通过这种方法，即可设置包含所有外观的主视图，以及包含内容的各包含项：

```

function index(){
    $this->load->helper('form');
    $data['title'] = "Welcome to our Site";
    $data['headline'] = "Welcome!";
    $data['include'] = 'home';
    $this->load->vars($data);
    $this->load->view('template');
}

```

\$data 数组被传入到一个称为模板的视图(接下来即将构建此视图)。数组内的信息可使用键名访问，如果希望输出标题栏，通过 \$headline 访问它即可。

接下来，您将创建模板和主页视图(后者只是一个包含项)，并完成控制器。

## 创建视图

您的第一个视图极为简单 — 这是一个名为 template 的视图。我们将尽力保持其简单，展示视图可以有多么灵活。template 视图存储为 system/application/views 中的 template.php，初始形式如下所示：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title></title>
</head>
<body>

</body>
</html>

```

但应牢记，您正在传入三个变量：\$title、\$headline 和 \$include(一个包含项的名称)。下面是添加了粗体所示内容后的 template 视图：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title><?php echo $title;?></title>
<style>label { display:block;}</style>
</head>
<body>
<h1><?php echo $headline;?></h1>
<?php $this->load->view($include);?>
</body>
</html>

```

在添加的前两条语句中，您将分别显示在 \$data['title'] 和 \$data['headline'] 中找到的数据。随后使用 \$data['include'] 的值载入第二个视图。在本例中是一个名为 home 的视图(另请注意，为了简化后续的一些工作，我们还添加了少许 CSS 代码)。

如果需要调用，那么最好首先进行构建。下面就是一个简单的视图，其中包含一个文本块和一个从站点访问者处收集信息的表单：

```

<p>This is random text for the CodeIgniter article.

```

```

There's nothing to see here folks, just move along!</p>
<h2>Contact Us</h2>
<?php
echo form_open('welcome/contactus');
echo form_label('your name','name');
$data = array('name' => 'name', 'id' => 'id', 'size' => '25');
echo form_input($data);

echo form_label('your email','email');
$data = array('name' => 'email', 'id' => 'email', 'size' => '25');
echo form_input($data);

echo form_label('how can you help you?','notes');
$cdata = array('name' => 'notes', 'id' => 'notes', 'cols' => '40', 'rows' => '5');
echo form_textarea($cdata);

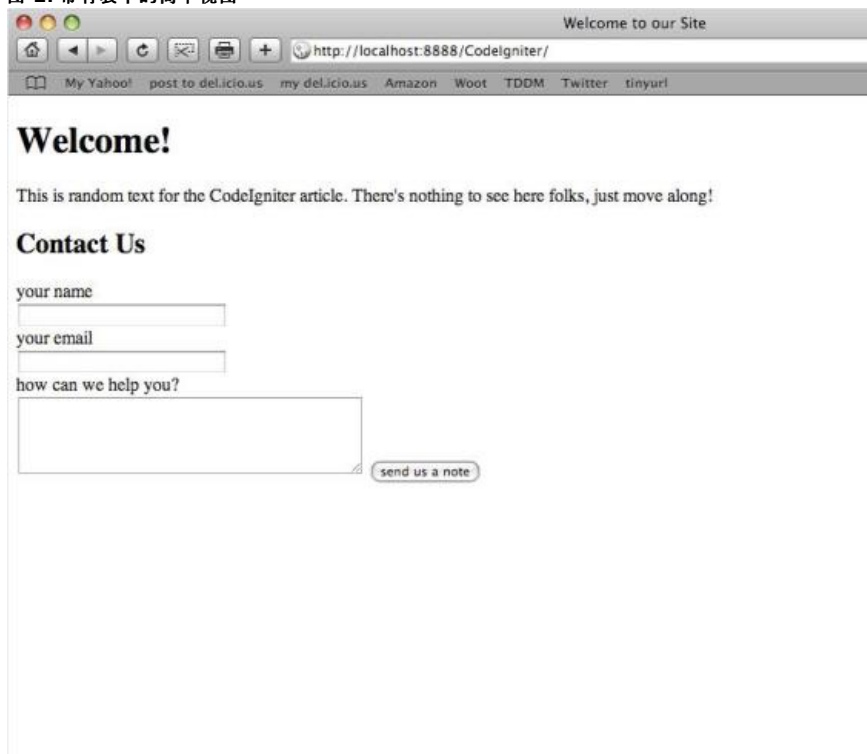
echo form_submit('submit','send us a note');
echo form_close();

?>

```

图 2 显示了将所有这些内容载入浏览器后的效果。

图 2. 带有表单的简单视图



同样，您使用了有用的 CodeIgniter 快捷方式集。这一次，使用的是 Form 帮助程序，将其载入控制器。form\_open() 函数允许打开表单 — 它具有必要的自变量，即表单发布的目标位置。下面，您将返回控制器并添加 contact() 函数来处理表单发布数据。

在表单中，您使用了 form\_label() 来创建可访问的标签，使用 form\_input() 和 form\_textarea() 来构建表单字段和文本区，使用 form\_submit() 来构建输入按钮。请注意，通过 form\_input() 和 form\_textarea() (以及其他表单函数)，您就可以传入一个信息数组，帮助跟踪字段名称、id、大小和其他信息。

最后，使用 form\_close() 关闭表单。

让我们回过头来完成控制器。

### 完成应用程序

现在已经有了两个视图，因而需要重新回到控制器，为其添加两个函数。您已经了解了第一个函数：即处理主页上的表单传入的 POST 的 contactus() 函数。第二个是 thankyou() 函数，它将用作该表单的最终确认页面。

contactus() 函数非常简单。载入 MContacts 模型，运行该模型内的 addContact() 函数，然后将用户转向 thank-you 页面。请注意，要使用 redirect() 函数，必须载入 URL 帮助程序。

代码如下所示：

#### 为什么要使用 Form 帮助程序？

或许您会感到疑惑，为什么要花时间来使用 Form 帮助程序。您可能充满怀疑：如果在 Web 站点方面经验丰富，为什么还需要使用表单帮助程序？归根到底，一切都与迅速、高效地完成工作有关，Form 帮助程序 (以及其他帮助程序) 能消除在使用 HTML 时遇到的大量繁杂工作。

```
function contactus(){
    $this->load->helper('url');
    $this->load->model('MContacts', TRUE);
    $this->MContacts->addContact();
    redirect('welcome/thankyou','refresh');
}
```

thankyou() 函数如下所示:

```
function thankyou(){
    $data['title'] = "Thank You!";
    $data['headline'] = "Thanks!";
    $data['include'] = 'thanks';
    $this->load->vars($data);
    $this->load->view('template');
}
```

一切都非常简单, thanks 视图如下所示:

```
<p>Thanks so much for contacting us. Someone will be in contact with you soon.</p>
```

您可能感到迷惑, 为什么要为如此简单的一个视图浪费时间。为什么不在控制器中设置一个变量再运行它? 当然可以那样做, 但分离函数组件总是最佳做法。通过这样的方式, 就不存在遇到任何麻烦的风险。

## 添加安全性

现在, 还有一项工作需要完成。在 Welcome 控制器的 contactus() 函数中, 有着在数据库中创建多条空记录的风险 — 这会导致某人连续将联系人目标载入其浏览器或使用某种类型的机器人。

要避免此类情况发生, 最简单的方法就是在控制器中添加简单的测试。如果存在 POST 数据, 则载入模型和函数。如果没有, 则将其返回主页。改写后的函数如下所示:

```
function contactus(){
    $this->load->helper('url');
    if ($this->input->post('email')){
        $this->load->model('MContacts', TRUE);
        $this->MContacts->addContact();
        redirect('welcome/thankyou','refresh');
    }else{
        redirect('welcome/index','refresh');
    }
}
```

## 结束语

在不到一个小时的时间里, 您安装并配置了 CodeIgniter, 创建了一个包含主页、将信息添加到数据库的表单和 thank-you 页面的 Web 站点。

当然, 要学习的东西还有很多。例如, 您可自动加载所需的模型和任何帮助程序或库。可以为应用程序调整缓存和性能。可以为视图添加更多 CSS 内容。可以添加在数据库插入操作结束后发送电子邮件通知的功能。

目前, 您只是掌握了开始使用 CodeIgniter 所需的一些内容。

## 参考资料

学习

- 您可以参阅本文在 developerWorks 全球网站上的 [英文原文](#)。
- 随时关注最新的 [developerWorks 技术活动和网络广播](#)。
- 利用 developerWorks [Web 开发专区](#) 内的专业 Web 技术文章和教程拓展您的 Web 开发技能。

获得产品和技术

- 下载 [CodeIgniter 项目](#) 的最新软件包(在本文撰写时, 最新版本是 1.6.2)。
- 阅读 [在线 CodeIgniter 文档](#)。
- 访问 [CodeIgniter 论坛](#)。
- 访问 [CodeIgniter wiki](#)。

## 讨论

- 通过参与 [developerWorks blogs](#) 加入 developerWorks 社区。

## 关于作者

Thomas Myer 是一名技术书籍作者、顾问和 Web 开发人员。在 2001 年，他在德克萨斯州奥斯汀创立了 Triple Dog Dare Media。Triple Dog Dare Media 帮助公司创建基于 CodeIgniter 的发布应用程序，比如内容管理、门户网站和电子商务系统。Myer 是 *No Nonsense XML Web Development with PHP* (Sitepoint, 2004) 和 *Lead Generation on the Web* (O'Reilly 2007) 的作者。他的新书，WROX 的 *Professional CodeIgniter* 在 2008 年 7 月在书店上架。他还为 IBM developerWorks、Amazon Web Services、AOL、Darwin Magazine 等撰写了数十篇技术和商业文章。

## 对本文的评价

- 太差！(1)
- 需提高(2)
- 一般；尚可(3)
- 好文章(4)
- 真棒！(5)

建议？

反馈意见

[↑ 回页首](#)

IBM 公司保留在 developerWorks 网站上发表的内容的著作权。未经 IBM 公司或原始作者的书面明确许可，请勿转载。如果您希望转载，请通过 [提交转载请求表单](#) 联系我们的编辑团队。

[关于 IBM](#) | [隐私条约](#) | [联系 IBM](#) | [使用条款](#)