

全景式讲透Laravel理论、技术与项目实战

# Laravel

## 「 框架开发详解 」

从零基础到运用框架  
快速开发**PHP**网站

洪可郡 (KeJyun) 著

课外借



北京大学出版社  
PEKING UNIVERSITY PRESS

# Laravel 「框架开发详解」

从零基础到运用框架  
快速开发PHP网站

洪可郡 (Kejun) 著



北京大学出版社  
PEKING UNIVERSITY PRESS

## 内 容 提 要

Laravel是PHP的框架（Framework），给出了很多开发网站或API所需的工具及环境，经过简单的设置就可以完成数据的处理及显示，使开发者可以很优雅且快速地开发出各个不同的产品。本书适合有PHP基础，但不知道怎么选择框架的读者，不用框架的读者通过本书也能够明白它的好处。

本书以建置一个完整的电子商务网站为例，从环境建置、网站规划、数据库建置、发送电子邮件及网站优化等各个流程，教大家如何用Laravel去建置整个网站。建置过程中可以了解到整个Laravel的设计架构及逻辑，使读者得以初窥Laravel的面貌。

在了解整个Laravel的架构概念后，可以试着用Laravel给出的更多方法及扩展包，协助自己快速地进行产品的开发。越熟悉Laravel，越能感受到Laravel带给开发者优雅且快速的感受。让我们成为幸福的开发者吧！

### 图书在版编目（CIP）数据

Laravel框架开发详解：从零基础到运用框架快速开发PHP网站 / 洪可郡（KeJyun）著. —北京：北京大学出版社，2018.11

ISBN 978-7-301-29830-5

I. ①L… II. ①洪… III. ①网页制作工具—PHP语言—程序设计 IV. ①TP393.092 ②TP312

中国版本图书馆CIP数据核字（2018）第193914号

书 名 Laravel框架开发详解：从零基础到运用框架快速开发PHP网站

Laravel KUANGJIA KAIFA XIANGJIE: CONG LING JICHU DAO YUNYONG KUANGJIA KUAI SU KAIFA PHP WANGZHAN

著作责任者 洪可郡（KeJyun） 著

责任编辑 吴晓月

标准书号 ISBN 978-7-301-29830-5

出版发行 北京大学出版社

地 址 北京市海淀区成府路205号 100871

网 址 <http://www.pup.cn> 新浪微博：@北京大学出版社

电子信箱 [pup7@pup.cn](mailto:pup7@pup.cn)

电 话 邮购部 010-62752015 发行部 010-62750672 编辑部 010-62570390

印刷者 北京大学印刷厂

经 销 者 新华书店

787毫米×1092毫米 16开本 17.5印张 350千字

2018年11月第1版 2018年11月第1次印刷

印 数 1—3000册

定 价 69.00元

---

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究

举报电话：010-62752024 电子信箱：[fd@pup.pku.edu.cn](mailto:fd@pup.pku.edu.cn)

图书如有印装质量问题，请与出版部联系，电话：010-62756370

# 前言

## Introduction

PHP (Hypertext Preprocessor, 超级文本预处理语言) 是一种在服务器端执行的嵌入 HTML 文档的脚本语言, 现在被很多网站编程人员广泛运用。笔者目前为关键评论网资深工程师, 使用 PHP 开发网站已经有好几年的经验, 并且一直持续不断地精进网站开发的能力, 如数据库的设计规划、设计模式、大流量数据处理、Coding Style (代码风格)、团队合作开发等。笔者这几年使用过很多 PHP 框架, 如 Codeigniter、Kohana、FuelPHP 等, 也曾接触过 Rails 框架, 目的是学习更好的开发模式、积累更多的经验, 让自己在开发产品时能够更快速, 也能更好地维护代码。

偶然在技术教学影片平台看到 Jeffrey Way 介绍 Laravel, 笔者发现 Laravel 给出了所有在开发网站或 API (应用程序调用接口) 中所需要的东西, 并有详细的说明文档、强大的社区及弹性的扩展包。而且 Laravel 是使用自己最熟悉的 PHP 语言, 这让笔者一头扎进了 Laravel 的世界。

笔者当时学习的是 Laravel 4.2, 且在中国台湾还没有 Laravel 繁体中文文件。在学习 Laravel 的过程中, 笔者就顺手将官方文档翻译成繁体中文。随着 Laravel 版本的演进, 笔者也持续不断地学习最新的开发模式及技术。在学习过程中, 同时将自己学习的心得记录到“Laravel 5 学习笔记”中, 希望能替 Laravel 社区做一点贡献。

此次和北京大学出版社合作, 出版本书的简体中文版, 希望能够为更多读者提供学习帮助。

## 本书特点

本书不是一本详细介绍 Laravel 函数及功能的书, 也不是随着 Laravel 版本更新而会过时的书。本书会从无到有, 以使用 Laravel 构建一个电子商务网站作为案例, 带大家从宏观的角度完整了解 Laravel 的设计概念, 发现 Laravel 的美, 从而发现原来网页开发可以如此优雅且快速。

本书想让大家知道 Laravel 在网站开发中所扮演的角色, 知道它可以完成什么样的工作, 能解决哪些问题, 让大家对 Laravel 有一个初步完整的认识, 并在未来使用 Laravel 做更多进阶功能的开发时, 能够更加得心应手。

在了解了 Laravel 的全貌后, 如果想让自己在开发 Laravel 上学习更多进阶的用法, 可以到 Laravel 官方网站查阅更多、更详细的用法。

## 适合对象

本书适合的对象包括有基础网站开发经验的开发者、有开发网站需求的新创公司、有网站需要改版的上市公司，以及开始使用 Laravel 来进行网站开发的开发者及开发团队。

## 致谢

写这本书的过程中，最感谢爸爸、妈妈及妹妹的支持。他们体谅笔者需要牺牲家庭时间来完成这本书，在背后默默地支持笔者的创作。笔者将这本书献给他们。

感谢笔者在关键评论网的主管 Randy Hsieh（谢旻桦），他在笔者的写作过程中不断鼓励笔者，在笔者人生遇到困难及瓶颈时，总能给笔者一些不同的经验分享及指引，让笔者能够克服困难完成本书。

感谢关键评论网的创办人 Mario Yang（杨士范）。他指导笔者要怎样入门，让笔者有机会成为一位称职的作者。

感谢在这几年工作中，在技术上指导笔者，并陪伴笔者一起成长的 Deckran（小柯）、Rex Lu（卢建洲）、Sankai Liao（廖三凯）。没有他们在旁边指导，一起教学相长，不会有今日的笔者，也不会有这本书问世。

感谢 Shengyou Fan（范圣佑）努力地经营中国台湾 Laravel 社区，努力举办活动，推荐 Laravel 给所有的网站开发者及公司。

感谢博硕文化的总编辑 Simon（陈锦辉），在网络上看到笔者的“Laravel 5 学习笔记”，热情来信邀稿。没有他的邀稿，也不会有这本书出版的机会。

感谢博硕文化的编辑 Sam Wei（魏声圩）的照顾，细心地解答笔者在写书过程中遇到的任何问题，让本书能够更臻完美。

感谢在笔者人生中，包容笔者的任性，并持续耐心地在笔者身边提点笔者、陪伴笔者的好朋友们。他们在生活中任何直接或间接的慰问关心，都是笔者能够走到今天的最大养分及动力。

感谢全世界各地的 Laravel 开发者，没有你们的加入，Laravel 就没有今天的成长及规模。

在多年工作及开发数个产品的过程中，笔者觉得使用 Laravel 开发网站是一件令人幸福且开心的事。希望更多的开发者能够加入 Laravel 这个大家庭。

## 资源获取和交流

读者可通过下载链接<https://github.com/kejyun/laravel5-for-beginner-shop-laravel>进行源代码的下载，也可以通过扫描右侧二维码进行下载。有任何意见或疑问，可通过邮箱和笔者进行交流。笔者邮箱：[kejyun@gmail.com](mailto:kejyun@gmail.com)。



# 目 录

## Contents

第 1 章 Laravel 概述..... 1	3.2 情境分析..... 36
1.1 选择 Laravel 的原因..... 2	3.3 功能需求分析..... 37
1.2 选择 Laravel 的意义..... 2	3.4 数据表字段规划..... 39
1.3 如何选择一套适合的框架..... 4	3.5 结语..... 43
1.4 Laravel 特色与价值..... 9	第 4 章 网址设计规划..... 44
1.5 结语..... 10	4.1 网址设计规范..... 45
第 2 章 设置 Laravel 开发 环境..... 11	4.2 网址主从关系设计..... 45
2.1 环境建置及系统需求..... 12	4.3 网址路由 (Route)..... 48
2.2 Laravel 系统环境需求..... 12	4.4 结语..... 55
2.3 Laravel Homestead..... 13	第 5 章 控制器与模板设计 规划..... 56
2.4 安装 Laravel..... 22	5.1 控制器的位置..... 57
2.5 在 Windows 系统中安装并 设置环境..... 26	5.2 命名空间..... 58
2.6 结语..... 34	5.3 使用模板..... 60
第 3 章 电子商务网站规划..... 35	5.4 模板数据显示..... 61
3.1 前言..... 36	5.5 模板继承..... 62
	5.6 模板组件化..... 65

5.7 载入共用的 css 与 js 文件.....	66	7.14 新增用户数据至数据库 .....	110
5.8 结语 .....	68	7.15 发送用户注册通知信 .....	115
<b>第 6 章 数据库设置及建立.....</b>	<b>69</b>	7.16 结语 .....	117
6.1 连线设置 .....	70	<b>第 8 章 用户登录验证 .....</b>	<b>118</b>
6.2 连线到 MySQL.....	72	8.1 登录及退出登录路由 .....	119
6.3 .env 环境变量文件 .....	73	8.2 登录页控制器和模板 .....	120
6.4 设置环境变量.....	75	8.3 用户登录数据验证.....	121
6.5 env() 函数.....	76	8.4 取得用户数据.....	123
6.6 数据库迁移 ( Migration ) .....	76	8.5 导出 Eloquent SQL 语法.....	124
6.7 结语 .....	91	8.6 验证密码 .....	126
<b>第 7 章 用户注册.....</b>	<b>92</b>	8.7 用户登录 .....	127
7.1 注册路由 .....	93	8.8 Session 设置 .....	128
7.2 注册页控制器及模板.....	93	8.9 登录身份界面检视.....	129
7.3 注册页数据处理控制器 .....	96	8.10 用户退出登录.....	129
7.4 CSRF 保护.....	96	8.11 结语 .....	130
7.5 接收用户注册数据.....	97	<b>第 9 章 商品管理与浏览 .....</b>	<b>132</b>
7.6 数据验证 .....	98	9.1 商品路由 .....	133
7.7 验证规则 .....	101	9.2 建立商品页控制器.....	134
7.8 开始验证数据.....	102	9.3 商品编辑页控制器.....	135
7.9 显示验证错误信息.....	103	9.4 网站图片存储方式.....	136
7.10 验证信息语言设置.....	104	9.5 商品编辑页模板 .....	137
7.11 字段名称语言 .....	108	9.6 模板显示预设数据.....	140
7.12 验证错误表单数据保留 .....	108	9.7 权限管控 .....	140
7.13 密码加密 .....	109	9.8 建立中间件 .....	141

9.9 登录用户中间件 .....	145	11.1 加入 Github ID 字段 .....	182
9.10 表单 HTTP 方法设置 .....	147	11.2 执行 Migration 变更 .....	183
9.11 商品图片处理 .....	151	11.3 安装 laravel/socialite 扩展包 .....	183
9.12 裁切图片 .....	153	11.4 设置 Github 登录验证 数据 .....	184
9.13 管理商品 .....	154	11.5 设置 Github 登录路由 .....	185
9.14 结语 .....	157	11.6 设置 OAuth 重定向 URI .....	186
<b>第 10 章 购买商品 .....</b>	<b>158</b>	11.7 Github 登录控制器 .....	187
10.1 商品购买路由 .....	159	11.8 取得 Github 授权数据 .....	188
10.2 商品清单检视页控制器 .....	160	11.9 更新 Github ID 或注册 新用户 .....	189
10.3 商品清单检视页模板 .....	161	11.10 Github 登录网站 .....	192
10.4 商品单品检视页控制器 .....	163	11.11 结语 .....	192
10.5 商品单品检视页模板 .....	164	<b>第 12 章 工作延迟处理 .....</b>	<b>194</b>
10.6 商品购买处理数据验证 .....	165	12.1 队列 (Queue) 秘书 .....	195
10.7 数据交易处理 .....	167	12.2 建立工作事项 (Job) 便利贴 .....	195
10.8 交易数据锁定 .....	168	12.3 建立注册成功发信工作 内容 .....	196
10.9 交易检查及商品数据更新 .....	171	12.4 变更注册信发送方式 .....	197
10.10 交易异常处理 .....	172	12.5 使用 Redis 待办工作笔记 .....	198
10.11 记录交易记录 .....	173	12.6 设置 Redis 待办工作笔记 .....	199
10.12 检视用户交易记录 .....	175	12.7 记录工作事项便利贴 .....	199
10.13 交易记录与商品关联 .....	176		
10.14 获取交易的商品关联 数据 .....	177		
10.15 结语 .....	179		
<b>第 11 章 社区登录 .....</b>	<b>181</b>		



12.8 执行待办工作.....	200	14.4 侦测用户指定语言.....	228
12.9 队列秘书工作流程.....	200	14.5 结语.....	230
12.10 持续执行待办工作.....	201	第 15 章 挖掘更深.....	231
12.11 重新载入 Queue 程序.....	203	15.1 辅助函数.....	232
12.12 结语.....	203	15.2 Collection 数据集合对象.....	236
第 13 章 用户电子报.....	204	15.3 加解密.....	248
13.1 建立自定义 artisan 命令.....	205	15.4 错误与日志.....	248
13.2 发送最新商品电子报.....	206	15.5 HTTP 异常错误代码.....	250
13.3 测试发送电子报 artisan		15.6 artisan 命令.....	250
命令.....	212	15.7 缓存.....	252
13.4 设置排程工作.....	212	15.8 存储 ( Storage ).....	254
13.5 设置定时执行任务排程		15.9 结语.....	256
命令.....	214	第 16 章 你好, Laravel.....	257
13.6 任务排程命令信息.....	214	16.1 Laravel 文件夹结构.....	258
13.7 任务排程命令信息颜色.....	216	16.2 请求 ( Request ).....	259
13.8 工作事项 ( Job ) 优先		16.3 模板 ( Blade ).....	263
顺序.....	217	16.4 路由 ( Route ).....	265
13.9 结语.....	219	16.5 重定向 ( Redirect ).....	266
第 14 章 多国语言.....	220	16.6 CSRF 保护.....	266
14.1 语言文件设置.....	221	16.7 响应 ( Response ).....	267
14.2 模板语言设置.....	224	16.8 artisan 命令.....	268
14.3 用户指定语言.....	227	16.9 结语.....	271



第1章

---

**Laravel 概述**

## 1.1 选择 Laravel 的原因

Laravel 是一套 PHP 的开发框架（Framework），提供了很多开发网站或 API 所需的工具及环境，经过简单的设置就可以完成数据的处理及显示，使开发者可以很优雅且快速地开发出不同的产品。本书适合有 PHP 基础但不知道怎样选择框架，或者不会用框架却想了解它的人使用。

虽然 WordPress（一种使用 PHP 语言开发的博客平台）也能够建站，但如果有定制化需求，要开发各式各样的网站，或者给出 App 使用的 API，就只能选择用框架，而 Laravel 是目前最受欢迎的开发框架。

本书将介绍为什么要使用框架，以及了解框架的优缺点后怎样选择框架，并用框架快速构建一个网站。如非必要，书中避免使用专业技术用语，尽量使用最生活化且易懂的例子，带大家进入 Laravel 的世界。

## 1.2 选择 Laravel 的意义

在开发程序时，要不要选择框架，是以工作模式的 SOP（Standard Operating Procedure，标准作业程序）而定的。那么，什么是工作模式的 SOP 呢？

在开发 Web 程序中，以“查找商品”为例。常见的做法是将“取得用户传入数据”“验证传入数据”“数据库连线”“存取数据库数据”“显示取得的数据”等程序逻辑分别做出来，而这就算是一种工作模式的 SOP。上述的代码如下。

```
<?php
//=== 取得用户传入数据 ===
$merchandise_keyword = $_GET['query'];

//=== 验证传入数据 ===
if (empty($merchandise_keyword) OR strlen($merchandise_keyword)
== 0) {
    //数据验证错误
    die("传入数据错误");
}

//=== 数据库连线 ===
$db_server_name = 'localhost';
$db_username = 'username';
```

```
$db_password = 'password';

//建立数据库连线
$db_connection = new mysqli($db_server_name, $db_username, $db_
password);

//检查数据库连线
if ($db_connection->connect_error) {
    die("数据库连线失败: " . $db_connection->connect_error);
}

//=== 存取数据库数据 ===
$sql = "SELECT id, name, price
FROM `merchandise`
WHERE `name` LIKE '%{$merchandise_keyword}%'
LIMIT 20";
$result = $conn->query($sql);

//=== 显示取得的数据 ===
if ($result->num_rows > 0) {
    //输出查找商品结果
    while($row = $result->fetch_assoc()) {
        echo " 编号: " . $row["id"];
        echo " 商品名称: " . $row["name"];
        echo " 价格: " . $row["price"];
        echo "<br>";
    }
} else {
    echo "找不到任何商品";
}

//关闭数据库连线
$db_connection->close();
```

换句话说，用一般的方式直接开发是一种工作模式的 SOP，需要什么样的程序就直接编写，通常一个 PHP 文件就可以完成。

这样的工作模式没有对错，只是选择问题。当只需取得数据库中某些数据的功能需求时，无须任何系统化程序架构，只要给予的时间足够多就行，而且也不需要与其他人合作，只需自己写一些小程序，那么这种开发的工作模式或许是最适合的。

但是，当需要系统化的程序架构，希望不同类型的程序可以放在不同的文件中，就像商品的程序放在 `merchandise.php` 中，而用户的程序放在 `user.php` 中时，该怎么办呢？如果需要快速做完这个工程，那么不同工程都需要的功能，如账户身份认证（Auth）、用户输入数据验证（Validation）、数据库连线存取（Database）等，就要能重复使用。这个工程

是需要与其他人一起合作、互相帮助的，最好是有相近的编程风格（Coding Style），这样在帮彼此做代码译审（Coding Review）和除错（Debug）时，能够更快进入状态。当有上述这些需求时，选择一套适合自己工作模式的框架或许是个不错的选择。

工作模式 SOP 就是做事的方法与流程，每个人对任何事情都有不同的看法及做法。就像做菜一样，有人喜欢先放麻油再放酱油，有人则相反；有人希望比较咸一点，酱油或盐都会放多一点，但煮出来的菜不一定适合每个不同口味的人。不能说谁的厨艺好，只能说做的菜适不适合。但一个拥有好厨艺的人，说明他做菜的方式与做出来的菜品，很受大家的青睐，值得大家去学习。当学会他的做菜方式后，或许其他人也能够练出像他一样的好厨艺。

综上所述，当这个工程需要与其他人一起合作时，就不能采用随意工作模式的 SOP 了，而是需要“框架”。而选择一套框架，也是选择一套程序的设计架构方法与工作流程。不同的框架有不同的优缺点，只需去了解这些框架的优缺点，找到适合自己需要的即可。

## 1.3 如何选择一套适合的框架

在每次开发不同的 Web 程序时，除了每个工程会有的定制化商业逻辑外，还有很多功能是不同的工程都需要的，如账户身份认证（Auth）、用户输入数据验证（Validation）、数据库连线存取（Database）、程序路由逻辑（Routing）、画面显示（View）、数据安全性处理（Security）及其他功能。而在 PHP 语言中，不同的框架如雨后春笋般冒出来，其实都是要尝试解决上述问题。但框架除了需要解决这些常见的开发逻辑外，最重要的还是框架的维护性及扩充性（Composer）。

### 1.3.1 维护性

产品都是构造在框架上，再延伸出来的。然而，即使再好的框架，也是工程师写出来的，工程师也是人，所以写出来的框架难免会有 bug 存在。而当发现框架有 bug 时，表示整个构造在框架上的网站都是有 bug 存在的。如果 bug 不处理，那么整个网站都将暴露在 bug 中，随时都会出现安全性的问题。

## 1.3.2 长期支持

除了 Laravel 开发者 Taylor Otwell 宣布会全职投入开发 Laravel 外，最重要的还是 Laravel 目前在 5.1 及 5.5 版本的基础上会给出长期支持（LTS: Long-term Support）的版本，这也是第一个给出 LTS 版本的 PHP 框架。

Laravel 保证 LTS 版本，至少给出 2 年的 bug 修复期及 3 年的安全性修复期。而非 LTS 版本的 Laravel，则给出 6 个月的 bug 修复期及 1 年的安全性修复期，框架支持 LTS 则表示，开发的产品不需要随着框架的版本演进，而是被迫进行升级。

## 1.3.3 框架更新问题

开发一个商业用途的网站，如果框架进行版本更新时，会有以下两种选择。

### 1. 沿用旧版本框架

当选择沿用旧版本框架时，如果官方宣布不再支持 bug 修复及安全性更新，你的产品将会永远地暴露在 bug 及不安全的环境中。而且通常框架和第三方扩展包的开发者，也会试着让扩展包能够支持新版本的框架。但对于支持旧版本的扩展包，除非是扩展包开发者想去做，否则通常不会去考虑持续维护旧版本的扩展包。所以，不只是框架可能会产生 bug 及安全性问题，第三方扩展包也会产生这样的问题。

### 2. 更新框架版本

当选择更新框架版本时，表明你至少要花费几个月的时间（视产品规模而定）去进行框架和程序的更新及测试。当更新时发现 bug 或旧的程序无法与新版本的框架相容时，就要花费更多的时间对旧版本进行修复。如果没有写单元测试（Unit Test），就会导致有些程序的 bug 没有测试到，从而导致上线时才发现要去修复这些因更新框架而产生的 bug。在整个更新期间，都是在对原本就可以正常运作的程序进行修复重构。在这个阶段，不管是对产品还是对公司，工程师都不可能创造新价值。产品对于公司最重要的事，是能够持续性地做出更好的服务，给用户或公司带来更大的价值，替公司创造更多的营收。除非这个框架在更新后，可使产品给出的价值远远大于在这个期间内开发更多的功能服务所给出的价值。否则对于公司来说，更新框架是在浪费时间，浪费公司的成本。现在是信息大爆炸的时代，随时都有新的挑战发生，公司需要有随时应对市场变化的能力，所以，框架的更新可能会让公司失去应变能力，失去更多抢得先机的机会。

### 1.3.4 框架版本的选择

现在的技术随时都在更新，工程师必须要随时掌握重要的新技术及其发展趋势，才能让自己在市场上保持一定的竞争力，而公司产品却不能随时去追求新技术。

从 Laravel 框架维护的有效期来看，建议公司在商业用途的产品中，选择比较稳定的 LTS 版本的 Laravel。也因为 Laravel 给出了 LTS 长期的维护支持，所以，大型的企业工程可以放心地使用 Laravel 进行开发。而如果是比较小的工程，建议选择最新版本的 Laravel，让自己随时掌握新的技术趋势，而且因为工程较小，在进行框架版本更新时，不会有很高的更新成本。

### 1.3.5 扩充性 (Composer)

扩充性是指当框架有不足的部分时，其他开发者可以很方便地协助开发相关扩展包，并很容易地将其整合到该框架中。可以协同合作的人越多，就可以让框架有更多的扩充性

扩展包。虽然协作的人越多，使用的人也会越多，但对于这些第三方扩展包的讨论及除错也会越来越多。扩充性越好，对该框架的社区及讨论的人也会越来越多，框架的稳定性也会越来越高，这是一个很好的正向循环。

目前 PHP 是使用 Composer 做扩展包的相依性管理系统，如图1-1所示，不同扩展包之间不同版本相互依赖，使用扩展包管理系统在指定特定版本号后（如 laravel/ laravel 5.5.\*），可以保证目前执行的扩展包版本都是可以正确执行的。所以当在安装或更新扩展包时，会以目前扩展包版本号当作参考标的进行安装或更新。当扩展包有大改版时，如 Laravel 从5.5.\* 版本更新为5.6.\*

版本时，扩展包管理系统不会做跨版本号的扩展包更新，来维持并保证扩展包能够在目前版本需求下正常运作。而 Laravel 也是用 Composer 做扩展包管理工具的。

使用 Composer 指定安装 Laravel 5.5.\* 版本。

```
composer create-project --prefer-dist laravel/laravel blog "5.5.*"
```

在其他语言或环境中也有类似PHP Composer 的扩展包管理系统，如 Node.js 的 npm 或 Python 的 pip，都是类似的扩展包管理工具，而目前安装 Laravel 是一定要通过 Composer



Dependency Manager for PHP

图 1-1 Composer-PHP 扩展包相依性  
管理工具

进行安装的。

目前支持 Composer 的所有扩展包都可以在 Packagist 网站中找到,如图 1-2 所示。若是需要找 Laravel 相关的扩展包,可以在网址列末尾加上 tags 查找,也可以到 Packalyst 网站上查找被整理过的 Laravel 相关的扩展包,如图 1-3 所示。或者直接在网站上搜索,例如,要找 Laravel 的 Excel 相关扩展包,则在搜索框中输入框架名称 (Laravel) 及扩展包关键字 (Excel) 查找即可。

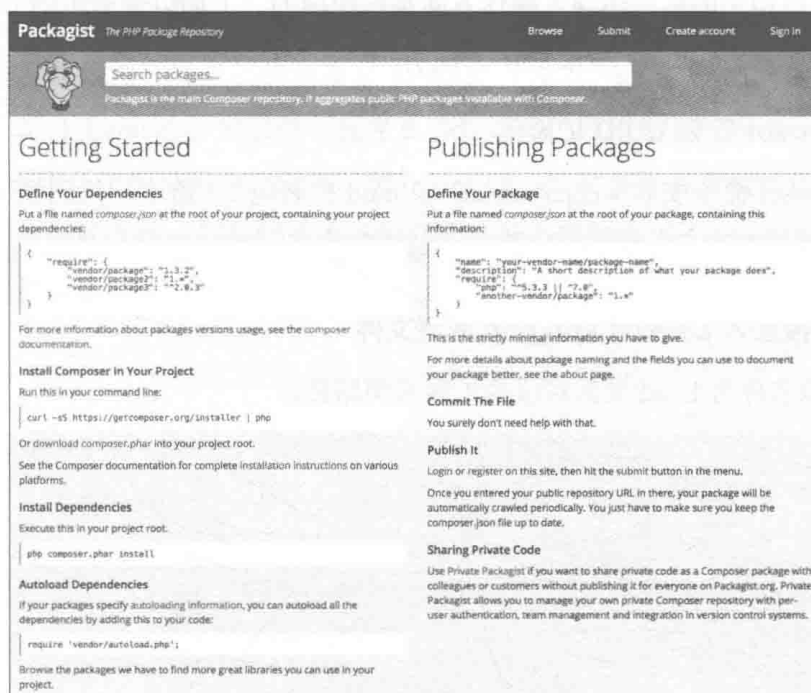


图 1-2 Packagist-PHP 扩展包资源库

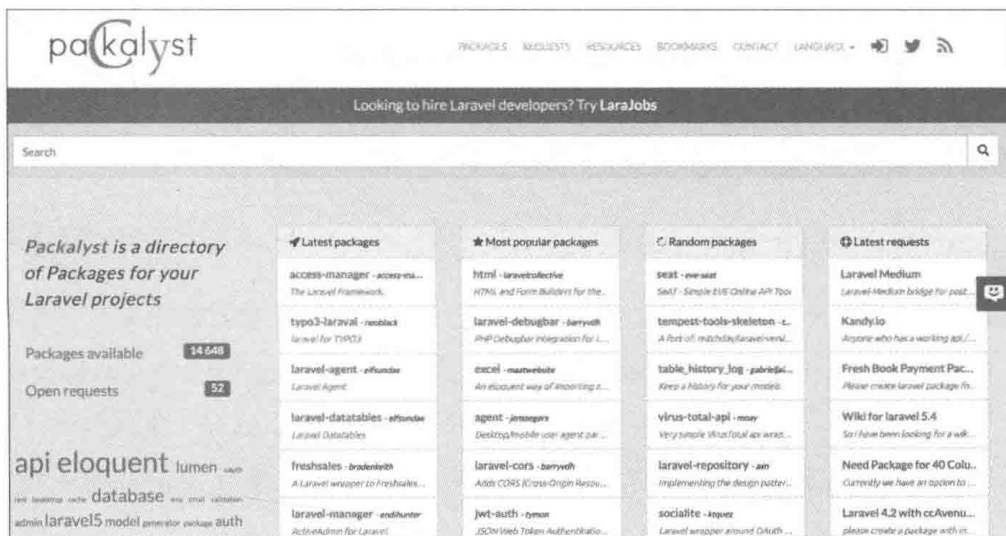


图 1-3 Packalyst-Laravel 扩展包



## 1.3.6 扩展包安装管理

在做工程开发时，偶尔会遇到一些特殊的需求，如数据输出、图片转档或社区 API 等。当遇到这类需求时，希望能够找到相关的扩展包直接导入工程中。若找不到相关的扩展包，再自己开发此扩展包。在导入扩展包到工程中时，很容易遇到与目前程序架构兼容性的问题，没有办法快速且正确地导入。

而 Laravel 在相关的扩展包导入操作方面是很便捷的，下面以最简单的 UUID 产生器扩展包（webpatser/laravel-uuid）安装为例来说明。

### 1. 使用 Composer 安装 UUID 扩展包

使用 Composer 命令安装 webpatser/laravel-uuid 扩展包 2.\* 版本。代码如下。

```
composer require "webpatser/laravel-uuid:2.*"
```

### 2. 扩展包类路径加入 Laravel app.php 配置文件

指定扩展包名称为 Uuid 及其对应的扩展包类路径。

```
<?php
//config/app.php

return [
    'aliases' => [
        'Uuid' => Webpatser\Uuid\Uuid::class,
    ],
];
```

### 3. 立即使用扩展包

在程序中立即使用 Uuid 扩展包产生需要的 UUID 数据。

```
<?php
//app/Http/Controllers/UserController.php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Uuid;

class UserController extends Controller {
    public function uuidTest() {
        //产生 UUID
        $user_uuid = Uuid::generate();
    }
}
```