

第 1 章 symfony 介绍

symfony 能做什么？使用 symfony 需要掌握哪些知识？这一章会解答这些问题。

symfony 简介

开发框架可以自动化地完成一些特定的开发模式来简化应用程序开发过程。开发框架还使程序代码结构更好，通过开发框架开发人员可以写出更好、更易读的、更容易维护的代码。总之，开发框架简化了编程，因为它把复杂的操作封装成了简单的语句。

symfony 是一个完整的 web 应用开发框架，它为加速开发提供了几个关键功能。首先，它把 web 应用的业务规则，服务逻辑还有表现页面分割开来。它为减少开发复杂 web 应用提供了大量的工具和类。另外，它把一些常用的任务变成了自动化的方式从而使开发人员能够完全专注于每个程序的独特的地方。这三个优点综合起来就意味着每次开发新的 web 应用的时候不用重新发明轮子了！

symfony 完全由 PHP 5 编写。它经过了广泛的测试并被用于很多实际的项目中，一些高访问量的电子商务网站正在使用 symfony。symfony 与包括 MySQL、PostgreSQL、Oracle 还有 Microsoft SQL Server 在内的大多数数据库系统兼容。symfony 能够在 *nix 与 Windows 平台上运行。我们现在来看一下 symfony 具体有哪些功能。

symfony 的功能

symfony 是为了满足下面的需求而开发的：

- 在大多数平台上都能够很容易安装（保证能运行在标准 *nix 和 Windows 平台）
- 不依赖某种特定的数据库
- 多数情况下容易使用，并且有足够的弹性来实现更复杂的功能
- 遵循惯例重于配置原则——开发人员只需要配置与惯例不同的部分
- 能够适应大多数 web 最佳实践与设计模式
- 可供企业使用——能够适应现有的 IT (information technology) 策略与体系，并且很稳定适合长期项目
- 非常易读的代码、带有 phpDocumentor 注释，维护轻松
- 易于扩展，允许与非 symfony 的库进行整合

自动化 Web 项目功能

symfony 能够自动化 web 开发的大部分要素：

- 内建的国际化层能够实现数据与界面翻译，还有内容本地化。
- 表现部分由于使用了模板和布局，不懂任何框架知识的网页设计师也可以掌握。辅助函数封装了大量的代码从而减少了表现部分的代码。
- 表单支持自动验证和重新提交，这确保了数据库里的数据的质量，用户体验也更好。
- 输出转义能够保护程序不受到利用错误数据的攻击。
- 缓存管理功能减少了网络带宽和服务器负载消耗。
- 使用验证与证书功能能很容易地实现受限制区域还有用户权限管理。
- 路由与漂亮的 URL 使网页的网址容易被搜索引擎接受。
- 内建的 e-mail 与 API 管理使 web 程序超越传统的浏览器交互。
- 方便的数据列表提供了自动的分页、排序还有筛选功能。
- Factories, plug-ins, 还有 mixins 提供了高层次的扩展性。
- 容易使用的 Ajax 交互，以一行代码(辅助函数 helper)去封装跨浏览器支持的 Javascript

开发环境与工具

symfony 可以完全的定制以满足有自己代码规范与项目管理规则的企业。 它自带了一些开发环境还有多种工具来实现自动化软件工程任务：

- 代码生成工具非常适合于原形设计与一键式后台管理。
- 内建的单元测试与功能测试框架为测试驱动的开发提供了完美的工具。
- 测试工具条能够把当前页面的所有开发者需要的信息显示出来从而加快调试速度。
- 能够实现两台服务器之间自动化部署的命令行工具。
- 能够有效地即时更新配置。
- 日志功能让管理员能够掌握程序的一举一动。

谁开发了 symfony？为什么要开发 symfony？

symfony 的第一个版本由项目发起者同时也是本书的合著者 Fabien Potencier 于 2005 年 10 月发布。Fabien 是 Sensio (<http://www.sensio.com/>) 的 CEO，Sensio 是一个法国对 web 开发有创新见解的知名网络机构。

早在 2003 年，Fabien 研究了当时的 PHP 开源开发工具，发现没有一个能够满足上面的这些需求。PHP 5 发布的时候，他发现现有的工具已经都很成熟，很难整合到一个全功能的框架里。随后他花了 1 年的时间开发了 symfony 的核心，这个核心基于 Mojavi MVC (Model-View-Controller) 框架，Propel ORM (object-relational mapping)，还有 Ruby on Rails 的模板辅助方法。

Fabien 最初是为了 Sensio 的项目开发了 symfony，因为一个高效的开发框架能大大加快开发效率。它使得开发更直观，开发的程序更健壮更容易维护。这个框

架最初在一个女性内衣的零售商的电子商务网站的项目中应用并取得了成功，随后被用到了其他的项目中。

在成功地将 symfony 运用于几个项目之后，Fabien 决定用开源的协议来发布 symfony。他之所以这么做是为了把自己的工作成果捐献给社区，并且收集用户的意见来改进框架，展示 Sensio 的经验，另外这样做本身也是一种乐趣。

NOTE 为什么是“symfony”而不是“FooBarFramework”呢？因为 Fabien 想要一个简单的名字，包含代表 Sensio 的“s”还有代表 framework 的“f”，并且容易记忆，不跟其他的开发工具名字相似。另外，他不喜欢大写字母。尽管不是标准的英语，“symfony”成为了 Fabien 的选择，并且 symfony 也是这个项目的名称。另外一个项目名称是“baguette”。

要成为一个成功的开源项目，symfony 需要大量的英文文档，这样能使更多的开发者使用 symfony。Fabien 请 Sensio 的职员 François Zaninotto (本书的作者) 深入源代码来写电子版的手册。写作花了不少时间，但是当项目公开的时候，它优秀的文档吸引了不少开发者。其余的就什么也不用说了。

symfony 社区

当 symfony 的网站 (<http://www.symfony-project.com/>) 推出后，全世界许许多多的开发者下载、安装、阅读了在线文档，随后去开发了他们的第一个 symfony 的程序，社区开始热闹起来。

当时 web 应用程序开发框架正在变得流行，开发者们对全功能的 PHP 开发框架的需求很高。symfony 由于代码质量还有文档的数量成为一个引人注目的解决方案，这也是它相对于其他框架的优势。很快就有志愿者参与进来，提出修改或增强的意见，校对文档，并参与到其他一些必要的工作中来。

我们欢迎所有愿意参与的人加入 symfony 开发，我们提供公开的源代码仓库还有 ticket 系统。目前 symfony 的代码主干主要还是 Fabien 在维护，这保证了代码的质量。

目前，symfony 的论坛，邮件列表，还有 IRC 频道为整个社区提供了很优秀的支持，平均每个问题会有 4 个回复之多。每天都有人安装 symfony，wiki 与代码片段部分汇集了大量的用户提交的文档。每个星期平均会出现 5 个新的 symfony 项目，这个数字还在不断地增加。

symfony 社区是这个框架的第三个力量，我们希望读了本书后你能加入到这个社区中来。

symfony 适合我吗？

不论你是 PHP 5 专家或 Web 应用程序开发的新手，你都能用 symfony。决定是否使用 symfony 的主要因素是项目的大小。

如果你想开发一个只有 5 到 10 个网页的简单网站，少量的数据库访问，并且不需要保证速度或者提供文档，那么你只要用 PHP 就够了。这种情况下使用 web 开发框架并没有太大的价值，而且面向对象或者 MVC 模型会使开发速度变慢。另外，symfony 在只能以 CGI 模式下运行 PHP 的虚拟主机环境下并不能很有效率的运作。

另一方面，如果你要开发一个更复杂的有更多业务逻辑的 web 应用程序，那么单独使用 PHP 是不够的。如果你打算以后维护或扩展你的应用程序，那么你的代码需要是简洁，易读和高效的。如果你打算在用户界面中方便的使用最新的技术（例如 Ajax），你不能只是写几百行 Javascript 代码。如果你想愉快而且快速的开发，那么单独使用 PHP 可能会令你失望。上面这些，就是 symfony 适合你的原因。

当然，如果你是一个专业的 web 开发人员，你已经知道了 web 应用框架的所有优点，并且你需要一个成熟的，具有详细文档和一个大的社区支持的 web 应用框架。那么不要再犹豫，symfony 就是你的解决方案。

TIP 如果你想看比较直观的演示，建议去 symfony 的网站上的 screencast。看了以后你会发现使用 symfony 开发是多么快速而愉快。

基本概念

在开始学习 symfony 之前，有几个概念需要理解。如果你已经知道 OOP、ORM、RAD、DRY、KISS、TDD、YAML 还有 PEAR 你可以跳过本节。

PHP 5

symfony 本身是由 PHP 5 (<http://www.php.net/>) 开发的，symfony 专注于利用 PHP 5 制作网络应用程序。所以，想要彻底地了解 symfony 框架必须要对 PHP 5 有十分深刻的理解。

建议熟悉 PHP 4 但刚刚接触或者不熟悉 PHP 5 的开发者专注于 PHP 5 语言的新面向对象模型部分。

面向对象程序设计 (OOP)

我们在本章不去详细解释面向对象程序设计 (OOP)，因为这个话题足够写一本书了。由于 symfony 大量运用了 PHP 5 中的面向对象机制，面向对象程序设计 (OOP) 是学习 symfony 的先决条件。

维基百科这样解释 OOP:

面向对象程序设计 (OOP) 可以被视作一种在程序中包含各种独立而又互相调用的单位和对象的思想, 这与传统的思想刚好相反: 传统的面向过程程序设计主张将程序看作一系列函数的集合, 或者直接就是一系列对电脑下达的指令。

PHP 5 实现了面向对象中的类、对象、方法、继承等。如果你对这些概念不熟悉, 建议阅读相关的 PHP 文档, 网址如下

<http://www.php.net/manual/zh/language.oop5.basic.php>。

魔术方法 (Magic Methods)

PHP 对象的一个优势是可以使用魔术方法。 这些方法可以不需要修改外部代码而重写一个类的默认行为。这使得 PHP 语法有更少的冗余性和更具有扩展性。这些方法很好识别, 他们都是以双下划线(__)开始的。

例如, 当显示一个对象的时候, PHP 会暗中去寻找是否开发者定义过 `__toString()` 方法:

```
$myObject = new myClass();  
echo $myObject;  
// Will look for a magic method echo  
$myObject->__toString();
```

symfony 使用了魔术方法, 所以你必须完全了解这些概念。 这些在 PHP 文档中有描述 (<http://www.php.net/manual/zh/language.oop5.magic.php>)。

PHP 扩展与应用程序库 (PEAR)

PEAR 是“一个 PHP 可重用代码的框架和发布系统”。PEAR 可以下载、安装、升级及删除 PHP 脚本。使用 PEAR 包的时候, 不用为了脚本的位置或者怎么找到他们而担心, 扩展命令行接口 (CLI) 也很容易使用。

PEAR 是一个由社区推动的 PHP 项目, 官方发布的 PHP 中就包含了 PEAR。

TIP PEAR 网站, <http://pear.php.net/>, 有 PEAR 文档与分类的 PEAR 包下载。

PEAR 是最专业的安装 PHP 库的方法。建议使用 PEAR 来管理一个由多个项目共用的 symfony 框架。symfony 的插件 (plug-ins) 是一种有特殊设置的 PEAR 包。symfony 本身也可以通过 PEAR 安装。

使用 symfony 并不需要懂 PEAR 命令的语法。你只要知道 PEAR 的用途还有确定它已经安装好就行了。你可以通过在你的电脑的命令行 (CLI) 下输入下面的命令来检查 PEAR 是否安装：

```
> pear info pear
```

这个命令会返回安装在你的电脑上的 PEAR 的版本。

symfony 项目有自己的 PEAR 仓库 (或频道)。仓库功能只有 PEAR 1.4.0 以上版本才支持，所以如果你的版本比较老就需要升级。升级 PEAR，只要在命令行输入下面的命令就可以了：

```
> pear upgrade PEAR
```

对象关系映射 (ORM)

数据库是关系型数据库。PHP5 和 symfony 是面向对象的。为了用面向对象的方法访问数据库，必须用一个接口来表示对象之间的逻辑关系。这个接口就叫作对象关系映射 (object-relational mapping) 或者 ORM。

ORM 是由能够访问数据和存放业务规则的对象组成。

对象/关系抽象层的其中一个优点是可以不用直接去访问数据库。它会使用经过优化的模型对象来访问当前的数据库。

这就意味着在项目中后期换一套数据库将是很简单的事情。想象一下当你必须为程序写一个原型的时候，客户并不能确定哪种数据库最适合他们。你可以先使用 SQLite 来开发程序，当客户决定使用 MySQL, PostgreSQL 或者 Oracle 的时候，我们只要在配置文件中稍作修改就可以正常工作了。

抽象层封装了数据逻辑。其他程序并不需要了解 SQL 的查询语句，却依旧能轻松的访问数据库。那些数据库开发专家也很清楚的知道该做些什么。

使用对象而非记录，用类而非表，还有其他益处：你能为你的表增加一些新的存取方法。例如，你有一个名叫 Client 的表，有两个字段，分别是 FirstName 和 LastName，你也许想直接获得一个完整的姓名。在面向对象中，为一个 Client 类添加一个访问方法，就像这样：

```
public function getName()
{
    return $this->getFirstName().' '.$this->getLastName();
}
```

所有重复数据访问功能和数据的业务逻辑都可以在对象中维护。例如，有一个用来生成对象的 ShoppingCart 类。我们在结帐时想获得一个总价，你可以加一个 getTotal() 方法，就像这样：

```
public function getTotal()
{
    $total = 0;
    foreach ($this->getItems() as $item)
    {
        $total += $item->getPrice() * $item->getQuantity();
    }
    return $total;
}
```

就这好了，想象一下要写多长的 SQL 语句才能完成同样的事情！

Propel，另一个开源项目，是当前最好的基于 PHP5 的对象/关系抽象层。symfony 框架无缝集成了 Propel，所以本书大多数的数据处理描述都使用了 Propel 的语法。本书将描述如何使用 Propel 对象，但是更详细的资料可以参考 Propel 的网站 (<http://propel.phpdb.org/trac/>)。

快速应用程序开发 (RAD)

开发网页程序是一件单调乏味的，慢速的事情。按照常规软件开发生存周期(类似于 Rational 统一过程里设想的)，要等到有完整的需求，绘制大量的 UML 图，产生了大量的正式文档之前准备阶段的文档才能开始开发。这是由于一般的开发速度，缺少通用性的程序语言（在能正式看到程序可用之前不知道要多少次的建立，编译，重运行），最主要的就是客户不会经常改变他们的主意。

今天，商业节奏更快了，客户总是倾向于在制作项目中经常改变他们的需求。当然，他们期望开发组能接受他们的需求并且快速更改应用程序的架构。幸运的是，使用脚本语言就像 Perl 和 PHP 会更容易的去实现这些，例如快速应用程序开发 (RAD) 和敏捷程序开发。

尽快去建立一个工作原型以便客户可以审阅并且提出问题是一个好方法。如此周而复始，在较短的开发周期发布新的功能。

有很多东西需要开发者考虑。开发者不需要去想在未来如何实现一个新功能。用最简单的方法去实现需要的功能。这就是 KISS 原则的一个很好体现：Keep It Simple, Stupid.

当需求变化或者功能增加的时候，有时需要重写代码。这就叫做重构，这经常发生在网页应用程序开发中。代码会根据需要改变存放位置。把重复的地方合并到一个地方，这就是 Don't Repeat Yourself (DRY) 原则。

当程序改变后要确保依旧能够运行，这需要一系列的自动测试来完成。如果写得好的话，单元测试将是在重构代码后检查的一个好的方法。一些开发方法学（development methodologies）甚至于规定在编写代码前先写测试——这就使称之为测试驱动开发(TDD)。

NOTE 关于敏捷开发还有其它一些原则和好习惯。其中一个最有效率的方法叫做极限编程（Extreme Programming）（简称 XP），极限编程的教材会教你如何快速而有效的去开发一个程序。推荐从 Kent Beck (Addison-Wesley) 的极限编程系列开始学习。

symfony 非常适合进行 RAD。事实上，制作这个框架的网络公司就使用 RAD 原则进行他们的项目。这意味着学习使用 symfony 不是去学习新语言，而是去学习如何正确的反应和判断从而更有效率的开发应用程序。

symfony 项目的网站有一个步骤详细的教程，完整的介绍了如何快速进行开发。它叫做 askeet (<http://www.symfony-project.com/asket>)，推荐想了解敏捷开发的朋友阅读这个教程。

YAML

来自 YAML 官方网站 (<http://www.yaml.org/>) 的定义：YAML 是一种直观的能够被电脑识别的数据序列化格式，并且它容易被人类阅读，容易与脚本语言交互的。换种说法，YAML 是一种非常简单的类似于 XML 的数据描述语言，语法比 XML 简单很多。他在描述可以被转化成数组或者 hash 的数据是非常有用，例如：

```
$house = array(
    'family' => array(
        'name' => 'Doe',
        'parents' => array('John', 'Jane'),
        'children' => array('Paul', 'Mark', 'Simone')
    ),
    'address' => array(
        'number' => 34,
        'street' => 'Main Street',
        'city' => 'Nowheretown',
        'zipcode' => '12345'
    )
);
```

解析这个 YAML 将会自动创建下面的 PHP 数组：


```
house:
  family:
    name: Doe
    parents:
      - John
      - Jane
    children:
      - Paul
      - Mark
      - Simone
  address:
    number: 34
    street: Main Street
    city: Nowheretown
    zipcode: 12345
```

在 YAML 里面，结构通过缩进来表示，连续的项目通过减号“-”来表示，map 结构里面的键/值（key/value）对用冒号“:”来分隔。YAML 也有用来描述好几行相同结构的数据的缩写语法，数组用‘[]’包括起来，hash 用‘{}’来包括。因此，前面的这个 YAML 可以缩写成这样：

```
house:
  family: { name: Doe, parents: [John, Jane], children: [Paul, Mark, Simone] }
  address: { number: 34, street: Main Street, city: Nowheretown, zipcode: 12345 }
```

YAML 是“Yet Another Markup Language(另一种标记语言)”的缩写，读音“yamel”，或者“雅梅尔”。这种格式大约是 2001 年出现的，目前为止已经有多种语言的 YAML 解析器。

TIP YAML 格式的详细规格可以在 YAML 官方网站 <http://www.yaml.org/> 找到。

如你所见，写 YAML 要比 XML 快得多(不需要关闭标签或者引号)，并且比‘.ini’文件功能更强(ini 文件不支持层次)。所以 symfony 选择 YAML 作为配置信息的首选格式。在本书你会看到很多 YAML 文件，不过它很直观你用不着更深入地研究 YAML。

总结

symfony 是一个 PHP 5 web 应用程序开发框架。他在 PHP 语言的基础上增加了一个新层，为加速开发复杂的 web 应用程序提供了工具。本书全面介绍 symfony 的使用，你只需要熟悉并且理解现代编程的基本概念——面向对象程序设计(OOP)、对象关系映射(ORM)还有快速程序开发(RAD)。唯一需要的技术背景是 PHP 5 的知识。

