

Welcome to knowledge-base

# 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"
        ]
    }
}
```

# 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::helper

## Throw an exception

```
<?php

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

// or
$apiKey = $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",
]
```



# 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](#)

# 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](#)

# 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 ListUserQuery($request);
    }
}
```





# 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

## Install python3

Weitere Information siehe [Python3.de](https://python3.de)

## Install mkdocs

```
pip3 install mkdocs
```

# Pip Commands

## 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
```

# Python3

## Install python3

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

## python3 --version

```
python3 --version
```

# Templates

```
pip3 install mkdocs-material
```



# Kommandos

pest: composer test:pest-coverage

Siehe hier: [Pest.de](#)

pest: Datei Testen

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



# Call destruct

## 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](#)

# 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
```

# 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
```