

Welcome to knowledge-base

Pest

Kommandos

pest: composer test:pest-coverage

Siehe hier: pest.de

pest: Datei Testen

```
composer test:pest-coverage Tests\Pfad\Zur\Datei\DateiOhneEndung
```


php

Try catch finally

Beim Ausführen unserer Skripte treten Fehler und Ausnahmen auf, das gehört zur Natur der Sache. Jedoch ist es nötig, Fehler voneinander zu unterscheiden und entsprechend zu behandeln. Dafür können wir in PHP die Schlüsselwörter **try**, **catch** und **finally** verwenden.

1. Ein einfacher PHP try catch Block

Fehler und Ausnahmen können wir behandeln, wenn der Code, der diese produziert, in einem **try** Block geschrieben wird. Mit dem Schlüsselwort **catch** können wir diese dann abfangen.

```
<?php
try {
    # Tu was
} catch (Exception $e) {
    # Kümmere dich um Ausnahmen
}
```

1.1 Fehler und Ausnahmen

Ein Fehler tritt zum Beispiel auf, wenn du eine Funktion aufrufst, die nie definiert wurde.

```
<?php undefinierte_funktion(); ?>
```

Output:

```
Fatal error: Uncaught Error: Call to undefined function undefinierte\_funktion() in C:\xampp\htdocs\codecitrus\
\try\_catch.php:1 Stack trace: #0 {main} thrown in C:\xampp\htdocs\codecitrus\try\_catch.php on line 1
```

Um Code zu definieren, der nur ausgeführt wird, wenn ein Fehler (**Error**) auftritt, kannst du **try** und **catch** anwenden.

```
<?php
try {
    undefinierte_funktion();
    echo 'Es ist kein Fehler aufgetreten';
} catch (Error $e) {
    echo 'Ein Fehler ist aufgetreten' . '<br>';
    echo $e->getMessage();
}
```

Output:

```
Ein Fehler ist aufgetreten  
Call to undefined function undefinierte_funktion()
```

Wenn stattdessen eine Ausnahme (**Exception**) auftritt, musst du das im **catch** Block so angeben, um diese zu behandeln.

```
<?php  
function ausnahme() {  
    throw new Exception('Eine Ausnahme ist aufgetreten');  
}  
try {  
    ausnahme();  
} catch (Exception $e) {  
    echo $e->getMessage();  
}
```

Output:

```
Eine Ausnahme ist aufgetreten
```

1.2 Beide abfangen mit Throwable

Die vorigen Erklärungen haben sowohl Exceptions als auch Errors behandelt. Um beide abzufangen, könntest du zwei **catch** Blöcke hintereinander definieren.

```
<?php  
try {  
    undefinierte_funktion();  
} catch (Exception $e) {  
    echo 'Exception';  
} catch (Error $e) {  
    echo 'Error';  
}
```

Output:

```
Error
```

Eine weitere Möglichkeit ist, **Throwable** mit **catch** abzufangen. Throwable ist das Interface, das jede PHP-Klasse implementieren muss, soll sie mit **throw** aufgerufen werden. Für das nächste Beispiel wird die Funktion *ausnahme*, die wir vorhin definiert haben, erneut genutzt.

```
<?php  
try {  
    undefinierte_funktion();  
} catch (Throwable $t) {  
    echo 'Gotta catch \'em all!';  
}  
// Gotta catch 'em all!
```

```
try {
    ausnahme();
} catch (Throwable $t) {
    echo 'Schnapp sie dir alle!';
}
// Schnapp sie dir alle!
```

2. finally um den Block zu beenden

Mit dem Schlüsselwort **finally** kannst du deinem Block weiteren Code hinzufügen. **finally** ermöglicht es dir, Code zu schreiben, der ausgeführt wird egal ob ein Fehler auftritt oder nicht.

```
<?php
function finally_beispiel_1() {
    try {
        throw new Exception();
    } catch (Exception $e) {
        echo "Exception!" . '<br>';
    } finally {
        echo "Finally!";
    }
}
finally_beispiel_1();
```

Output:

```
Exception!
Finally!
```

2.1 Vorsicht mit return!

return kann bei Verwendung von **finally** zu unerwarteten Ergebnissen führen.

```
<?php
function finally_beispiel_2() {
    try {
        throw new Exception();
        return 1;
    } catch (Exception $e) {
        echo "Exception!" . '<br>';
        return 2;
    } finally {
        echo "Finally!" . '<br>';
        return 3;
    }
}
echo finally_beispiel_2();
```

Output:

```
Exception!  
Finally!  
3
```

Im vorigen Beispiel wurde der Wert von **return** aus dem **catch** Block einfach mit dem aus **finally** überschrieben. Dasselbe würde passieren, wenn **return** bereits einen Wert aus **try** empfangen hätte.

3. Fazit – PHP try & catch

Mit **try**, **catch** und **finally** stellt uns PHP mächtige Werkzeuge zur Verfügung, um Ausnahmefälle und Fehler zu behandeln. Damit können wir Logs erstellen, alternative Routinen ausführen oder einfach unser Skript stoppen. Vorsicht ist jedoch geboten, wenn wir unseren Code in solchen Blöcken mit **return** verwenden, da dies zu unerwarteten Ergebnissen führen kann.

- [Original Artikel von Patrick](#)

When does PHP call __destruct()?

In PHP,

`__construct()` is called while creating an object and `__destruct()` is called while the object is being removed from memory. Using this knowledge, we can create more fluent APIs as demonstrated by Freek Van der Herten in this short video.

Now let's see when PHP calls

`__destruct()` exactly.

An object is removed from memory if you explicitly remove it:

```
<?php  
$object = new Object();  
  
unset($object); // __destruct will be called immediately.  
  
$object = null; // __destruct will be called immediately.
```

It's also called when the scope where the object live is about to be terminated, for example at the end of a controller method:

```
<?php  
function store(Request $request)  
{  
    $object = new Object();  
  
    User::create(...);  
}
```

```
// __destruct will be called here.  
  
return view('welcome');  
}
```

Even if we're within a long running process, queued job for example, __destruct will be called before the end of the handle method:

```
<?php  
function handle()  
{  
    $object = new Object();  
  
    User::create(...);  
  
    // __destruct will be called here.  
}
```

It'll also be called when the script is being terminated:

```
<?php  
function handle()  
{  
    $object = new Object();  
  
    User::create(...);  
  
    // __destruct will be called here.  
  
    exit(1);  
}
```

- [Original article by Mohamed Said](#)

phpstan

Kommandos

phpstan: Der tägliche Wahnsinn

```
vendor/bin/phpstan clear-result-cache --memory-limit=5G  
vendor/bin/phpstan analyse --ansi --memory-limit=5G  
vendor/bin/phpstan --generate-baseline --memory-limit=5G
```


Shortcuts

Visual Studio Code

Shortcut	Beschreibung
Command + Shift + c	Öffnet Consolen-IDE Item und springt direkt zum Projekt
Command + Shift + e	Öffnet den Workspace-Explorer
Command + Rechts	Spring zum Ende der Zeile
Command + Links	Spring zum Anfang der Zeile
Command + Oben	Springt zur ersten Zeile
Command + Unten	Springt zur letzten Zeile
Command + Enter	Zeile nach unten einfügen
Command + L	Komplette Zeile markieren
Command + D	Wort auswählen - Wiederholen Sie die Auswahl anderer Vorkommen
Command + F	Wort markieren und es wird im Suchfenster angezeigt, Suchfenster wird selektiert
Command + E	Wort markieren und es wird im Suchfenster angezeigt, Markierung bleibt auf das Wor
Command + A	Alles selektieren
Command + C	Markierung kopieren
Command + X	Markierung ausschneiden
Command + V	Kopierten Zwischenspeicher einfügen

Shortcut	Beschreibung
Shift + ESC	Suchfenster schließen
Option + Oben	Zeile nach oben verschieben
Option + Unten	Zeile nach unten verschieben
Option + Shift + oben	Zeile nach oben kopieren
Option + Shift + unten	Zeile nach unten kopieren
Command + Option + Shift + Oben	Zeile nach oben markieren und cursor platzieren
Command + Option + Shift + rechts	Zeile nach rechts markieren und cursor platzieren
Command + Option + Shift + unten	Zeile nach unten markieren und cursor platzieren
Command + Option + Shift + links	Zeile nach links markieren und cursor platzieren
Option + Rechts	Cursor zum Wort-Ende
Option + Shift + Rechts	Markiert das nächste Wort nach rechts
Option + Links	Cursor zum Wort-Anfang
Option + Shift + Rechts	Markiert das nächste Wort nach links
Ctrl + A	Zeilen Anfang
Ctrl + E	Zeilen Ende
Ctrl + Shift + A	Vom Cursor alles bis zum Anfang markieren
Ctrl + Shift + E	Vom Cursor alles bis zum Ende markieren
Ctrl + Tab	Zwischen den offenen Dateien wechseln

Shortcut	Beschreibung
Ctrl + D	Löschen nach Rechts
Ctrl + H	Löschen nach Links
Tab	Markierten Text ein Tab Schritt nach rechts verschieben
Command + Tab	Markierten Text ein Tab Schritt nach lins verschieben
Option + H	Github History anzeigen
Option + Enter	Öffnet Workbench View
Command + DEL	Vom Curosr alles nach links löschen
Command + Backspace	Vom Curosr alles nach rechts löschen
Command + Ctrl + DEL	Vom Curosr das nächste WortPaar nach links löschen
Command + Ctrl + Backspace	Vom Curosr das nächste WortPaar nach rechts löschen
Command + K + C	Zeile auskommentieren
Command + K + U	Zeile einkommentieren
Command + Shift + A	Toggle Block-Kommentare / ein- und auskommentieren
Commando + Shift + /	Toggle Line-Kommentare / ein- und auskommentieren
Command + Shift + K	Zeile löschen
Command + Option + Control + /	Gehe zum Klammerende

Mac

Durch Drücken einer bestimmten Tastenkombination kannst du Aktionen ausführen, für die du normalerweise eine Maus, ein Trackpad oder ein anderes Eingabegerät benötigst.

Um einen Tastaturkurzbefehl zu verwenden, halte mindestens eine Sondertaste gedrückt, während du die letzte Taste des Tastaturkurzbefehls drückst. Beispielsweise hältst du zur Verwendung des Kurzbefehls "Befehlstaste-C" ("Kopieren") die Befehlstaste gedrückt, drückst dann die Taste "C" und lässt anschließend beide Tasten los. Mac-Menüs und -Tastaturen verwenden häufig Symbole für bestimmte Tasten, darunter auch die Sondertasten:

- Befehlstaste (oder Cmd-Taste) ⌘
- Umschalttaste ⇧
- Wahlstaste (oder Alt-Taste) ⌥
- Control-Taste (oder Ctrl-Taste) ^
- Feststelltaste ⌵
- Funktionstaste ![]

Auf Tastaturen, die für Windows-PCs entwickelt wurden, verwendest du die Alt-Taste anstelle der Wahlstaste und die Windows-Logo-Taste anstelle der Befehlstaste.

Einige Tasten auf manchen Apple-Tastaturen zeigen besondere Symbole und Funktionen, z. B. für die Display-Helligkeit, die Tastaturhelligkeit, und andere. Wenn deine Tastatur diese Funktionen nicht zur Verfügung stellt, kannst du einige davon reproduzieren, indem du eigene Tastaturkurzbefehle definierst. Um diese Tasten als F1, F2, F3 oder andere standardmäßige Funktionstasten zu verwenden, kombiniere sie mit der Fn-Taste.

Ausschneiden, Kopieren, Einsetzen und andere häufig verwendete Kurzbefehle

- **Befehlstaste-X:** Ausgewähltes Objekt aus dem Originaltext löschen und in die Zwischenablage kopieren.
- **Befehlstaste-C:** Ausgewähltes Objekt in die Zwischenablage kopieren. Dies funktioniert auch mit Dateien im Finder.
- **Befehlstaste-V:** Den Inhalt der Zwischenablage in das aktuelle Dokument oder die aktuelle App einfügen. Dies funktioniert auch mit Dateien im Finder.
- **Befehlstaste-Z:** Den vorherigen Befehl widerrufen. Du kannst anschließend zum Wiederholen den Tastaturkurzbefehl Umschalttaste-Befehlstaste-Z drücken, um den Widerrufen-Befehl umzukehren. In manchen Apps kannst du mehrere Befehle widerrufen und wiederholen.
- **Befehlstaste-A:** Alle Objekte auswählen.
- **Befehlstaste-F:** Elemente in einem Dokument suchen oder ein Suchfenster öffnen.

- **Befehlstaste-G:** Das nächste Auftreten des gefundenen Objekts suchen (vorwärts suchen). Um das vorherige Auftreten zu suchen (rückwärts suchen), drücke Umschalttaste-Befehlstaste-G.
- **Befehlstaste-H:** Die Fenster der vordersten App ausblenden. Um nur die vorderste App anzuzeigen und alle anderen Apps auszublenden, drücke Wahltaste-Befehlstaste-H.
- **Befehlstaste-M:** Das vorderste Fenster im Dock ablegen. Um alle Fenster der vordersten App im Dock abzulegen, drücke Wahltaste-Befehlstaste-M.
- **Befehlstaste-O:** Das ausgewählte Objekt oder ein Dialogfenster zur Auswahl einer Datei öffnen.
- **Befehlstaste-P:** Das aktuelle Dokument drucken.
- **Befehlstaste-S:** Das aktuelle Dokument sichern.
- **Befehlstaste-T:** Einen neuen Tab öffnen.
- **Befehlstaste-W:** Das vorderste Fenster schließen. Um alle Fenster der App zu schließen, drücke Wahltaste-Befehlstaste-W.
- **Wahltaste-Befehlstaste-Esc:** Beenden einer App erzwingen.
- **Befehlstaste-Leertaste:** Das Spotlight-Suchfeld ein- oder ausblenden. Um eine Spotlight-Suche aus einem Finder-Fenster durchzuführen, drücke Befehlstaste-Wahltaste-Leertaste. (Wenn du mehrere Eingabequellen verwendest, um in anderen Sprachen zu schreiben, ändern diese Kurzbefehle die Eingabequelle, anstatt Spotlight anzuzeigen. Hier erfährst du, wie du einen nicht eindeutigen Tastaturkurzbefehl änderst)
- **Ctrl-Befehlstaste-Leertaste:** Die Zeichenübersicht einblenden, in der du wählen kannst.
- **Ctrl-Befehlstaste-F:** App im Vollbildmodus verwenden, falls die App dies unterstützt.
- **Leertaste:** verwenden, um eine Vorschau des ausgewählten Objekts zu sehen.
- **Befehlstaste-Tabulatortaste:** Unter den geöffneten Apps zur zuletzt verwendeten wechseln.
- **Befehlstaste-Abwärtsakzent ('):** Zwischen den Fenstern der App wechseln, die du gerade verwendest. (Das Zeichen auf der zweiten Taste ist je nach Tastatur unterschiedlich. Die Taste befindet sich in der Regel über der Tabulatortaste und links von der Zahl 1.)
- **Umschalttaste-Befehlstaste-5:** In macOS Mojave ein Bildschirmfoto oder eine Bildschirmaufnahme erstellen. Verwende bei älteren macOS-Versionen Umschalttaste-Befehlstaste-3 oder Umschalttaste-Befehlstaste-4, um Bildschirmfotos aufzunehmen.
- **Umschalttaste-Befehlstaste-N:** Einen neuen Ordner im Finder anlegen.
- **Befehlstaste-Komma (,):** Einstellungen für die vorderste App öffnen.

Kurzbefehle für Ruhezustand, Abmelden und Ausschalten

Möglicherweise musst du einige dieser Kurzbefehle etwas länger gedrückt halten als andere. So kannst du vermeiden, sie versehentlich zu verwenden.

- **Ein-/Ausschalter:** Drücken, um deinen Mac einzuschalten oder den Ruhezustand zu beenden. Halte den Schalter 1,5 Sekunden lang gedrückt, um deinen Mac in den Ruhezustand zu versetzen.* Halte den Schalter länger gedrückt, um das Ausschalten des Mac zu erzwingen.
- **Wahltaste-Befehlstaste-Ein-/Ausschalter*** oder **Wahltaste-Befehlstaste-Medienauswurf taste:** Ruhezustand des Mac aktivieren.
- **Ctrl-Umschalttaste-Ein-/Ausschalter*** oder **Ctrl-Umschalttaste-Medienauswurf taste:** Displays in Ruhezustand versetzen.
- **Ctrl-Ein-/Ausschalter*** oder **Ctrl-Medienauswurf taste:** Ein Dialogfenster anzeigen, das dir die Wahlmöglichkeiten "Neustart", "Ruhezustand" oder Ausschalten" anbietet.
- **Ctrl-Befehlstaste-Ein-/Ausschalter:*** Neustart des Mac erzwingen, ohne zum Sichern von geöffneten oder nicht gesicherten Dokumenten aufzufordern.
- **Ctrl-Befehlstaste-Medienauswurf taste:** Alle Apps beenden und den Mac anschließend neu starten. Falls geöffnete Dokumente nicht gesicherte Änderungen enthalten, wirst du gefragt, ob du sie sichern möchtest.
- **Ctrl-Wahltaste-Befehlstaste-Ein-/Ausschalter*** oder **Ctrl-Wahltaste-Befehlstaste-Medienauswurf taste:** Alle Apps beenden und den Mac anschließend ausschalten. Falls geöffnete Dokumente nicht gesicherte Änderungen enthalten, wirst du gefragt, ob du sie sichern möchtest.
- **Ctrl-Befehlstaste-Q:** Den Bildschirm sofort sperren.
- **Umschalttaste-Befehlstaste-Q:** Von deinem macOS-Benutzeraccount abmelden. Du wirst zur Bestätigung des Vorgangs aufgefordert. Um dich sofort ohne Bestätigung abzumelden, drücke Wahl taste-Umschalttaste-Befehlstaste-Q.

* Gilt nicht für den **Touch ID-Sensor**.

Finder- und Systemkurzbefehle

- **Befehlstaste-D:** Ausgewählte Dateien duplizieren.
- **Befehlstaste-E:** Ausgewähltes Laufwerk oder Volume auswerfen.
- **Befehlstaste-F:** Eine Spotlight-Suche im Finder-Fenster starten.
- **Befehlstaste-I:** Fenster "Informationen" für eine markierte Datei anzeigen.
- **Befehlstaste-R:** (1) Wenn im Finder ein Alias ausgewählt ist: ursprüngliche Datei für ausgewählten Alias anzeigen. (2) In einigen Apps wie Kalender oder Safari die Seite aktualisieren oder neu laden. (3) Unter Softwareupdate nochmals nach Softwareupdates suchen.
- **Umschalttaste-Befehlstaste-C:** Fenster "Computer" öffnen.

- **Umschalttaste-Befehlstaste-D**: Ordner "Schreibtisch" öffnen.
- **Umschalttaste-Befehlstaste-F**: Fenster "Zuletzt benutzt" öffnen, das alle kürzlich angesehenen oder geänderten Dateien auflistet.
- **Umschalttaste-Befehlstaste-G**: Ein Fenster "Gehe zum Ordner" öffnen.
- **Umschalttaste-Befehlstaste-H**: Benutzerordner des aktuellen macOS-Benutzeraccounts öffnen.
- **Umschalttaste-Befehlstaste-I**: iCloud Drive öffnen.
- **Umschalttaste-Befehlstaste-K**: Fenster "Netzwerk" öffnen.
- **Wahltaste-Befehlstaste-L**: Ordner "Downloads" öffnen.
- **Umschalttaste-Befehlstaste-N**: Neuen Ordner erstellen.
- **Umschalttaste-Befehlstaste-O**: Ordner "Dokumente" öffnen.
- **Umschalttaste-Befehlstaste-P**: Vorschaufenster in Finder-Fenstern ein- oder ausblenden.
- **Umschalttaste-Befehlstaste-R**: Fenster "AirDrop" öffnen.
- **Umschalttaste-Befehlstaste-T**: Tableiste in Finder-Fenstern ein- oder ausblenden.
- **Ctrl-Umschalttaste-Befehlstaste-T**: Das ausgewählte Finder-Objekt dem Dock hinzufügen (OS X Mavericks oder neuer)
- **Umschalttaste-Befehlstaste-U**: Ordner "Dienstprogramme" öffnen.
- **Wahltaste-Befehlstaste-D**: Dock ein- oder ausblenden.
- **Ctrl-Befehlstaste-T**: Das ausgewählte Objekt der Seitenleiste hinzufügen (OS X Mavericks oder neuer).
- **Wahltaste-Befehlstaste-P**: Pfadleiste in Finder-Fenstern ein- oder ausblenden.
- **Wahltaste-Befehlstaste-S**: Seitenleiste in Finder-Fenstern ein- oder ausblenden.
- **Befehlstaste-Schrägstrich (/)**: Statusleiste in Finder-Fenstern ein- oder ausblenden.
- **Befehlstaste-J**: Darstellungsoptionen einblenden.
- **Befehlstaste-K**: Das Fenster "Mit Server verbinden" öffnen.
- **Ctrl-Befehlstaste-A**: Erstelle einen Alias für das ausgewählte Element.
- **Befehlstaste-N**: Ein neues Finder-Fenster öffnen.
- **Wahltaste-Befehlstaste-N**: Neuen intelligenten Ordner erstellen.
- **Befehlstaste-T**: Tableiste ein- oder ausblenden, wenn nur ein Tab im aktuellen Finder geöffnet ist.
- **Wahltaste-Befehlstaste-T**: Symbolleiste ein- oder ausblenden, wenn nur ein Tab im aktuellen Finder-Fenster geöffnet ist.
- **Wahltaste-Befehlstaste-V**: Dateien in der Zwischenablage von ihrem ursprünglichen zum aktuellen Speicherort bewegen.
- **Befehlstaste-Y**: Die Funktion "Übersicht" zum Anzeigen einer Vorschau der ausgewählten Dateien nutzen.

- **Wahltaste-Befehlstaste-Y**: In der Übersicht eine Diashow der ausgewählten Dateien anzeigen.
- **Befehlstaste-1**: Die Objekte im Finder-Fenster als Symbole anzeigen.
- **Befehlstaste-2**: Die Objekte im Finder-Fenster als Liste anzeigen.
- **Befehlstaste-3**: Die Objekte im Finder-Fenster in Spalten anzeigen.
- **Befehlstaste-4**: Die Objekte im Finder-Fenster in einer Galerie anzeigen.
- **Befehlstaste-linker eckige Klammer (⌈)**: Zum vorherigen Ordner wechseln.
- **Befehlstaste-rechter eckige Klammer (⌋)**: Zum nächsten Ordner wechseln.
- **Befehlstaste-Aufwärtspfeil**: Ordner mit dem aktuellen Ordner öffnen.
- **Befehlstaste-Ctrl-Aufwärtspfeil**: Ordner mit dem aktuellen Ordner in einem neuen Fenster öffnen.
- **Befehlstaste-Abwärtspfeil**: Ausgewähltes Objekt öffnen.
- **Rechtspfeil**: Ausgewählten Ordner öffnen. Diese Funktion ist nur in der Listendarstellung verfügbar.
- **Linkspfeil**: Ausgewählten Ordner schließen. Diese Funktion ist nur in der Listendarstellung verfügbar.
- **Befehlstaste-Rückschritttaste**: Das markierte Objekt in den Papierkorb verschieben.
- **Umschalttaste-Befehlstaste-Rückschritttaste**: Den Papierkorb leeren.
- **Wahltaste-Umschalttaste-Befehlstaste-Rückschritttaste**: Den Papierkorb ohne Bestätigungsdiallog leeren.
- **Befehlstaste-Helligkeit verringern**: Bildschirmsynchronisation ein- bzw. ausschalten, wenn der Mac mit mehr als einem Monitor verbunden ist.
- **Wahltaste-Helligkeit erhöhen**: Systemeinstellung "Monitore" öffnen. Dies funktioniert mit beiden Helligkeitstasten.
- **Ctrl-Helligkeit erhöhen** oder **Ctrl-Helligkeit verringern**: Helligkeit des externen Displays ändern (falls vom Display unterstützt).
- **Wahltaste-Umschalttaste-Helligkeit erhöhen** oder **Wahltaste-Umschalttaste-Helligkeit verringern**: Display-Helligkeit in kleineren Schritten ändern. Füge die Ctrl-Taste hinzu, um die Anpassung am externen Display vorzunehmen, sofern das Display dies unterstützt.
- **Wahltaste-Mission Control**: Systemeinstellung "Mission Control" öffnen.
- **Befehlstaste-Mission Control**: Den Schreibtisch anzeigen.
- **Ctrl-Abwärtspfeil**: Alle Fenster der vordersten App einblenden.
- **Wahltaste-Lauter**: Systemeinstellung "Ton" öffnen. Dies funktioniert mit allen Lautstärketasten.
- **Wahltaste-Umschalttaste-Lauter** oder **Wahltaste-Umschalttaste-Leiser**: Lautstärke in kleineren Schritten regeln.
- **Wahltaste-Tastaturhelligkeit erhöhen**: Systemeinstellung "Tastatur" öffnen. Dies funktioniert mit beiden Tastaturhelligkeitstasten.

- **Wahltaste-Umschalttaste-Tastaturhelligkeit erhöhen** oder **Wahltaste-Umschalttaste-Tastaturhelligkeit verringern**: Tastaturhelligkeit in kleineren Schritten anpassen.
- **Wahltaste-Doppelklick**: Einen Ordner in einem neuen Fenster öffnen und das aktuelle Fenster schließen.
- **Befehlstaste-Doppelklick**: Einen Ordner in einem neuen Tab oder Fenster öffnen.
- **Befehlstaste beim Ziehen auf ein anderes Volume**: Gezogenes Objekt auf ein anderes Volume verschieben, anstatt es zu kopieren.
- **Wahltaste beim Ziehen**: Gezogenes Objekt kopieren. Der Mauszeiger ändert sich, während du das Objekt ziehst.
- **Wahltaste-Befehlstaste beim Ziehen**: Alias des gezogenen Objekts erstellen. Der Mauszeiger ändert sich, während du das Objekt ziehst.
- **Wahltaste-Klick auf ein Erweiterungsdreieck**: Alle Ordner im ausgewählten Ordner öffnen. Diese Funktion ist nur in der Listendarstellung verfügbar.
- **Befehlstaste-Klick auf Fenstertitel**: Ordner anzeigen, die den aktuellen Ordner enthalten.
- Klicke in der Finder-Menüleiste auf das Menü "Gehe zu", um Kurzbefehle zum Öffnen vieler häufig benutzter Ordner zu sehen, z. B. "Programme", "Dokumente", "Downloads", "Dienstprogramme" und "iCloud Drive".

Kurzbefehle für Dokumente

Das Verhalten dieser Kurzbefehle kann je nach verwendeter App variieren.

- **Befehlstaste-B**: Den ausgewählten Text fett formatieren bzw. Fettschrift ein- oder ausschalten.
- **Befehlstaste-I**: Den ausgewählten Text kursiv formatieren bzw. Kursivschrift ein- oder ausschalten.
- **Befehlstaste-K**: Einen Weblink hinzufügen.
- **Befehlstaste-U**: Den ausgewählten Text unterstrichen formatieren bzw. Unterstreichung ein- oder ausschalten.
- **Befehlstaste-T**: Das Fenster "Schriften" ein- oder ausblenden.
- **Befehlstaste-D**: Den Ordner "Schreibtisch" in den Dialogfenstern "Öffnen" oder "Sichern" auswählen.
- **Ctrl-Befehlstaste-D**: Die Definition eines ausgewählten Worts ein- oder ausblenden.
- **Umschalttaste-Befehlstaste-Doppelpunkt (:)**: Fenster "Rechtschreibung und Grammatik" einblenden.
- **Befehlstaste-Semikolon (;)**: Falsch geschriebene Wörter im Dokument suchen.
- **Wahltaste-Rückschritttaste**: Das Wort links von der Einfügemarke löschen.
- **Ctrl-H**: Das Zeichen links von der Einfügemarke löschen. Verwende alternativ die Rückschritttaste.

- **Ctrl-D:** Das Zeichen rechts von der Einfügemarke löschen. Verwende alternativ Fn-Rückschritttaste.
- **Fn-Rückschritttaste:** Vorwärts löschen auf Tastaturen, die keine Entf-Taste haben. Verwende alternativ Ctrl-D.
- **Ctrl-K:** Text zwischen der Einfügemarke und dem Ende der aktuellen Zeile oder des Absatzes löschen.
- **Fn-Aufwärtspfeil:** Seite nach oben: Eine Seite nach oben blättern.
- **Fn-Abwärtspfeil:** Seite nach unten: Eine Seite nach unten blättern.
- **Fn-Linkspfeil:** Pos1: Zum Anfang eines Dokuments blättern.
- **Fn-Rechtspfeil:** Ende: Zum Ende eines Dokuments blättern.
- **Befehlstaste-Aufwärtspfeil:** Einfügemarke an den Anfang des Dokuments bewegen.
- **Befehlstaste-Abwärtspfeil:** Einfügemarke an das Ende des Dokuments bewegen.
- **Befehlstaste-Linkspfeil:** Einfügemarke an den Anfang der aktuellen Zeile bewegen.
- **Befehlstaste-Rechtspfeil:** Einfügemarke an das Ende der aktuellen Zeile bewegen.
- **Wahltaste-Linkspfeil:** Einfügemarke an den Anfang des vorhergehenden Worts bewegen.
- **Wahltaste-Rechtspfeil:** Einfügemarke an das Ende des nächsten Worts bewegen.
- **Umschalttaste-Befehlstaste-Aufwärtspfeil:** Text zwischen der Einfügemarke und dem Anfang des Dokuments markieren.
- **Umschalttaste-Befehlstaste-Abwärtspfeil:** Text zwischen der Einfügemarke und dem Ende des Dokuments markieren.
- **Umschalttaste-Befehlstaste-Linkspfeil:** Text zwischen der Einfügemarke und dem Anfang der aktuellen Zeile markieren.
- **Umschalttaste-Befehlstaste-Rechtspfeil:** Text zwischen der Einfügemarke und dem Ende der aktuellen Zeile markieren.
- **Umschalttaste-Aufwärtspfeil:** Textauswahl auf das der aktuellen Position am nächsten liegende Zeichen in der darüberliegenden Zeile erweitern.
- **Umschalttaste-Abwärtspfeil:** Textauswahl auf das der aktuellen Position am nächsten liegende Zeichen in der darunterliegenden Zeile erweitern.
- **Umschalttaste-Linkspfeil:** Textauswahl um ein Zeichen nach links erweitern.
- **Umschalttaste-Rechtspfeil:** Textauswahl um ein Zeichen nach rechts erweitern.
- **Wahltaste-Umschalttaste-Aufwärtspfeil:** Textauswahl bis zum Anfang des aktuellen Absatzes erweitern, bei nochmaligem Drücken bis zum Anfang des nächsten Absatzes.
- **Wahltaste-Umschalttaste-Abwärtspfeil:** Textauswahl bis zum Ende des aktuellen Absatzes erweitern, bei nochmaligem Drücken bis zum Ende des nächsten Absatzes.

- **Wahltaste-Umschalttaste-Linkspfeil:** Textauswahl bis zum Anfang des aktuellen Worts erweitern, bei nochmaligem Drücken bis zum Anfang des nächsten Worts.
- **Wahltaste-Umschalttaste-Rechtspfeil:** Textauswahl bis zum Ende des aktuellen Worts erweitern, bei nochmaligem Drücken bis zum Ende des nächsten Worts.
- **Ctrl-A:** Zum Anfang der Zeile oder des Absatzes bewegen.
- **Ctrl-E:** Zum Ende einer Zeile oder eines Absatzes bewegen.
- **Ctrl-F:** Ein Zeichen vor bewegen.
- **Ctrl-B:** Ein Zeichen zurück bewegen.
- **Ctrl-L:** Cursor oder Auswahl im sichtbaren Bereich zentrieren.
- **Ctrl-P:** Eine Zeile nach oben bewegen
- **Ctrl-N:** Eine Zeile nach unten bewegen.
- **Ctrl-O:** Neue Zeile hinter der Einfügemarke einfügen.
- **Ctrl-T:** Das Zeichen hinter der Einfügemarke mit dem Zeichen davor austauschen.
- **Befehlstaste-linke geschweifte Klammer ({}):** Linksbündig.
- **Befehlstaste-rechte geschweifte Klammer ({}):** Rechtsbündig.
- **Umschalttaste-Befehlstaste-Senkrechtstrich (|):** Zentrieren.
- **Wahltaste-Befehlstaste-F:** Zum Suchfeld wechseln.
- **Wahltaste-Befehlstaste-T:** Symbolleiste in der App ein- oder ausblenden.
- **Wahltaste-Befehlstaste-C:** Stil kopieren: Formatierungseinstellungen des ausgewählten Objekts in die Zwischenablage kopieren.
- **Wahltaste-Befehlstaste-V:** Stil einsetzen: Die kopierten Formatierungseinstellungen auf das ausgewählte Objekt anwenden.
- **Wahltaste-Umschalttaste-Befehlstaste-V:** Einsetzen und Stil anpassen: Den Stil des umgebenden Inhalts auf das darin eingesetzte Objekt anwenden.
- **Wahltaste-Befehlstaste-I:** Das Fenster "Informationen" ein- oder ausblenden.
- **Umschalttaste-Befehlstaste-P:** Papierformat: Ein Fenster zur Auswahl von Dokumenteinstellungen anzeigen.
- **Umschalttaste-Befehlstaste-S:** Dialogfenster "Sichern unter" anzeigen oder aktuelles Dokument duplizieren.
- **Umschalttaste-Befehlstaste-Minus-Zeichen (-):** Ausgewähltes Objekt verkleinern.
- **Umschalttaste-Befehlstaste-Plus-Zeichen (+):** Ausgewähltes Objekt vergrößern. Befehlstaste-Gleichheitszeichen (=) führt dieselbe Funktion aus.
- **Umschalttaste-Befehlstaste-Fragezeichen (?):** Hilfe-Menü öffnen.

Valet

Kommandos

Valet: Erstellt V-Hosts Links: <http://www.meine-lokale-webseite.test> und <https://www.meine-lokale-webseite.test>

```
valet link --secure meine-lokale-webseite
```

Valet: Löscht die V-Hosts Links: <http://www.meine-lokale-webseite.test> und <https://www.meine-lokale-webseite.test>

```
valet unlink --secure meine-lokale-webseite
```

Valet: Alle registrierten Valet-Links anzeigen

```
valet links
```

Valet: Listet alle verfügbaren Valet Kommandos auf

```
valet -h  
valet --help
```


Plugin

[Auto Close Tag](#)

[Better Comments](#)

[Better Pest](#)

[DotENV](#)

[Dummy Text Generator](#)

[EditorConfig for VS Code](#)

[Filament Snippets](#)

[Git Blame](#)

[Git History](#)

[GitHub Pull Requests and Issues](#)

[GitHub Theme](#)

[gitignore](#)

[Image preview](#)

[Laravel Artisan](#)

[Laravel Blade formatter](#)

[Laravel Blade Snippets](#)

[Laravel Create View](#)

[Laravel Extension Pack](#)

[Laravel goto view](#)

Plugin

laravel intellisense

Laravel Pint Formatter

Laravel Snippets

laravel-goto-components

Livewire Language Support

Markdown All in One

Markdown Preview Mermaid Support

Mermaid Markdown Syntax Highlighting

Mermaid Preview

MSSQL Snippets

Output Colorizer

Pest Snippets

PHP Intelephense

PHP IntelliSense

Rainbow Brackets 2

Remote - SSH

Remote - SSH: Editing Configuration Files

SVG

TODO Highlight

Plugin

[vscode-icons](#)

[Sort lines](#)

Pest

composer.json

```
{
  "require-dev": {
    "pestphp/pest": "^1.22",
    "pestphp/pest-plugin-laravel": "^1.3",
    "pestphp/pest-plugin-livewire": "^1.0",
    "phpstan/extension-installer": "^1.1",
    "phpstan/phpstan-deprecation-rules": "^1.0",
    "phpstan/phpstan-phpunit": "^1.0",
    "phpunit/php-code-coverage": "^9.2",
    "phpunit/phpunit": "^9.5"
  },
  "scripts": {
    "test:pest": "vendor/bin/pest --order-by default -d memory_limit=6144M",
    "test:pest-coverage": "php -dpcov.enabled=1 -dpcov.directory=. -dpcov.exclude='~vendor~' vendor/bin/pest -d memory_limit=6144M --testdox --verbose --coverage --min=85",
    "test:unit": "vendor/bin/testbench package:test --no-coverage",
    "test:types": "vendor/bin/phpstan analyse",
    "test": [
      "@lint:fix",
      "@test:types",
      "@test:unit"
    ]
  }
}
```

Git

Force push git branches

```
git push --force origin main  
git push -f origin main  
git push origin +main
```

- [Original tweet by Stefan Judis](#)

Github key anpassen bzw. bereinigen

```
composer update --lock
```

```
composer update --lock s
```


Issues verlinken

Sie können einen Pull-Request oder Branch mit einem Issue verknüpfen, um anzuzeigen, dass eine Behebung in Bearbeitung ist, und um das Issue automatisch zu schließen, wenn der Pull-Request oder Branch zusammengeführt wird.

Hinweis: Die speziellen Schlüsselwörter in einer Pull-Request-Beschreibung werden interpretiert, wenn die Pull-Request auf den *Default* - Branch des Repositorys abzielt. Wenn die Basis des PR jedoch *ein anderer Zweig* ist, werden diese Schlüsselwörter ignoriert, es werden keine Links erstellt und das Zusammenführen des PR hat keine Auswirkung auf die Ausgaben. **Wenn Sie einen Pull-Request über ein Schlüsselwort mit einem Issue verknüpfen möchten, muss sich der PR auf dem Standard-Branch befinden.**

Über verknüpfte Probleme und Pull-Requests

Sie können ein Problem manuell mit einer Pull-Anforderung verknüpfen oder ein unterstütztes Schlüsselwort in der Beschreibung der Pull-Anforderung verwenden.

Wenn Sie eine Pull-Anforderung mit dem Problem verknüpfen, auf das sich die Pull-Anforderung bezieht, können Mitarbeiter sehen, dass jemand an dem Problem arbeitet.

Wenn Sie einen verknüpften Pull-Request mit dem Standard-Branch eines Repositorys zusammenführen, wird das verknüpfte Issue automatisch geschlossen. Weitere Informationen zum Standard-Zweig finden Sie unter „[Ändern des Standard-Zweigs](#)“.

Verknüpfen einer Pull-Anforderung mit einem Problem mithilfe eines Schlüsselworts

Sie können eine Pull-Anforderung mit einem Problem verknüpfen, indem Sie ein unterstütztes Schlüsselwort in der Beschreibung der Pull-Anforderung oder in einer Commit-Nachricht verwenden. Die Pull-Anfrage **muss sich** im Standard-Branch befinden.

- nah dran
- schließt
- abgeschlossen
- Fix
- behebt
- Fest

- beschließen
- beschließt
- aufgelöst

Wenn Sie ein Schlüsselwort verwenden, um auf einen Pull-Request-Kommentar in einem anderen Pull-Request zu verweisen, werden die Pull-Requests verknüpft. Durch das Zusammenführen der referenzierenden Pull-Anforderung wird auch die referenzierte Pull-Anforderung geschlossen.

Die Syntax zum Schließen von Schlüsselwörtern hängt davon ab, ob sich das Problem im selben Repository wie die Pull-Anfrage befindet.

Verknüpftes Problem	Syntax	Beispiel
Ausgabe im selben Repository	<i>SCHLÜSSELWORT # AUSGABE - NUMMER</i>	Closes #10
Ausgabe in einem anderen Repository	<i>KEYWORD OWNER / REPOSITORY # ISSUE-NUMBER</i>	Fixes octo-org/octo-repo#100
Mehrere Probleme	Verwenden Sie für jedes Problem die vollständige Syntax	Resolves #10, resolves #123, resolves octo-org/octo-repo#100

Nur manuell verknüpfte Pull-Requests können manuell entkoppelt werden. Um die Verknüpfung eines Problems aufzuheben, das Sie mit einem Schlüsselwort verknüpft haben, müssen Sie die Pull-Request-Beschreibung bearbeiten, um das Schlüsselwort zu entfernen.

Sie können auch schließende Schlüsselwörter in einer Commit-Nachricht verwenden. Das Problem wird geschlossen, wenn Sie den Commit in den Standard-Branch zusammenführen, aber die Pull-Anforderung, die den Commit enthält, wird nicht als verknüpfte Pull-Anforderung aufgeführt.

Jeder mit Schreibberechtigungen für ein Repository kann eine Pull-Anforderung manuell mit einem Vorgang über die Seitenleiste für Pull-Anforderungen verknüpfen.

Sie können bis zu zehn Issues manuell mit jeder Pull-Anfrage verknüpfen. Das Issue und die Pull-Anforderung müssen sich im selben Repository befinden.

1. Navigieren Sie auf GitHub.com zur Hauptseite des Repositorys.
2. Klicken Sie unter Ihrem Repository-Namen auf **Pull-Anforderungen**.

[Auswahl der Registerkarte „Probleme und Pull-Requests“](#).

3. Klicken Sie in der Liste der Pull-Requests auf den Pull-Request, den Sie mit einem Issue verknüpfen möchten.
4. Klicken Sie in der rechten Seitenleiste im Abschnitt "Entwicklung" auf .

5. Klicken Sie auf das Problem, das Sie mit der Pull-Anforderung verknüpfen möchten. [Drop-down zum Link-Problem](#)

Jeder mit Schreibberechtigungen für ein Repository kann eine Pull-Anfrage manuell verknüpfen oder von der Issue-Seitenleiste zu einem Issue verzweigen.

Sie können bis zu zehn Issues manuell mit jeder Pull-Anfrage verknüpfen. Das Problem kann sich in einem anderen Repository befinden als der verknüpfte Pull-Request oder Branch. Ihr zuletzt ausgewähltes Repository wird gespeichert

1. Navigieren Sie auf GitHub.com zur Hauptseite des Repositorys.
2. Klicken Sie unter Ihrem Repository-Namen auf **Issues** .
[Registerkarte "Probleme"](#).
3. Klicken Sie in der Liste der Issues auf das Issue, mit dem Sie eine Pull-Anfrage oder einen Branch verknüpfen möchten.
4. Klicken Sie in der rechten Seitenleiste auf **Entwicklung** . [Entwicklungsmenü in der rechten Seitenleiste](#)
5. Klicken Sie auf das Repository mit der Pull-Anfrage oder dem Zweig, den Sie mit dem Problem verknüpfen möchten. [Drop-down, um das Repository auszuwählen](#)
6. Klicken Sie auf den Pull-Request oder Branch, den Sie mit dem Issue verknüpfen möchten. [Drop-down, um Pull-Request oder Branch zu verknüpfen](#)
7. Klicken Sie auf **Anwenden** . [Anwenden](#)

Weiterlesen

- ["Automatisch verlinkte Verweise und URLs"](#)

Arr

Throw an exception

```
<?php

$config = ['.....'];
$key = Arr::get($config, 'api_key', fn () => throw new Exception('your message here'));

// or
$key = $config['api_key'] ?? throw new Exception('your message here');
```

- [Original tweet by Steve Bauman](#)

Collection

sort

```
<?php

$exampleEntries = [
    'my example exa' => 'value_4',
    'my example' => 'value_1',
    'my example ex' => 'value_3',
    'my example e' => 'value_2',
];

$result = collect($exampleEntries)->sort()->toArray();

$result = [
    "my example" => "value_1",
    "my example e" => "value_2",
    "my example ex" => "value_3",
    "my example exa" => "value_4",
]
```

sortKeys

```
<?php

$exampleEntries = [
    'key_1' => 'value_1',
    'key_3' => 'value_3',
    'key_2' => 'value_2',
    'key_7' => 'value_7',
    'key_4' => 'value_4',
    'key_6' => 'value_6',
    'key_5' => 'value_5',
];

$result = collect($exampleEntries)->sortKeys()->toArray();

$result = [
    "key_1" => "value_1",
    "key_2" => "value_2",
    "key_3" => "value_3",
    "key_4" => "value_4",
    "key_5" => "value_5",
    "key_6" => "value_6",
    "key_7" => "value_7",
]
```

Eloquent

Touch eloquent method

```
<?php

$user = User::find(1);

// instead of
$user->update(['subscribed_at' => now()]);

// use
$user->touch('subscribed_at');
```

- [Original tweet by Oussama Sid](#)

Migrations

Unique columns

Laravel Example 1:

```
<?php

DB::update("
    ALTER TABLE tournament_league_game_days
    ADD COLUMN game_schedule_day_unique varchar (512)
    GENERATED ALWAYS AS
    (
        CONCAT(
            CONCAT(day, '#', game_schedule_id),
            '#',
            IF(deleted_at IS NULL, '-', deleted_at)
        )
    ) VIRTUAL;

");

DB::update("
    CREATE UNIQUE INDEX game_schedule_day_unique ON tournament_league_game_days
    (game_schedule_day_unique);

");
```

Laravel Example 2:

```
<?php

Schema::table('tournament_league_game_days', function (Blueprint $table) {
    $table->string('game_schedule_day_unique')
        ->virtualAs(
            DB::raw(
                "CONCAT(
                    CONCAT(day, '#', game_schedule_id),
                    '#',
                    IF(deleted_at IS NULL, '-', deleted_at)
                )"
            )
        );
});

Schema::table('tournament_league_game_days', function (Blueprint $table) {
    $table->unique(['game_schedule_day_unique', 'game_schedule_day_unique_index']);
});
```

- [Original tweet by Tobias_Petry.sql](#)

Route

Test-Route

```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/test-route', function () {
    // return your example
})
```

withoutMiddleware

```
<?php

use App\Http\Middleware\FirstMiddleware;
use App\Http\Middleware\SecondMiddleware;
use Illuminate\Support\Facades\Route;

Route::prefix('my-prefix')
->middleware([
    FirstMiddleware::class,
    SecondMiddleware::class,
])
->group(function () {
    Route::get('my-route', function () {
        // Uses first & second middleware
    });
    Route::withoutMiddleware(FirstMiddleware::class)
        ->get('route-without-first-middleware', function () {
            // Uses second middleware
        });
    Route::withoutMiddleware(SecondMiddleware::class)
        ->get('route-without-second-middleware', function () {
            // Uses first middleware
        });
    Route::withoutMiddleware([
        FirstMiddleware::class,
        SecondMiddleware::class,
    ])
        ->get('route-without-middleware', function () {
            // Uses no middleware
        })
    );
});
```


with Parameters

```
<?php

namespace App\Http\Middleware;

use Closure;

class EnsureUserHasRole
{
    public function handle($request, Closure $next, $role)
    {
        if (! $request->user()->hasRole($role)) {
            // Redirect...
        }

        return $next($request);
    }
}
```

```
<?php

use Illuminate\Support\Facades\Route;

Route::put('/post/{id}', function ($id) {
    //
})->middleware('role:editor');
```


Tests

expectException and expectExceptionMessage

```
<?php

use Application\User\Queries\ListUserQuery;
use Illuminate\Http\Request;
use Spatie\QueryBuilder\Exceptions\InvalidFilterQuery;
use Tests\TestCase;

class ListUserQueryTest extends TestCase
{
    /** @test */
    public function it_throws_an_exception_when_the_key_for_filtering_is_not_supported(): void
    {
        $this->expectException(InvalidFilterQuery::class);
        $this->expectExceptionMessage('Requested filter(s) `key_not_supported` are not allowed. Allowed filter(s) are `id, email, nickname`.');

        $request = new Request(['filter' => ['key_not_supported' => 'value is irrelevant']]);

        new ListMyModelQuery($request);
    }
}
```

it_uses_the_right_query_filters

```
<?php

/** @test */
public function it_uses_the_right_query_filters(): void
{
    $this->assertQueryFilterEquals(
        ListMyModelQuery::class,
        [
            AllowedFilter::exact('id'),
            AllowedFilter::exact('handle'),
            AllowedFilter::partial('name', 'display_name'),
        ]
    );
}
```

assertThrows | it_throws_an_error_if_model_doesnt_exist

```
<?php
```

```

/** @test */
public function it_throws_an_error_if_model_doesnt_exist(): void
{
    $className = User::class;

    $this->assertThrows(
        fn () => User::findOrFail(0),
        ModelNotFoundException::class,
        "No query results for model [{$className}] 1"
    );
}

```

it_uses_the_right_query_class

```

<?php

/** @test */
public function it_uses_the_right_query_class(): void
{
    $query = $this->mock(
        ListMyModelQuery::class,
        fn (MockInterface $mock) => $mock->shouldReceive('simplePaginate')
            ->once()
            ->andReturn(MyModelFactory::new()->create()->simplePaginate()),
    );

    $controller = new ViewMyModelListController();
    $controller($query, new Request());

    $this->assertControllerUsesClass(
        ViewMyModelListController::class,
        ListMyModelQuery::class
    );
}

```

it_uses_the_right_collection

```

<?php

/** @test */
public function it_uses_the_right_collection(): void
{
    $this->assertControllerReturns(
        ViewMyModelListController::class,
        MyModelCollection::class
    );
}

```

it_uses_the_right_middleware

```
<?php

public function middlewares(): array
{
    return [
        'group1' => ['group1_sub1', 'group1_sub2'],
        'group2' => ['group2_sub1', 'group2_sub2'],
        MyMiddleware::class => [MyMiddleware::class],
    ];
}

/**
 * @test
 *
 * @dataProvider middlewares
 */
public function it_uses_the_right_middleware($middleware)
{
    $this->assertControllerUsesMiddleware(
        ViewMyModelListController::class,
        $middleware,
    );
}
```

it_returns_the_right_structure | Collection

```
<?php

/** @test */
public function it_returns_the_right_structure(): void
{
    MyModelFactory::new()->create();

    $response = $this->createResource(MyModelCollection::class, MyModel::simplePaginate());

    $response->assertJsonStructureExact([
        'data',
        'links',
        'meta',
    ]);

    $this->assertEquals(MyModelResource::class, MyModelCollection::make([])->collects);
}
```

it_returns_the_right_structure | Resource

```
<?php
```

```

/** @test */
public function it_returns_the_right_structure(): void
{
    $response = $this->createResource(
        MyModelResource::class,
        MyModelFactory::new()->create()
    );

    $response->assertJsonStructureExact([
        'id',
        'my_column_one',
        'my_column_two',
        'created_at',
        'updated_at',
    ]);

    $response->assertJson(
        fn (AsserttableJson $json) => $json
            ->whereAllType([
                'id' => ['integer'],
                'my_column_one' => ['string'],
                'my_column_two' => [null, 'string'],
                'created_at' => ['string'],
                'updated_at' => ['string'],
            ])
            ->etc()
    );
}

```

examples_for_mock_actions

```

<?php

/** @test */
public function examples_for_mock_actions(): void
{
    $myAction = $this->spy(MyActionnn::class);
    $myAction->shouldReceive('execute')
        ->once()
        ->withArgs(fn ($arg) => $arg === 'foo')
        ->andReturn('bar');

    // or
    $spy = $this->spy(MyActionnn::class);
    $spy->shouldReceive('execute')
        ->once()
        ->with(MyExampleModel::class, MyExampleData::class)
        ->andReturn('bar');
}

```

Event::fake

Event::assertDispatched

```
<?php

/** @test */
public function event_assert_dispatched(): void
{
    Event::fake();

    // or

    Event::fake([
        ExampleCreated::class,
    ]);

    $myExample = MyExampleFactory::new()->create();

    // Assert a event was dispatched...
    Event::assertDispatched(ExampleCreated::class);

    // Assert a event was dispatched...
    Event::assertDispatched(ExampleCreated::class, function ($event) use ($myExample) {
        return $event->myExample->is($myExample);
    });
}
```

Event::assertNotDispatched

```
<?php

/** @test */
public function event_assert_not_dispatched(): void
{
    Event::fake();

    // or

    Event::fake([
        ExampleCreated::class,
    ]);

    $myExample = MyExampleFactory::new()->create();

    // Assert a event was not dispatched...
    Event::assertNotDispatched(ExampleCreated::class);

    Event::assertNotDispatched(ExampleCreated::class, function ($event, $payload) {
        return $payload[0]->name === 'John Doe';
    });
}
```

Event::assertDispatchedTimes

```
<?php

/** @test */
public function event_assert_dispatched_times(): void
{
    Event::fake();

    // or

    Event::fake([
        ExampleCreated::class,
    ]);

    $myExample = MyExampleFactory::new()->create();

    // Assert a event was dispatched exactly n times...
    Event::assertDispatchedTimes(ExampleCreated::class, 1);
}
```

Event::assertListening

```
<?php

/** @test */
public function event_assert_listening(): void
{
    Event::fake();

    // or

    Event::fake([
        ExampleCreated::class,
    ]);

    $myExample = MyExampleFactory::new()->create();

    // assert that a listener is listening to a given event
    Event::assertListening(ExampleCreated::class, ExampleListener::class);
}
```

Queue::fake

Queue::assertPushed

```
<?php

/** @test */
public function examples_for_queue_fakes(): void
{

```



```

Queue::fake();

// or

Queue::fake([
    ExampleJob::class,
]);

// queue a job
ExampleJob::dispatch();

// assert that a job was pushed...
Queue::assertPushed(ExampleJob::class);

// assert that a job was pushed a given number of times...
Queue::assertPushed(ExampleJob::class, 1);

// assert that a job was pushed with a given payload...
Queue::assertPushed(ExampleJob::class, function ($job) {
    return $job->example == 'example';
});
}

```

Queue:: assertNotPushed

```

<?php

/** @test */
public function queue_assert_not_pushed(): void
{
    Queue::fake();

    // or

    Queue::fake([
        ExampleJob::class,
    ]);

    // queue a job
    ExampleJob::dispatch();

    // assert that a job was not pushed
    Queue::assertNotPushed(ExampleJob::class);

    // asser that a job was not pushed with a given payload...
    Queue::assertNotPushed(ExampleJob::class, function (ExampleJob $job) {
        return $job->exampleProperty === 'exampleValue';
    });
}

```

Queue:: assertNothingPushed

```

<?php

/** @test */
public function queue_assert_nothing_pushed(): void
{
    Queue::fake();

    // or

    Queue::fake([
        ExampleJob::class,
    ]);

    // queue a job
    ExampleJob::dispatch();

    // assert that no jobs were pushed...
    Queue::assertNothingPushed();
}

```

Queue::assertPushedOn

```

<?php

/** @test */
public function queue_assert_pushed_on(): void
{
    Queue::fake();

    // or

    Queue::fake([
        ExampleJob::class,
    ]);

    // queue a job
    ExampleJob::dispatch();

    // assert that a job was pushed on a given queue...
    Queue::assertPushedOn('queue-name', ExampleJob::class);

    // assert that a job was pushed a given number of times on a given queue...
    Queue::assertPushedOn('queue-name', ExampleJob::class, 1);

    // assert that a job was pushed with a given payload on a given queue...
    Queue::assertPushedOn('queue-name', ExampleJob::class, function ($job) {
        return $job->example == 'example';
    });
}

```

Queue::assertPushedWithChain

```
<?php
```

```
/** @test */
public function queue_assert_pushed_with_chain(): void
{
    Queue::fake();

    // or

    Queue::fake([
        ExampleJob::class,
    ]);

    // queue a job
    ExampleJob::dispatch();

    // assert that a job was pushed with a given chain...
    Queue::assertPushedWithChain(ExampleJob::class, [
        new AnotherJob,
        new YetAnotherJob,
    ]);

    // assert that a job was pushed with a given chain...
    Queue::assertPushedWithChain(ExampleJob::class, [
        new AnotherJob,
        new YetAnotherJob,
    ], function ($job) {
        return $job->user->id === 1;
    });
}
```

Queue::assertPushedWithoutChain

```
<?php
```

```
/** @test */
public function queue_assert_pushed_without_chain(): void
{
    Queue::fake();

    // or

    Queue::fake([
        ExampleJob::class,
    ]);

    // queue a job
    ExampleJob::dispatch();

    // assert that a job was pushed without a given chain...
    Queue::assertPushedWithoutChain(ExampleJob::class);

    // assert that a job was pushed without a given chain and with a given payload...
    Queue::assertPushedWithoutChain(ExampleJob::class, function ($job) {
```

```
    return $job->exampleProperty === 'exampleValue';  
  });  
}
```


Validation

prepareForValidation

```
<?php

use Illuminate\Validation\Rule;
use Smake\Common\Http\Requests\Request;

class EditUserRequest extends Request
{
    protected function prepareForValidation()
    {
        parent::prepareForValidation();

        $this->merge([
            'nickname' => $this->input('nickname', '') . ' with Love',
        ]);
    }
}
```

Rule::unique with ignore method on update

Adding a record:

```
<?php

use Illuminate\Validation\Rule;
use Smake\Common\Http\Requests\Request;

class AddUserRequest extends Request
{
    public function rules()
    {
        return [
            'email' => [
                'required',
                'email:strict',
                'max:255',
                Rule::unique('users', 'email'),
            ],
        ];
    }
}
```

Update an existing record:

```

<?php

use Illuminate\Validation\Rule;
use Smake\Common\Http\Requests\Request;

class EditUserRequest extends Request
{
    public function rules()
    {
        return [
            'email' => [
                'required',
                'email:strict',
                'max:255',
                Rule::unique('users', 'email')->ignore($this->route('userId')),
            ],
        ];
    }
}

```

passedValidation

```

<?php

use Illuminate\Validation\Rule;
use Smake\Common\Http\Requests\Request;

class EditUserRequest extends Request
{
    protected function passedValidation()
    {
        $this->merge([
            'nickname' => $this->input('nickname', '') . '❤️',
        ]);

        $this->getValidatorInstance()->setData(
            collect($this->input())
                ->only(collect($this->rules())->keys())
                ->all()
        );
    }
}

```

Kommandos

mkdocs help

```
mkdocs --help
```

mkdocs new

```
mkdocs new
```

mkdocs build

```
mkdocs build
```

mkdocs serve

```
mkdocs serve
```

mkdocs serve --livereload

```
mkdocs serve --livereload
```


Install python3

Weitere Information siehe python3.de

Install mkdocs

```
pip3 install mkdocs
```

Pip Kommandos

pip3

```
pip3 --version
```

Plugins

mkdocs-glightbox

```
pip3 install mkdocs-glightbox
```

mkdocs-minify-plugin

```
pip3 install mkdocs-minify-plugin
```

mkdocs-static-i18n

```
pip3 install mkdocs-static-i18n  
pip3 install mkdocs-static-i18n --use-pep517
```

mkdocs-admonition

```
pip3 install mkdocs-admonition
```

mkdocs-pdf-export-plugin

```
pip3 install mkdocs-pdf-export-plugin
```

Install python3

```
python3 -m ensurepip --upgrade
```

python3 --version

```
python3 --version
```

Templates

```
pip3 install mkdocs-material
```