**What's New in Laravel 5.4**

/

И снова настал тот день! Laravel находится на пороге выпуска версии 5.4. Вместе с ней приходит целый ряд новых оптимизаций и возможностей. Пометка почты, сообщения более высокого порядка, Dusk, фасады в реальном времени... да, пожалуйста! В этой серии я проведу вас в темпе по тому, что нового в Laravel 5.4.

/

[What's New in Laravel 5.4:](https://laracasts.com/series/whats-new-in-laravel-5-4)Sanitizing Middleware

**EPISODE 1 • RUN TIME 1:44**

5.4 содержит несоклько новых middleware/давайте перейдем в http – kernel и прямо тут trimstrings и convert empty strings to null/эти двое на самом деле очищают middleware и я думаю что они очень очень полезны/давайте посморим что они делают/я перейду в мой routes web файл и вы увидите что тут я добавил пару маршрутов/одну для отображения формы и другую для ее обработки/и все что это делает это просто возвращает данные с формы/окей/откроем form blade и вы увидите тут я установил bootstrap форму/тут у нас есть поле для имени/секция для комментария и кнопка отправить/самое базовая форма/ и если сейчас мы перейдем в хром/напечатаем чушь и добавим кучу пробелов в конце и тоже самое тут/и после того как я отправил данные вы увидите что эти пробелы были срезаны/но раньше вы бы их все равно увидели/это новая фишка в этой версии/если вернемся обратно в kernel и закоментим/вернемся в хром и повторим/то теперь вы увидите все эти пробелы/но что насчет другой?/ну на самом деле это делает то как оно называется/если вы разрешаете не заполнять строки/например имя/и после того как я отправил это/пустая строка будет автоматически преобразована в null что особенно полезно в некоторых ситуациях когда в вашей базе данных null значение по умолчанию но когда вы обновляете массово этот null может быть изменен в пустую строку/и еще раз в прошлом вы бы просто получили пустую строку как вы видите здесь/и это все что вам нужно знать об этом

/

Начало этой серии представляет собой обзор двух новых middleware, которые входят в laravel по умолчанию: TrimStringsи ConvertEmptyStringsToNull

. Фактически, для каждого запроса формы Laravel автоматически обрезает все пробелы в конце строки, а также преобразует все пустые поля запроса в null

.

/

# [What's New in Laravel 5.4:](https://laracasts.com/series/whats-new-in-laravel-5-4)Higher Order Collection Messages

**EPISODE 2 • RUN TIME 2:41**

Следующий стафф это сообщения более высокого порядка/у меня тут имеется класс post и представьте что у нас есть метод архиватор для него/и все что он будет делать это обновлять колонку активный на значение false/теперь давайте используем tinker для этого/скажем что у нас есть куча постов и мы их захватим и далее то что мы хотим сделать это архивировать все посты для представленного набора/окей/традиционно мы бы использовали foreach или бы вызвали each метод для вашей коллекции и затем могли бы написать/для каждого поста мы применяем функцию архиватор/и если мы запустим это – конечно же каждый посту будет иметь статус активный false/и это точно сработает/однако теперь мы можем использовать сокращение которые мы зовем сообщения более высокого порядка/звучит немного напыщено но на деле это просто сокращения/в этом конкретном случае мы говорим – для каждого поста вызвать метод archive/окей/но теперь мы просто можем сказать – each – динамическое свойство тут-и далее просто archive/ и мы получили тоже самое что и до этого/так что давайте сделаем следующее/давайте сделаем обратное действие/создадим метод activate и в основном это тот же код/но теперь мы установим тут true/окей давайте запустим это еще один раз/и тепеь я пишу post – для каждого из них я хочу вызвать метод activate/запустим/и честно говоря магия подобного рода сводит людей с ума))/и вы должны попробовать это тоже/вы увидите что это было сделано и для других методов в коллекции/представим что из нашего набора постов мы бы хотели получить все заголовки/чтож как вы знаете у нас есть метод pluck и мы могли бы написать pluck title/и это сработало отлично/однако возможно вы не знаете этот метод и вы напишете titles и это длинный способ сделать это/posts map и для каждого поста я хочу вернуть заголовок поста/это очень долгоно мы все еще получили тоже самое/однако что мы могли бы сделать – posts map и для каждого поста мы возвращаем title/и опять мы привяжем это к переменной titles/вот мы получили тот же самый результат/чтож и в этом прмере мы использовали метод pluck и вы видите насколько это гибко и удобно/и это будет работать также для кучи других методов таких как contains/each/sum или filter или map/их тут много и вы их можете найти в документации/не заставляйте себя применять это если вам кажется что это менее понятно/но вабще это очень полезно

/

Laravel 5.4 включает в себя поддержку обмена сообщениями более высокого порядка. Это необычный термин, который по сути означает сокращение. Что делать, если вместо фильтрации коллекции, передачи closure и возвращения каких либо данных, вы можете просто ввести: $collection->filter->isArchived()? Насколько это круто? Я покажу вам, как это все работает в этом эпизоде.

/

# [What's New in Laravel 5.4:](https://laracasts.com/series/whats-new-in-laravel-5-4)Fluent Routing

**EPISODE 4 • RUN TIME 1:07**

Следующим шагом будет обзор некоторых обновлений для элементов routes/в прошлом если вы хотели, например, закрепить имя за некоторым маршрутом вы бы делали нечто подобное/и дальше если бы вы захотели применить группу/вы бы вызвали group метод и затем prefix и любое пространство имен которое вы захотите/и затем сослаться на closure тут и это то место где ваши маршруты будут размещаться/и нет ничего плохово в этом но теперь это стало более гладким/например/вместо того чтобы определять ваше имя в конце если вы хотите вы можете тут написать route name home и затем получить этот маршрут/и затем такие вещи как префикс тоже упростились/скажем/я хочу применить префикс api и затем я хочу эту группу/клево/но дальше вы могли бы применить middleware и я могу сказать/ну например у нас есть дашборд и мы хотим применить middleware к этому/конечно вы можете сделать это внутри контроллера/но если хотите вы можете применить его прямо тут/ middleware и затем мы применяет middleware/как то так/и это пока все)

/ Компонент маршрутизации Laravel 5.4 был обновлен для поддержки более плавного синтаксиса маршрутизации. Больше не требуется ссылаться на массив неудобных параметров, если требуется группа маршрутов, префикс или имя. /

# [What's New in Laravel 5.4:](https://laracasts.com/series/whats-new-in-laravel-5-4)Resourceful Controllers With Model Boilerplate

**EPISODE 5 • RUN TIME 1:32**

Взглянте на это/если мы запустим php artisan help make controller вы увидите тут опцию черточка м /сгенерировать контроллер ресурсов для указанной модели/давайте попробуйем/давайте создадим контроллер для наших posts и затем я хочу обозначить эту модель как пост/окей/ когда мы делаем это/давайте посмотрим/вы увидите что да мы получили контроллер post и да у нас есть все ресурсные методы но также все релевантно/вы увидите что тут теперь также указан post/это очень удобно на самом деле/вы также знаете что если мы воспользуемся преимуществом связывания маршрутов это будет значить что когда вы вызываете этот метод вы тут же получите пост/давайте попробуем/напишем вернуть пост и затем установим маршрут для этого/как насчет – когда вы получаете get запрос по адресу пост слеш определенный пост слеш edit назначить postcontroller edit/ давайте посмотрим как оно в браузере/перейдем к определеному посту и вот результат/так как laravel имеет функцию связывания маршрутов по умолчанию то если вы заметите эту wildcard которая соответствует названию этой переменной и соответственно это отслеживает пост который имеет этот идентификатор/и это все что оно делает/так что когда вы требуете контроллер ресурсов и вы не хотите делать это вручную и импортировать связанный класс – то просто запустите php artisan make controller и после добавьте флаг model

/

При создании контроллера из командной строки (php artisan make:controller) теперь вы обнаружите новый –model флаг. При вызове с именем модели, Laravel автоматически сгенерирует ресурсный контроллер, который имеет необходимый функционал для того чтобы добавить эту модель во все релевантные действия контроллера.

/