# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)

Из ниоткуда, Microsoft ввязывается в войну редакторов с невероятным предложением, из-за которого команда Sublime Text сильно разволновалась. Но на самом деле это просто может превзойти Sublime! Так что давайте пройдемся вместе по моей демонстрации входов, выходов, советов и техник. Поздоровайтесь с новым лучшим другом: **[Visual Studio Code](https://code.visualstudio.com/)**.

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)Essential File Manipulation

**EPISODE 2 • RUN TIME 3:16**

В любое время когда я начинаю пользоваться новым редактором кода я пытаюсь оптимизировать мой рабочий процесс настолько насколько это возможно/это может включать создание редактирование удаление файлов и на самом деле достаточно неудобно пользоваться мышью для создания и подтверждения создания фалойв/это полностью замедляет меня/так тчо я стараюсь сделать это проще и интуитивней насколько можно/в настоящий момент я могу создать новый файл нажав command N/попробуем и затем command S чтобы сохранить его и это откроет привычный нам диалог/мы также можем выбрать папку -правый клик -новый файл и затем сделать это прямо тут/однако тут есть расширение которое я бы хотел подтянуть/поэтому я открою панель задач нажам shift command p и давайте найдем что то связанное с отображением расшиений/и вы увидите что у нас есть сокращение - shift command Х которое вы разумеется можете изменить/расширение называется advance new file/ и это расширение в оригинале для sublime/я не очень знаком вот с этим и я в основном испольхую вот этот/мы подтянем его/и вы заметите что жто открое диалог как у панели задач где вы можете создать новый файл/поэтому я перезагружусь и давайте попробуем/это должно выскочить в панели команд и сейчас это связано с option command N/и это также как в сублайм/давайте попробуем создать контроллер и вы видите как мы собираемся указаь директорию – не полное имя/поэтмоу например если я хочу контроллер я сделаю вот так – foocontroller php/сохраню это и нажму shift command E чтобы вернуть мое древо файлов/вот он/давайте еще что то сделаем/нажму command backspace чтобы удалить его и снова command option N/мы создадим контролелр foocontroller php и как вы видите/это все работает достаточно быстро/но вабще то мне не нравится переходить сюда и затем удалять это/я знаю что все еще быстро/но я хотел бы сделать это еще более гибким/сейчас если я зажму shift command Р и вызову панель команд – да я не вижу тут никаких сокращений/давайте посмотрим есть ли тут расширение/мы собираемся найти file utils/это предоставляет удобный способ по созданию-дублкации/изменени названия и удалению фалов и директорий/мне это нравится и я подтяну его и затем перезагружу/окей давайте епосмотрим что он умеет/в панели команд пишу файл и тут множество команд/все они доступны после установки расширения/скажем мы хотим сделать дубликат этого фала/вы можете назначить горячую клавишу но обычно вы не используете это так часто поэтому просто можете напечатать duplicate/дадим название и сомтрите – это было очень быстро/но потом я понял что мне он не нужен так что посмотрим shift command Р - delete/выберем и вот мы закончили/или что более популярно – изменить имя/скажем я хочу назвать это member - shift command P – rename/member php и готово/но это не phpstorm так что не забудьтеобновить имя класса тут/и теперь я с удобством могу манипулировать фалами и увидимся в следующем эпизоде

/

Чистый рабочий процесс для обращения с файлами это основа основ. В этом видео вы узнаете, как оптимизировать процесс создания, удаления, переименования и дублирования файлов.

/

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)The Glorious Built-in Terminal

Если вы пришли сюда из atom или sublime text очень вероятно что вы перезодите каждый раз в терминал для запуска типичных команд/может вы используете git или вам нужно запустить команду artisan для laravel/любые из эитх вещей/чтоже альтернативно code имеет встроенный терминал который работает очень очень хорошо/взгляните прямо сюда – control tildа поднимет окно терминала и уже тут если вам нужно вы можете выполнить все эти команды/и как вы тут видите вы можете создать любое количество вкладок и потом закрыть их/что значит что вы можете делать такие вещи как эта/если у меня есть установленный laravel я мог бы использовать laravel mix webpack/чтож я мог вы напечатать npm install тут и подтянуть все мои расширения/и вот я могу запустить мой watcher/нажать control tilda чтобы закрыть его/и все готово/так например я могу отредактировать некоорый файл тут/как то так/сохранить это и все это скомпилируется на фоне/как то так/и сейчас эта вкладка была занята и я мог бы открыть новую вкладку для моих стандартных команд и вы можете переключаться между ними когда вам нужно/вот что важно держать в голове что некоторый расширения сделают некоторую работу за вас/например/с расширением laravel artisan которое мы могли бы подтянуть вам больше не нужно переключаться в терминал для выполнения artisan команд/тоже самое будет работать и для git или даже для запуска комплекта phpunit/тем не менее как любой разработчик вы будете использовтаь терминал постоянно поэтому это очень приятно что вы можете поднять его простой комбинацией control tilda

/

Если вы привыкли открывать отдельный Terminalили окно iTerm каждый раз, когда вам нужно вызвать консольную команду, если вы хотите, вы можете выполнить это прямо из Visual Studio Code, через его терминальную панель. Ctrl-`- ваш друг.

/

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)Git Ready

**EPISODE 7 • RUN TIME 2:44**

Давайте теперь поговорим о git/я нажму control tilda для отображения терминала где мы инициализируем git/ добавить все и между прочим я могу наблюдать как в сайдбаре отображаются изменения и да это все бесплатно/давайте сделаем коммит с сообщением Установить фреймворк/и сейчас это исчезло/теперь я могу написать git log чтобы увидеть это/хорошо/и да кстате вы в любое время можете нажать command k чобы очистить все сообщения тут/поэтому если у вас есть тут множество всего - command k и это пропало/в любом случае/давайте скроем терминал сейчас нажав command tilde/перейдем в класс member и сделаем изменение внутри него/и заметьте еще раз/пока я тут что то меняю тут подсвечивается изменение в коде/и это круто потмоу что если я кликну сюда я увижу разницу/и как пример давате представим что я тут возвращаю foo b теперь shift command p и вы увидите что у нас тут доступно множество git команд/вы вабще то можете делать это через терминал и если честно я так и делаю но если пожелаете можете делать это тут/и теперь мы можем перевести в stage все эти изменения и давайте посмотрим на это/перейдем в view в source control и тут я могу видеть все изменения с привлекательным предпросмотром тут/и я могу также написать тут для любого изменения сообщенеи/и это все/теперь если я захочу изменить это снова это тут же отразится в source control и это то что вы бы хотели скрывать или открывать когда вам нужно/и по умолчанию это назначено на control shift G/это немного неудобно и вы можете изменить это в любое время/так давайте посмотрим на изменения которые мы тут сделали/мы изменили foo на foobar/и кстате если вам не нравится разделение экранов вы можете переключиь на подстрочный вид/и это то что я часто делаю/если честно этого досаточно для меня/но некоторые люди идут дальше и устанавливают пакет git lens/вы можете посомтреть тут что это делает/ и в основном это дает вам больше возможностей для фильтра коммитов и тут такжы вы можете увидеть git blame что достаточно клево/что касается меня – я не использую это слишком часто/но вы возможно захотите это подтянуть/и еще раз представим ваш рабочий проесс/вы редактируете контроллер/например тут вы добавляете некоторую функцию/далее вы переключаетесь используя control shift g по умолчанию/чтобы увидеть все ваши изменения если хотите/можете перевести в stage это все или по отдельности и затем закомитить/иначе вы можете отменить все изменения использузя эту кнопку и затем увидеть как это было прежде/вот/все очень просто и полезно/эо все зависит от вас если вы хотите делать все через терминал или через интерфейс

/

Visual Studio Code предлагает первоклассную поддержку git по умолчанию. Давайте взглянем.

/

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)Laravel Artisan Shortcuts

**EPISODE 8 • RUN TIME 1:34**

Сейчас у нас есть установленный проект laravel и мочему бы нам не поискать laravel в расширениях и посмотрим что мы сможем найти/да вот у нас тут снипеты и тому подобное но давайте лучшие взглянем на laravel atrisan/это позволит нам запустить любую из этих команд используя панель команд/достаточно полезно/установим и перезагрузим/теперьыhift command p artisan и вот это все/давайте просто отобразим список всех маршрутов/вот что мы имеем/у нас тут пока только основная линия/ окей дальше/почему бы нам не создать контроллер/я могу написать только art controller и это все или можно просто controller/foobar controller/должен ли я создать ресурсный контроллер?конечно/и на выходе это создало нам файл и тут же показало нам его/если я зажму shift command И для того чтобы показать сайдбар/конечно же я увижу этот новый фалй здесь/или control shift g для управления ресурсами и вот этот файл тоже тут/великолепно/возможно мы также создаем модель/назовем ее foobar/должны ли мы сделать миграцию?да/и ответ нет для создания нового контроллера для нее/да это задаст общие вопросы относительно опций/вот этот файл и у нас также должна быть миграция create\_foobars\_table /и например мы тутмогли бы написать/table string заголовок/и очень быстро настроим соединение с базой данных sqlite и база данных memory/окей/теперь shift command p/migrate и это сработало/и это все

/

Для пользователей Laravel расширение Artisan позволяет вызвать любую команду artisan

прямо из GUI. Это значит, что в пределах нескольких секунд, вы можете сгенерировать и нагрузить ваш новый контроллер.

/

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)Better PHPUnit Execution

**EPISODE 10 • RUN TIME 2:52**

Как php разработчик ускорение рабочего процесса в работе с php unit вроде как жизненная необходимость для меня/поэтому я открываю встроенный терминал/печатаю phpunit тут и это меня устраивает/однако/если вы хотите давайте посмотрим что мы могли бы тут улучшить/давайте нажмем command x чтобы потсмореть мои расширения/я собираюсь найти пакет под названием better php unit/это было создано caleb который является активным участником php сообщества/и мы подтянем это – перезагрузим и теперь попробуем/откроем example test и я установлю курсор на название функции или где угодно внутри его и далее найдем phpunit run/отлично/как вы видите это динамически определило что должно быть сделано/в этом случае он определил что вы внутри этого метода и мы запустим только его/и это выполнилось/или если мы выйдем за его пределы или на имя класса и запустим его то вы теперь увидите что это выполнило только этот тест/давайте попробуем сделать что то очень простое например- функция и назовем тест загружает страницу about/и скажем/this get страницу about и assertSee about/действительно просто/и я могу это выполнить таким образом но это немного болезненно/вместо этого почему бы нам не назначить сокращение/перейдем в keybord shortcuts и посмотрим что сейчас назначено на наше расширение/ну да/cimmand k command r но что я обычно делаю это command t/но вы можете установить что хотите/t для теста/и теперь я запущу это/это провалилось/и еще одна прелестная вещь что вы можете нажать любую клавишу чтобы закруыть это/ну что/давайте попробуем отладить это/мы добавим новый маршрут тут/если кто то посещает страницу about 'то просто загрузит страницу about и теперь мы должны пройти тест/и да я могу вручную вернуться в этот файл и нажав command t получить зеленый свет или мы можем использовать команду - запустить предыдущий тест/давайте посмотрим better php unit run previous/nак почему ы нам не назначить простое сокращение для этого/ и изменим на ahift command t/ окей и теперь даже если я нахожусь за рамками тестируемого класса нажатие shift command t запустит тест все равно для меня/и это на самом деле очень удобный рабочий процесс/и попробуем еще несколько раз/и закроем/command t внутри метода – запустиля для этого метода /command t где угодно за рамками метода – запустит весь тест внутри файла и затем shift command t запустить предыдущий тест/я использую это бесчисленное количество раз каждый день/так что убедитесь что вы подтянули это/

/

Построенное [Caleb Porzio](https://github.com/calebporzio/better-phpunit) расширение "Better PHPUnit" предлагает нам невероятно чистый процесс для тестирования приложений.

/

[Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)The Context Menu Convenience

**[EPISODE 12](https://laracasts.com/series/visual-studio-code-for-php-developers/episodes/12)● RUN TIME 1:33**

Для тех кто предпочитает использовать мышь меньше тут есть один недостаток/предположим у нас есть это выражение где мы хотим сократить путь request/я мог бы просто кликнуть и выбрать это/или я вернусь обратно на секнду/да я все еще могу назначить горячую клавишу/но иногда для этих действий я просто предпочитаю открыть диалог и сделать это отсюда вместо того чтобы запоминать в этом случае 5-6 горячих клавиш/это просто тяжелее содержать в голове/но что мы могли бы сделать это назначить комбинацию для вызова этого контекстного меню/окей я собираюсь найти что то связанное с keybord и далее context и вот оно/показать редактор контекстного меню/и в настоящий момент это привязано к shift f10/давайте изменим это на alt enter/это пришло к нам из мира php storm и может быть более знакомо вам/окей и теперь адвайте взглянем/сейчас если я нажму alt enter это отобразит нам контекстное меню/окей и теперь я могу написать request input foo или что то вроде того и да/ если я хочу спуститься внутрь этого метода я просто нажимаю alt enter и давайте найдем определение этого метода/и вот тут показано что он делает/это такая мелкая вещь и не такое большое дело если вы используете мышь/но вот что я обнаружил в своем рабочем процессе – это чуточку быстрее и проще вместо использования мыши/может быть вам понравится тоже/

/

Мы становимся придирчивыми к этому моменту, но это нормально. Необходимость добраться до мыши, чтобы щелкнуть правой кнопкой мыши по символу, чтобы выполнить любой вид контекстного действия немного раздражает. Вместо этого давайте назначим комбинацию клавиш клавиатуры, чтобы мои руки могли оставаться на клавиатуре постоянно.

/

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)Faster Write-offs With Multiple Cursors

**EPISODE 14 • RUN TIME 3:01**

Следующий шаг/у нас есть несколько курсоров/представьте что для этих трех методов мы хотим установить свойство видимости как protected/конечно вы можете делать это один за одним но в этом нет нужды/вместо этого это отличный случай для использования нескольких курсоров и мы можем сделать это несколькими способами/первый – мы можем кликнуть где угодно мышкой или зажать альт и затем кликнуть в нескольких местах/и теперь вы видите что у нас есть несколько курсоров с помощью которых я изменяю это на protected/и это был способ с использованием мыши/мы могли бы зажать control z чтобы вернуть обратно/другая опция это использовать command D и между прочем это то же сокращение которое вы можете использовать в sublime/

**036 продолжить**

# [Visual Studio Code for PHP Developers:](https://laracasts.com/series/visual-studio-code-for-php-developers)Measure Your JavaScript Dependency Cost

**EPISODE 16 • RUN TIME 2:21**

Так у меня все еще открыт этот example project / перейдем в ресурсы активы js и взглянем на эти компоненты/например image upload /и что первое бросается в глаза – тут нет никакого форматирования/vs code yе распознает vue файлы/но нам повезло/тут есть подсказка установить Vetur расширение/сделаем это/и пока это устанавливается мы оглядимся тут немного/вот основные фишки/подсветка синтаксиса/сниппеты/эммет/ автоматическое форматирование и поддержка автозавершения/так давайте перезагрузимся и я нажму shift command P чтобы открыть панель команд/откроем image upload и теперь это выглядит намного лучше/я ищу импотр но такое ощущение что тут этого нет/давайте откроем reply vue вместо этого/и вот тут у нас есть 2 импорта/ moment js и компонент favorite/взгянем/я нажму shift command X и установим расширение import cost/'то очень полезно/давайте я пока вам что это делает/усановим обновим и посомтрим/если я опять открою reply vue я ожидаю увидеть стоимость каждого импорта но так как это vue файл я не увижу его в коробочной версии/взгляните на это/import cost и поглядим/мы должны видеть 2 опции/насколько я понял даже если я не использу typescript это хороший вариант и я его выберу/зажму command запятая и я вставю его прямо сюда и в нашем случае мы бы хотели распознавать vue файлы/окей/закроем это/вернемся обратно к reply vue и тут же мы увидим небольшую обратную связь/это немного текста который не сломает ваш код просто немного информации/и так если бы я писал этот код/я бы увидел – оу это действительно большая заисимость так тчо я должен убедиться что я оптимизировал это насколько возможно или может быть даже переключился на альтернативный вариант/вы увидите тут есть даже больше/давайте посмотрим на наш bootstrap файл/и да это сейчас расчитывается и теперь это показало нам что в зависимости от веса это будет иметь разную подсветку/так тчо да этот плагин действительно полезен когда вы работаете с фалом и хотите знать сколько весят зависимости/

/

Полезное расширение "Import Cost" позволяет нам автоматически визуализировать размер каждой импортируемой зависимости JavaScript. Он делает это, добавляя небольшую, ненавязчивую метку рядом с каждым импортом. Вы импортируете огромную зависимость, о которой не знали? Ну, если да, то это расширение покажет это кристально ясно.

/