

<<symfony权威指南>>

图书基本信息

书名：<<symfony权威指南>>

13位ISBN编号：9787121067495

10位ISBN编号：7121067498

出版时间：2008-7

出版时间：电子工业出版社

作者：(法)扎尼诺托,(法)波唐西耶著

页数：358

字数：471

译者：江乐,王唯真,杭健,王天石译

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<symfony权威指南>>

内容概要

本书介绍了symfony这个领先的PHP开发框架，展示如何利用symfony的众多功能更快更有效率地开发Web应用程序，即使你只懂一点PHP也没有关系。

在本书里，你将了解如何使用symfony建立Web应用程序。

本书分成五篇：“基础知识”篇，包含所有的基本概念和开始symfony的基本知识；“核心架构”篇，讲述模型视图控制器（MVC）在symfony中的实现，以及如何用这样的三层结构模型建立页面和应用程序；“特殊功能”篇，解释如何使用symfony加快“漂亮”的URL、表单、验证、Ajax交互、缓存和国际化应用程序的开发；“开发工具”篇，介绍symfony提供的加快日常开发的工具，代码生成、单元测试框架、命令行任务和插件；“成为symfony专家”篇，揭示symfony代码里的一些秘密，讲述了如何调整框架的行为。

本书的目的在于降低框架入门的难度，为PHP开发者提供1.0版的参考书，让这个框架能被企业接受。

<<symfony权威指南>>

作者简介

Francois Zaninotto是一名互联网应用项目顾问和项目经理。1997年毕业于法国商业学校Ecole des Mines计算机专业。在选择从事互联网行业之前他尝试了多份工作：儿童机构的社工、自行车出租店经理、轮胎制造商的Web项目经理、为这个轮胎制造商撰写到德国旅游的导游文章，以及M é decins Sans Fronti è res的物流师和某消费信用公司的IT架构师。他于2003年加入Sensio网络公司，此后负责了许多互联网和Intranet的Web应用项目，解决了很多复杂的可用性问题（ Usability ），涉及许多应用敏捷开发模式模型和先进的Web技术.symfony项目开始之后，他负责编写文档.symfony在线书籍和教程。

Fabien Potencier 是天生的企业家.从10岁开始，他就总是梦想着成立和运营公司.他在法国商业学校Ecole des Mines取得工程学学位和在巴黎高等商业学院（ HECParis ）取得企业MBA学位之后开始职业生涯.1998年，他刚毕业的那年，Fabien和他的同学成立了他的第一家公司.这是一家专注于简化技术和开源技术的网络公司，它的名字是Sensio.他靠自己敏锐的专业知识和无止境的求知欲获得了很多法国大公司的项目.随着Sensio的不断发展（写这本书的时候，它有30多名员工），Fabien开始从事其他业务：一个在Lille（法国）的室内卡丁车赛道，一个汽车配件电子商务网站，还有一个在法国最著名的赛道的自动驾驶训练生意.Fabien是symfony框架的主要开发者，他负责所有代码的95%.现在，Fabien把大部分时间花在Sensio的CEO和symfony项目负责人的工作上。

<<symfony权威指南>>

书籍目录

第1篇 基础知识	第1章 symfony介绍	1.1 symfony简介	1.1.1 symfony的功能	1.1.2 谁开发了symfony?
	为什么要开发symfony	1.1.3 symfony社区	1.1.4 symfony适合我吗	1.2 基本概念
	1.2.1 PHP 5	1.2.2 面向对象程序设计 (OOP)	1.2.3 魔术方法 (Magic Methods)	
	1.2.4 PHP扩展与应用程序库 (PEAR)	1.2.5 对象关系映射 (ORM)	1.2.6 快速应用程序开发 (RAD)	
	1.2.7 YAML	1.3 总结	第2章 探索symfony代码	2.1 MVC模式
	2.1.1 MVC层次	2.1.2 MVC以外的层分离方式	2.1.3 symfony的MVC实现方式	2.1.4 symfony核心类
	2.2 代码组织	2.2.1 项目结构：应用程序、模块与动作	2.2.2 目录结构	
	2.3 常用工具	2.3.1 参数存储器 (Parameter Holoder)	2.3.2 常量 (Contant)	
	2.3.3 类自动载入 (Class Autoloading)	2.4 总结	第3章 运行symfony	3.1 安装沙盒 (Sandbox)
	3.2 安装symfony库	3.2.1 PEAR方式安装symfony	3.2.2 从SVN库中获得symfony	3.4 配置Web服务器
	3.3 配置一个程序	3.3.1 建立一个项目	3.3.2 建立一个应用程序	3.5 安装问题
	3.4.1 设置虚拟主机	3.4.2 配置一个共享服务器	3.5 安装问题	3.5.2 symfony资源
	3.5.2 symfony资源	3.6 源代码版本控制	3.7 总结	第4章 页面创建基础
	4.2 增加一个页面	4.2.1 增加一个动作	4.2.2 增加一个模板	4.2.3 动作传递信息给模板
	4.3 通过表单从用户获取数据	4.4 链接到另一个动作	4.5 从请求中取得信息	4.6 总结
	第5章 配置symfony	5.1 配置系统	5.1.1 YAML语法与symfony惯例	5.1.2 救命，YAML文件把我的程序搞死了
	5.2 配置文件概述	5.2.1 项目配置	5.2.2 应用程序配置	5.2.3 模块配置
	5.3 环境	5.3.1 什么是环境	5.3.2 配置层叠	5.5 从代码里访问配置信息
	5.5.1 sfConfig类	5.5.2 自定义应用程序配置	5.6 使用配置文件的技巧	5.6.1 在YAML文件里使用常量
	5.6.2 在配置文件里使用脚本	5.6.3 浏览你的YAML文件	5.7 总结	第2篇 核心架构
	6.1 前端控制器	6.1.1 前端控制器的工作细节	6.1.2 默认的前端控制器	6.1.3 调用其他的前端控制器来切换环境
	6.2.2 另一种动作类语法	6.2.3 从动作里获取信息	6.2.4 动作结束	6.2.5 跳到另一个动作
	6.2.6 几个动作共享的代码	6.3 访问请求	6.4 用户会话	6.4.1 访问用户会话
	6.4.2 Flash属性	6.4.3 会话管理	6.5 动作安全	6.5.1 访问限制
	6.5.3 复合证书	6.6 检验和处理错误的方法	6.7 过滤器	6.7.1 过滤器链
	6.7.3 过滤器激活和参数	6.7.4 过滤器实例	6.8 模块配置	6.9 总结
	第7章 深入了解视图层	7.1 模板	7.1.1 辅助函数 (Helpers)	7.1.2 页面布局
	7.1.3 模板快捷变量	7.2 代码片段 (Code Fragments)	7.2.1 局部模板 (Partial)	7.2.2 组件 (Component)
	7.2.3 槽 (Slot)	7.3 视图配置	7.3.1 view.yml文件	7.3.2 对象
	7.3.3 视图配置	7.4 组件槽 (Component Slot)	7.5 输出转义 (Output Escaping)	7.5.1 开启输出转义
	7.5.2 转义策略	7.5.3 转义辅助函数	7.5.4 转义数组与对象	7.6 总结
	第8章 深入了解模型层	8.1 为什么使用ORM和抽象层	8.2 symfony的数据库	8.2.1 设计 (schema)
	8.2.2 基本设计语法	8.3 模型类	8.3.1 类和自定义类	8.3.2 对象和Peer类
	8.4 访问数据	8.4.1 获得列值	8.4.2 获得相关的数据	8.4.3 保存和删除数据
	8.4.4 通过主键来获得记录	8.4.5 通过Criteria获得数据	8.4.6 直接使用SQL查询语句	8.4.7 使用特殊日期列
	8.5 数据库连接	8.6 扩展模型	8.6.1 增加新的方法	8.6.2 覆盖现有方法
	8.6.3 使用模型行为	8.7 扩展设计 (schema) 语法	8.7.1 属性	8.7.2 列的详细资料
	8.7.3 外键	8.7.4 索引	8.7.5 空列	8.7.6 i18n表
	8.7.7 超越schema.yml	8.7.7 schema.xml	8.8 不要重复建立模型	8.8.1 基于已有的设计建立
	6.1.3 一个SQL数据库结构	8.8.2 基于已有数据库建立	6.1.3 YAML数据模型	8.9 总结
	第3篇 特殊功能	第9章 链接和路由系统	9.1 什么是路由	9.1.1 URL是服务器指令
	9.1.2 URL是界面的一部分	9.1.3 URL如何工作		

<<symfony权威指南>>

9.2 URL重写	9.3 链接辅助函数	9.3.1 超链接、按钮和表单	9.3.2 链接辅助函数的可
选参数	9.3.3 伪装的GET和POST选项	9.3.4 把请求参数强制设置为	6.1.3 GET变量
9.3.5 使用绝对路径	9.4 路由配置	9.4.1 规则和模式	9.4.2 模式限制
置默认值	9.4.4 用规则名字来加快	6.1.3 路由速度	9.4.5 增加.html扩展名
不使用routing.yml创建规则	9.5 在动作中处理路由	9.6 总结	第10章 表单
数	10.1.1 主要的表单标签	10.1.2 标准的表单元素	10.1.3 日期输入控件
编辑富文本 (rich text)	10.1.5 选择国家和语言	10.2 对象的表单辅助函数	10.2.1 生成
对象的下拉列表	10.2.2 创建一个基于外键列的下拉列表	10.2.3 更新对象	10.3 表单验
证	10.3.1 验证器	10.3.2 验证文件	10.3.3 重新显示表单
错信息	10.3.5 重新填充表单数据	10.3.6 标准symfony验证器	10.3.7 具名验证器
(Named Validator)	10.3.8 重新指定验证方法	10.3.9 验证文件到底是什么样子	10.4 复
杂的验证	10.4.1 创建一个定制的验证器	10.4.2 用数组表示表单域	10.4.3 验证空域
10.5 总结	第11章 集成Ajax	11.1 基本的JavaScript辅助函数	11.1.1 模板中的JavaScript
11.1.2 更新一个DOM元素	11.1.3 轻松地降级Graceful Degradation)	11.2 Prototype	11.3
Ajax辅助函数	11.3.1 Ajax链接	11.3.2 Ajax驱动的表单	11.3.3 周期性调用远程函数
11.4 远程调用参数	11.4.1 根据响应状态更新不同的元素	11.4.2 根据位置更新元素	
11.4.3 根据条件更新元素	11.4.4 确定Ajax请求方法	11.4.5 授权脚本运行	11.4.6 创
建回调函数	11.5 创建视觉效果	11.6 JSON	11.7 用Ajax完成复杂的交互
完成	11.7.2 拖放	11.7.3 可排列表	11.7.4 就地编辑
12.1 缓存响应	12.1.1 全局缓存设置	12.1.2 缓存一个动作	12.1.3 缓存一个局部模板
、组件或者组件槽	12.1.4 缓存模板片段	12.1.5 动态配置缓存	12.1.6 使用极速缓存
(Super Fast Cache)	12.2 从缓存里删除项目	12.2.1 删除整个缓存	12.2.2 清除指定的缓
存	12.2.3 缓存目录结构	12.2.4 手工清除缓存	12.3 缓存测试与监测
个临时工作环境	12.3.2 监测性能	12.3.3 基准化分析 (Benchmarking)	12.3.4 识别缓
存的部分	12.4 HTTP	1.1与客户端缓存	12.4.1 增加ETag头信息来避免发送重复的内容
12.4.2 增加Last-Modified头信息避免发送仍然有效的内容	12.4.3 通过增加Vary头信息来保		
存一个页面的多个缓存版本	12.4.4 通过增加Cache-Control头信息来允许客户端缓存	12.5 总	
结	第13章 国际化 (i18n) 与本地化 (l10n)	13.1 用户的国家和	13.1 语言 (User Culture)
13.1.1 设置默认的国家 and 语言	13.1.2 改变用户的国家和语言	13.1.3 自动确定用户的	
国家和语言	13.2 标准与格式	13.2.1 根据用户culture值输出数据	13.2.2 从本地化输入获
取数据	13.3 数据库中的文本信息	13.3.1 创建本地化数据库设计 (schema)	13.3.2 运
用生成的i18n对象	13.4 界面翻译	13.4.1 翻译的配置	13.4.2 运用翻译辅助函数
13.4.3 运用字典文件	13.4.4 管理字典	13.4.5 处理其他需要翻译的元素	13.4.6 处理
复杂的翻译需求	13.4.7 在模板外调用翻译辅助函数	13.5 总结	第4篇 开发工具
器 (Generator)	14.1 基于模型生成代码	14.1.1 脚手架与管理界面	14.1.2 初始化或生
成代码	14.1.3 数据模型示例	14.2 脚手架	14.2.1 生成脚手架
14.3 管理界面	14.3.1 初始化管理界面模块	14.3.2 初探生成的代码	14.3.3
generator.yml配置文件	14.4 配置生成器	14.4.1 字段	14.4.2 视图定制
图相关的定制	14.4.4 edit视图相关的定制	14.4.5 处理外键	14.4.6 增加交互
14.4.7 表单验证	14.4.8 使用证书限制用户动作	14.5 修改生成模块的外观	14.5.1 使用
自定义样式表	14.5.2 增加自定义头部与尾部	14.5.3 自定义主题	14.6 总结
元测试和功能测试	15.1 自动化测试	15.1.1 单元测试和功能测试	15.1.2 测试驱动的开
发方法	15.1.3 lime测试框架	15.2 单元测试	15.2.1 单元测试概述
方法	15.2.3 测试参数	15.2.4 测试单元任务	15.2.5 测试桩 (Stub)、测试资源
(Fixture) 和自动加载	15.3 功能测试	15.3.1 功能测试概要	15.3.2 用sfTestBrowser对象
浏览	15.3.3 运用断言	15.3.4 运用CSS选择器	15.3.5 在测试环境中工作
功能测试任务	15.4 为测试命名	15.5 特别的测试需求	15.5.1 在测试框架 (Test Harness)

<<symfony权威指南>>

15.5.1 中进行测试	15.5.2 存取数据库	15.5.3 测试缓存	15.5.4 测试客户端交互
15.6 总结	第16章 应用程序管理工具	16.1 日志	16.1.1 PHP日志
16.2 调试	16.2.1 symfony调试模式	16.2.2 symfony异常	16.2.3 Xdebug扩展
16.2.4 网页调试工具条	16.2.5 手动调试	16.3 填充数据库	16.3.1 fixture文件格式
16.3.2 导入数据	16.3.3 使用关联表	16.4 部署应用程序	16.4.1 为FTP传输冻结项目
16.4.2 使用rsync进行增量	16.4.2 文件传输	16.4.3 忽略无关文件	16.4.4 管理投入
使用的应用程序	16.5 总结	第17章 扩展symfony	17.1 Mixins
17.1.2 Mixing类	17.1.3 声明一个类是可扩展的	17.1.4 注册扩展 (Extension)	17.1.5 理解多重继承
更精确的扩展	17.2 factory	17.3 桥接其他框架组件	17.4 插件
17.4.2 安装插件	17.4.3 解读插件	17.4.4 如何写一个插件	17.4.1 查找symfony插
为symfony专家	第18章 性能	18.1 调整服务器	18.2 调整模型
18.2.2 限制化合 (Hydrate)	18.2.2 对象数量	18.2.3 用Join让结果数量最小化	18.2.4 优化Propel整合
避免使用临时数组	18.2.5 绕过ORM	18.2.6 数据库加速	18.3 调整视图
用最快的代码片段	18.3.2 加速路由过程	18.3.3 略过模板	18.3.4 限制默认的辅助函数
18.3.5 压缩响应	18.4 调整缓存	18.4.1 选择性地清除部分缓存	18.4.2 生成缓存页
18.4.3 使用数据库存储系统	18.4.3 进行缓存	18.4.4 绕过symfony	18.4.5 缓存函
数调用的结果	18.4.6 在服务器上缓存数据	18.5 屏蔽未使用过的功能	18.6 优化你的代码
18.6.1 核心编译	18.6.2 sfOptimizer插件	18.7 总结	第19章 管理symfony配置文件
19.1 symfony配置参数	19.1.1 默认模块和动作	19.1.2 激活可选特性	19.1.3 功能
特性配置	19.2 扩展自动载入功能	19.3 定制文件结构	19.3.1 基本的文件结构
定制文件结构	19.3.3 修改项目的Web根目录	19.3.4 连接symfony库	19.4 理解配置处理
器	19.4.1 默认的配置处理器	19.4.2 加入你自己的处理器	19.5 控制PHP参数
总结			19.6

<<symfony权威指南>>

章节摘录

第1章 symfony介绍 symfony能做什么？

使用symfony需要掌握哪些知识？

本章将回答这些问题。

1.1 symfony简介 开发框架（framework）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化，使用开发框架，开发人员可以写出更好、更易读、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Web应用开发提供了几个关键功能。

首先，它把Web应用的业务规则，服务逻辑以及表现页面分割开来。

它还提供了大量的工具和类，以缩短开发复杂Web应用的时间。

另外，它把一些常用的任务变成了自动化的方式执行，从而使开发人员能够完全专注于每个程序独特的地方。

这三个优点综合起来就意味着每次开发新的Web应用的时候不用从头做了！

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，事实上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。

1.1.1 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和配置（保证能运行在标准*nix和Windows平台上）。

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策略与体系，并且足够稳定适合长期项目。

非常易读的代码、带有phpDocumentor注释，维护起来很轻松。

易于扩展，允许与非symfony的库进行整合。

自动化Web项目功能 Web项目的大部分功能在symfony中都能自动实现，如下： 内建的国际化层能够实现数据与界面翻译，以及内容本地化。

表现（presentation）部分使用了模板和布局，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的负载。

使用验证与证书功能能很方便地实现受限制区域和用户权限的管理。

路由与智能的URL使网页的网址容易被搜索引擎索引。

内建的E-mail与API管理使Web应用比传统的浏览器交互更灵活。

数据列表提供了自动的分页、排序和筛选功能，用起来十分方便。

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）就能封装跨浏览器支持的JavaScript。

开发环境与工具 symfony可以完全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，非常适合于原型设计与一键式后台管理。

<<symfony权威指南>>

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理想的工具。

调试工具条，能够把当前页面开发者需要的所有信息显示出来，从而加快调试速度。

命令行工具，能够实现两台服务器之间自动化部署。

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。

1.1.2 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是一个法国的网络机构，以其对Web开发的创新见解而知名。

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前面所述这些需求。PHP 5发布的时候，他发现现有的工具已经都很成熟，足以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的项目而开发symfony，因为一个高效的开发框架能大大加快开发效率。它使得Web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出售女性内衣的零售商的电子商务网站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决定用开源的许可证 (license) 来发布symfony。他这么做了，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，因为这样做本身也是一种乐趣。

注意 为什么是“symfony”而不是“FooBarFramework”呢？

因为Fabien想用一个简单的名字，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，Fabien选择了“symfony”，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，这样才能使更多的开发者使用symfony。

Fabien请Sensio的职员Fran ois Zaninotto (本书的另一位作者) 仔细研读源代码以撰写电子版的手册。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。

接下来的事情已经人尽皆知，就什么也不用说了。

1.1.3 symfony社区 当symfony的网站 (<http://www.symfony-project.com/>) 推出后，全世界许多的开发者下载、安装、阅读了在线文档，并开发了他们自己的第一个用symfony实现的程序，社区开始热闹起来。

当时Web应用程序开发框架正开始流行，开发者们十分需要全功能的PHP开发框架。symfony由于良好的代码质量和大量详实的文档而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，提出修改或改善的意见，校对文档，并参与其他一些必要的工作。

社区欢迎所有愿意参与的人加入symfony开发，并提供公开的源代码仓库和ticket系统。

目前symfony的代码主干主要还是由Fabien维护，这保证了代码的质量。

现在，symfony的论坛、邮件列表，以及IRC频道为整个社区提供了令人满意的支持，平均每个问题会有4个回复之多。

每天都有人安装symfony。

Wiki和snippets部分也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，而且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权威指南>>

1.1.4 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的主要因素是你的项目的规模。

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模式运行PHP的虚拟主机环境下并不能很有效率地运作。

另一方面，如果你要开发一个复杂一点的，有更多业务逻辑的Web应用程序，那么仅仅使用PHP是不够的。

如果你打算以后维护或扩展你的应用程序，那么你的代码应该是简洁、易读和高效的。

如果你打算在用户界面中方便地使用最新的技术（例如Ajax），你不可能只是写几百行JavaScript代码

。如果你想愉快而且快速地开发，那么仅使用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，已经知道了Web应用框架的所有优点，并且需要一个成熟的、具有详细文档和一个大社区支持的Web应用框架。

那么不要再犹豫了，symfony就是你的解决方案。

<<symfony权威指南>>

编辑推荐

开发新Web应用程序并不需要重新发明轮子，所以说框架应该是你开发工具箱中的重要工具。它帮助你遵循代码标准，写出健壮的、可维护的代码，专注于业务规则而不是在重复的任务上浪费时间。

本书将为你介绍symfony这个领先的PHP开发框架，向你展示如何利用symfony的众多功能来更快更有效率地开发Web应用程序，即使你只懂一点PHP也没有关系。

在本书中，你会了解模型—视图—控制器架构以及它在symfony这类框架中的重要作用。我们还会介绍框架的安装和配置，教你如何建立页面、处理模板管理请求和用户会话，以及与数据库的沟通。

通过symfony的表单数据管理来简化开发，利用Ajax增强用户体验，为全球用户开发国际化应用程序，以及使用智能URL。

我们花了很多篇幅介绍symfony提供的专业环境的开发工具，向你展示如何利用单元测试、脚手架、插件、命令行，以及可扩展配置。

由于框架经常导致性能问题，本书还提供了很多监视和提高应用程序性能的技巧和技术，包括缓存和专家级的配置调整。

我们非常重视实用性，全书包含了大量的代码示例、专家建议、最佳实践和插图。

作为symfony项目的建立者和主要贡献者，我们希望提供足够的教育资源来满足快速增长的用户社区的需要。

我们认为这本书达到了这个目的，希望你也这么认为。

Francois和Fabien

<<symfony权威指南>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>