Публикация на тему

Структура и жизненный цикл фрэймворка Laravel

Laravel - php-фрэймворк для разработки web-прилоежний. В структуре папок Laravel использует архитектурный шаблон проектирования HMVC. Улучшение стандартного HMVC происходит за счет облечения моделей и контроллей, логику которых принимают на себя вспомогательные файлы - request, middleware, resource, events и другие.

Автор
Михалькевич Александр Викторович

Публикация

Наименование Структура и жизненный цикл фрэймворка Laravel

Автор А.В.Михалькевич

Специальность Laravel - php-фрэймворк для разработки web-прилоежний. В структуре папок Laravel использует архитектурный шаблон проектирования HMVC. Улучшение стандартного HMVC происходит за счет облечения моделей и контроллей, логику которых принимают на себя вспомогательные файлы - request, middleware, resource, events и другие.,

Анотация

Anotation in English

Ключевые слова

Количество символов 6643

Содержание

Введение
1 Структура папок
2 Жизненный цикл запроса
Заключение
Список использованных источников
Приложения

Введение

1 Структура папок

Со структурой папок фрэймворка Laravel можно ознакомиться по ссылке https://github.com/laravel/laravel

app/ Главная рабочая папка всего фрэймворка. По умолчанию, это папка в пространстве имен App. Консольные комманды, обработчики исключительных ситуаций, контроллеры, модели и дргуие классы находятся здесь.

bootstrap/

Папка для конфигурационных файлов автозагрузки.

config/

Папка конфигурационных файлов.

database/

В папке database находятся папка migrations (для файлов миграций баз данных), seeds (предворительные данные таблиц базы данных) и factories (настройки для моделей).

public/

Корневая папка проекта. В этой папке находится файл index.php и .htaccess, которые загружаются первыми, а также папкимедиафайлов (css, js, изображения и др.)

resources/

Папка для шаблонов.

routes/

Папка для route-ов (маршрутов).

storage/

Папка для хранения временных файлов, создаваемых фрэймворком.

Папки внутри storage должны быть доступны веб-серверу для записи. Если вы устанавливаете фреймворк на Linux или MacOS, открыть папки на запись можно командой chmod -R 777 storage.

tests/

Папка содержит файлы автоматических тестов.

vendor/

Папка содержит composer – зависимости.

Основыные рабочие папки - это app/, resources, routes и корневая.

2 Жизненный цикл запроса

Точка входа для всех запросов фрэймворка начинается с файла index.php. Данный файл не содержит много кода. Основная его задача – загрузка необходимых файлов: **autoload.php и app.php**. А также дальнейшее перенаправление запроса.

Далее входящий запрос, в зависимости от типа запроса, отправляется либо в модуль, обрабатывающий консольные команды (**консольное ядро**), либо в **ядро фрэймворка**,

обрабатывающее http-запросы.

Сосредоточимся на ядре HTTP, который находится по адресу app/Http/Kernel.php. Данный файл является расширением класса Illuminate\Foundation\Http\Kernel (ядра фрэймворка). Ядро фрэймворка можно представить в виде большого черного ящика, который обрабатывает запросы, а также определяет перечень промежуточного программного обеспечения (специальных классов наследуемых от класса **middleware**), которые должен пройти фрэймворк прежде чем выдаст ответ.

Далее загружаются поставщики услуг для приложения (**servise providers**). Все поставщики услуг по применению настроены в массиве провайдеров конфигурационного файла config/app.php. Далее метод register() определяет необхоимые классы поставщиков услуг. Если некоторые поставщики услуг необходимо запустить сразу, то это можно выполнить в методе boot().

Сервис-провайдеры несут ответственность за все компоненты по фреймворку, такие как компоненты базы данных, очереди, проверки и маршрутизации. Так как через них можно настроить и загрузить все функции, предлагаемые фрэймворком, поставщики услуг являются наиболее важным аспектом всего процесса начальной загрузки Laravel.

И только после того, как все поставщики услуг были зарегистированы (или сразу выполнены методом boot()), запрос попадает в маршрутизатор. Маршрутизатор, как правило, отправляет запрос в какой-либо из контроллеров.

На этом запрос обработан, и в контроллере должен быть прописан ответ фрэймворка. Кстати, если ответ ожидается коротким, то можно обойтись без контроллера: ответы фрэймворка можно писать сразу в маршрутизаторе.

Есть много отклонений и различных вариантов путей запроса, но неизменно путь проходит три опорные точки, на которые надо обратить внимание:

1. Роуты - app/routes.php
2. Контроллеры - app/controllers/
3. Отображения (виды) - app/views/

Заключение

Список использованных источников

Приложения