

第20章 PHP与MVC开发模式

- MVC是一种源远流长的软件设计模式，早在上个世纪70年代就已经出现了基于MVC的开发模式。随着Web应用开发的广泛展开，也因为Web应用需求复杂度的提高，MVC这一设计模式也渐渐被Web应用开发所采用。
- 随着Web应用的快速增加，MVC模式对于Web应用的开发无疑是一种非常先进的设计思想，无论选择哪种语言，也无论应用多复杂，它都能为构造产品提供清晰的设计框架。MVC模式会使得Web应用更加强壮，更加有弹性，也更加个性化。本章先向读者介绍什么是MVC，然后介绍几款PHP中常见的基于MVC的开发框架及其应用。



20.1 什么是MVC模型

- MVC模型是开发大型Web应用时可以采用的程序架构。MVC是Model_View_Control的缩写，简单地讲，Model即程序的数据或数据模型，View是程序视图界面，Control是程序的流程控制处理部分。
- Model_View_Control是软件设计的典型结构。如今这一设计思想也开始在Web开发中实践并流行起来。在这种设计结构下，一个应用被分为三个部分：model、view和controller，每个部分负责不同的功能。model是指应用程序的数据，以及对这些数据的操作，view是指用户界面，controller负责用户界面和程序数据之间的同步，也就是完成两个方向的动作，这个两个动作如下所示。
- 根据用户界面（view）的操作完成对程序数据（model）的更新。
- 将程序数据（model）的改变及时反应到用户界面（view）上。



20.2 MVC模型的组成

- MVC是一个设计模式，它使Web应用程序的输入、处理和输出分开进行。MVCWeb应用程序被分成3个核心部件：模型（Model——M）、视图（View——V）、控制器（Controller——C）。一个好的MVC设计，不仅可以使模型、视图、控制器高效完成各自的任务处理，而且可以让它们完美地结合起来，完成整个Web应用。



20.2.1 控制器（Controller）

- 控制器负责协调整个应用程序的运转，简单来讲，控制器的作用就是接受浏览器端的请求。它接受用户的输入并调用模型和视图去完成用户的需求，当用户点击Web页面中的超链接或发送HTML表单时，控制器本身不输出任何东西，它只是接收请求并决定调用哪个模型构件去处理浏览器端发出的请求，然后用确定用哪个视图来显示模型处理返回的数据。



20.2.2 数据模型 (Model)

- 通常，Web应用的业务流程处理过程对其它层来说是不可见的，也就是说，模型接受视图请求的数据，并返回最终的处理结果。
- 数据模型的设计可以说是MVC最主要的核心。对一个开发者来说，需要专注于Web应用的业务模型的设计。MVC设计模式把应用的模型按一定的规则抽取出来，抽取的层次很重要，抽象与具体不能隔得太远，也不能太近。MVC并没有提供模型的设计方法，只是用来组织管理这些模型，以便模型的重构和提高重用性。从面向对象编程来做将，MVC定义了一个顶级类，在告诉它的子类有哪些是可以做的。这点开发人员非常重要。



20.2.3 视图 (View)

- 从用户角度说，视图就是用户看到的HTML页面。从程序角度说，视图负责生成用户界面，通常根据数据模型中的数据转化成HTML输出给用户。视图可以允许用户以多种方式输入数据，但数据本身并不由视图来处理，视图只是用来显示数据。在实际应用中，可能会有多个视图访问同一个数据模型。比如说“用户”这一数据模型中，就有一个视图显示用户信息列表，还有管理员使用用于查看、删除用户的视图。这两个视图同时访问“用户”这一数据模型。
- 在很多Web开发中，都会使用模板来生成用户最终看到的HTML页面。关于模板的有关知识，将在下节介绍。



20.3 PHP开发中的模板技术

- 在基于MVC模型的Web应用开发中，模板是不可或缺的。模板定义了一个并不完全的类HTML文件，它为用户视图提供了最基本内容的框架，一些重要的数据需要在程序中添加模板中，从而形成完整的用户视图。本节将首先向读者介绍模板的基本概念和其在PHP程序中的用法，然后介绍一个优秀的模板引擎——Smarty。



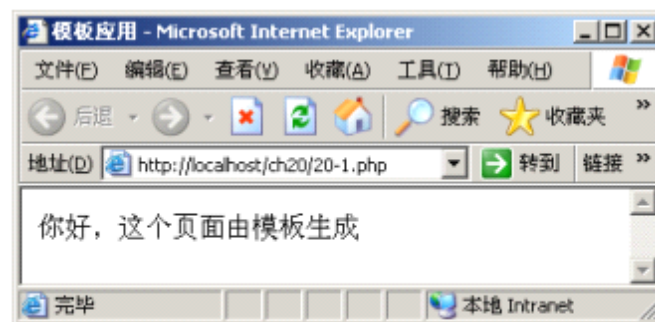
20.3.1 什么是模板

- 对于什么是模板这个问题，可以这样解释：模板是一组插入了HTML的PHP脚本或者说是插入了PHP脚本HTML，通过这种插入的内容来表示变化的数据。
- 当用户浏览时，由PHP程序文件打开该模板文件，将模板文件中定义的变量进行替换，动态生成内容，从而向用户显示一个完整的HTML页面。本例中的模板变量就是{greetings}和{pagetitle}，这两个个变量是在PHP程序使用该模板时，根据具体的内容来替换。下一小节将会看到处理该模板文件的具体解决办法。



20.3.2 在PHP程序中使用模板

- 这一节通过一个具体实例来演示如何在 PHP 程序中使用模板文件。首先，需要定义一个模板文件，
- 虽然这个在PHP程序中使用模板变量的示例程序很小，但却体现了模板在PHP程序中的处理思想。当然实际的模板引擎要比这个复杂得多，也更能满足实际需要。本章的下一小节将会为读者介绍一个被 PHP 官方推荐使用的模板引擎，并通过一些实例讲解模板引擎的使用。



在 PHP 程序中使用模板



20.3.3 Smarty模板引擎介绍

- 对PHP来说，有很多模板引擎可供选择，比如最早的 PHPLIB template和后起之秀Fast template，经过数次升级，已经相当成熟稳定。本小节要介绍的是一款易于使用且功能强大的PHP模板引擎——Smarty。它分开了逻辑程序和外在的内容，提供了一种Web页面易于管理的方法。Smarty显著特点之一是“模板编译”，这意味着Smarty读取模板文件然后用他们创建PHP脚本。这些脚本创建以后将被执行，而不是去解析模板文件的语法。可以通过 Smarty的官方网站获取Smarty模板引擎，官方网站的网址是 <http://www.smarty.net/>。下面关于Smarty的介绍将以稳定的2.5版本为准。
- 下载Smarty安装包，解压后有3个目录：libs模板文件目录，其下有4个类文件1个目录。首先介绍最的是Smarty.class.php，它应该是整个Smarty模板的核心类，通常，需要在Web应用程序目录下建立如下所示的目录结构。



20.3.4 Smarty模板引擎的使用

- 下面通过一个实例程序介绍Smarty模板引擎在PHP程序中的使用。首先定义一个简单模板文件，命名为20-2.tpl，并保存在当前目录下的template子目录下。tpl是Smarty模板文件使用的后缀名，



使用 Smarty 模板引擎输出的 HTML 页面



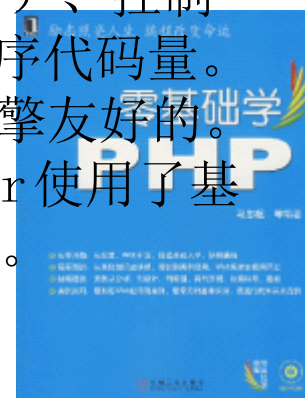
20.4 常见的基于MVC的PHP开发框架简介

- PHP开发永远是一个活跃的领域，随着MVC设计方法、敏捷开发理念的流行，Web应用领域产生了大量的开发框架，使用这些框架可以迅速搭建Web应用，降低开发成本和缩小开发周期。
- PHP社区也出现了大量的MVC开发框架，本节向读者介绍4种比较活跃的PHP开发框架：CodeIgniter、CakePHP、ZendFrame和FLEAPHP。这些框架都有各自的特点与不足，而且它们有各自的设计目标和设计理念，这决定了它们有其适应的范围。实际开发中，应该根据具体的需求和应用环境选择适合的开发框架。



20.4.1 CodeIgniter

- CodeIgniter 是一个小巧但功能强大的有PHP编写的基于MVC的Web应用开发框架，作为一个简单而不是强大的开发工具包，它可以为PHP程序员建立功能完善的Web应用程序。CodeIgniter它的确是一个不错的MVC框架。这里还需要说明一点，CodeIgniter是经过Apache/BSD-style开源许可授权的免费框架。
- CodeIgniter使用了模型（Model）、视图（View）、控制器（Controller）的方法，最小化了模板中的程序代码量。CodeIgniter生成的URL非常干净而且是对搜索引擎友好的。不同于标准的“字符串查询”方法，CodeIgniter使用了基于段的（segment-based）URL表示法，如下所示。
- www.mysite.com/aaa/bbb/123



20.4.2 CakePHP

- 第2个要介绍的PHP开发框架是CakePHP。CakePHP封装了数据库访问逻辑，对于小应用来说可以获得令人惊叹的开发效率。CakePHP比较有特色的地方是命令行代码生成工具让开发者可以快速生成应用程序框架。如果读者了解Ruby on Rails，在使用CakePHP构建Web应用之后，会发现CakePHP几乎就是Rails在PHP上的翻版。CakePHP也是完全基于MVC架构的Web开发框架，它有以下一些特点。
- 兼容PHP4和PHP5。
- 数据库交互和简单查询的集成。
- MVC体系结构。



20.4.2 CakePHP

- 自定义的URL的请求分配器（Request dispatcher）。
- 内置验证机制。
- 快速灵活的模版。
- 支持AJAX。
- 灵活的视图缓存。
- 可在任何Web站点的子目录里工作，不需要改变Apache配置。
- 命令行生成Web站点框架。



20.4.3 Zend Framework

- Zend Framework是完全基于PHP语言的针对Web应用开发的框架，与众多的其他PHP开发框架相比，Zend Framework是一个PHP“官方的”的框架，它由Zend公司负责开发和维护。Zend Framework同样基于MVC模式，Zend Framework采用了ORM思路，所谓ORM思路及时Object Relational Mapping，即对象关系映射，这是一种为了解决面向对象编程与关系数据库存在的互不匹配现象的技术。简单地说，这种技术将数据库中的一个表映射为程序中的一个对象，表中的字段映射为对象的属性，然后通过提供的方法完成对数据库的操作。就这一点而言，Zend Framework很相似于现在流行的非PHP的开发框架Ruby on Rails。另外，上一小节介绍的CakePHP也实现了这种技术。
- Zend Framework的另一个特点是，它实现了Front Controller模式，也就是说，所有的HTTP请求都会转发到同一个入口，然后再有路由功能模块转到相应的Controller。



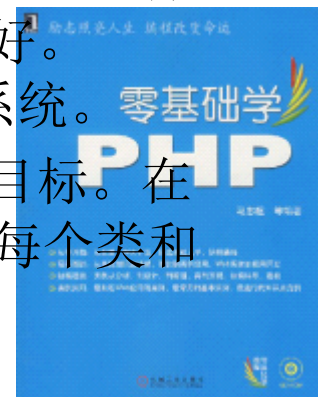
20.5 CodeIgniter框架应用

- 从上节介绍的4个框架中，本书选择CodeIgniter作为讲解实例。通过前面的几小节内容，读者已经了解到CodeIgniter是一个为用PHP编写Web应用程序的人员提供的工具包。它的目标是实现比从零开始编写代码更快速地开发项目，为此，CodeIgniter提供了一套丰富的类库来满足通常的任务需求，并且提供了一个简单的接口和逻辑结构来调用这些库。CodeIgniter可以将需要完成的任务代码量最小化，这样开发人员就可以把更多的精力放到项目的开发上了。另外，CodeIgniter提供了非常完善的文档，读者通过这些文档可以快速学习、理解CodeIgniter，并且可以在开发中高效使用CodeIgniter框架。



20.5.1 CodeIgniter的技术特点介绍

- CodeIgniter在设计之初就有其明确的目标，这个目标就是在最小化，最轻量级的开发包中得到最大的执行效率、功能和灵活性。为了这个目标，CodeIgniter在开发过程的每一步都致力于基准测试、重构和简化工作，拒绝加入任何不助于目标的东西。从技术和架构角度看，CodeIgniter按照下列目标创建。
- 动态实例化，在CodeIgniter中，组件的导入和函数的执行只有在被要求的时候才进行，而不是在全局范围。
- 松耦合，耦合是指一个系统的组件之间的相关程度。组件互相依赖越少，那么系统的重用性和灵活性就越好。CodeIgniter的目标就构建是一个非常松耦合的系统。
- 组件单一性。单一是指组件有一个非常小的专注目标。在CodeIgniter 里面，为了达到最大的用途最大的每个类和它的功能都是高度自治的。



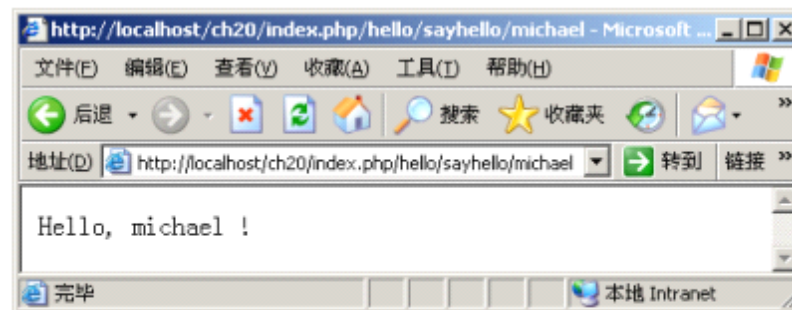
20.5.2 安装CodeIgniter

- CodeIgniter 安装分为如下所示的 4 个步骤。
- （1）解压缩安装包。
- （2）把CodeIgniter文件夹和里面的文件部署到服务器，通常index.php位于应用服务器的根目录。
- （3）编辑application/config/config.php 文件，设置基本URL。
- （4）如果需要使用数据库，编辑 application/config/database.php，在这个文件中设置数据库参数。
- 下面介绍CodeIgnite框架的组织结构。从官方网站下载CodeIgniter框架，解压缩后可以看到最主要的一个目录是system，该目录就是CodeIgnite框架的目录。除此之外还有一个index.php文件，该文件用来做一些初始化配置，它还起到处理HTTP请求的作用。



20.5.3 CodeIgniter的Controller（控制器）

- 在CodeIgnite中，一个controller就是一个类文件，controller所属的类和普通的PHP类几乎没有区别，唯一有特点的是Controller类的命名方式，它所采用的命名方式可以使该类和URI关联起来。例如下面个URL地址，就说明了这个问题。
- `www.mysite.com/index.php/news/`



向方法传递参数



20.5.4 CodeIgniter的Model（数据模型）

- 在CodeIgniter中，Model是专门用来和数据库打交道的 PHP 类。通常在Model类里包含插入、更新、删除数据的方法。CodeIgniter中的Model类文件存放在 application/models/ 目录，可以在里面建立子目录。最基本的 Model定义如下面的代码所示。

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



20.5.5 CodeIgniter的View（视图）

- 在CodeIgniter中，视图从不直接调用，必须被一个控制器来调用。
- （1）使用文本编辑器创建一个名为helloview.php的文件，
- （2）将该代码保存到application/views/目录下。然后，需要使用某个方法载入该视图文件。这个方法的使用法如下所示。
- `$this->load->view('name')`
- （3）这行代码中，name是需要载入的视图文件的名称，文件的后缀名没有必要写出。接下来，在hello控制器的文件hello.php中，写入这段用来载入视图的代码，



20.5.5 CodeIgniter的View（视图）

- （4）此时再通过地址
`http://localhost/ch20/index.php/hello` 浏览
`hello.php`,
- （5）通过这段代码，读者了解了如何载入一个视图。但视图中经常需要动态数据的内容，下面就介绍如何处理含有动态数据的视图。动态数据通过控制器以一个数组或是对象的形式传入视图，这个数组或对象作为视图载入方法的第二个参数，
- （6）此时访问`hello.php`会看到执行结果，浏览器上的页面标题和页面的heading文字都更换成动态数据内容。
- （7）最后还需要修改`helloview.php`，将在其中添加输出数据的PHP代码，修改后按`helloview1.php`保存在
`application/views/`目录下，修改后的代码如下所示，



20.6 小结

- 本章介绍了PHP中的MVC模型。首先介绍了什么是MVC模型，以及MVC模型中控制器、视图和数据模型的概念。然后介绍了PHP中的模板技术，包括什么是模板、如何在PHP程序中使用模板、Smarty模板引擎的基本用法。接着，向读者介绍了几款目前比较流行的PHP基于MVC的Web开发框架，包括：CodeIgniter、CakePHP、Zend Framework以及国产优秀框架FleaPHP。最后，以CodeIgniter为实例，介绍了使用CodeIgniter开发PHP网络应用程序的基本思路和用法。

