Published in Dev Genius

.com software Follow
Sep 1, 2022 · 3 min read · ✨ · Listen

Save

Start Using “__invoke” in PHP

Improve your SOLID-code programming skills with this simple yet powerful technique of self-contained classes.

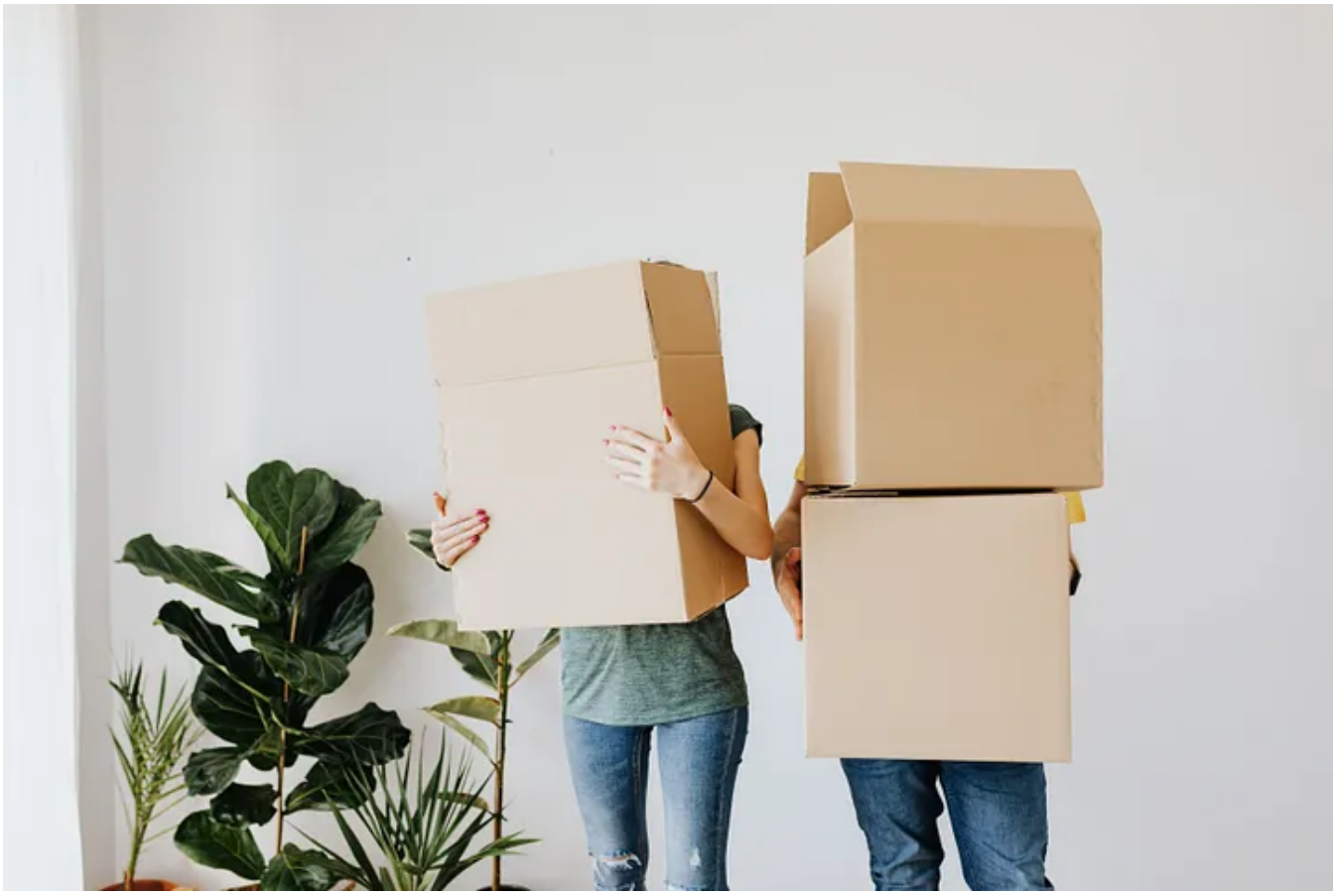

Image by Karolina Grabowska

Today I would like to share with you a useful programming technique that I use every day in my professional work.

Have you ever heard of **invokable classes**, also called **callable classes**? Certainly, some of you have met with them, for example in the so-called “single action controllers” in your favorite frameworks like Laravel or Symfony.

The callable classes have been around in PHP for some time. I saw them personally many years ago, but I appreciated their beauty only recently.

What is a callable class?

It’s just a class that has the `__invoke` magic method defined:

```php
<?php

class Adder
{
    public function __invoke(int $a, int $b): int
    {
        return $a + $b;
    }
}
```

invokable-example.php hosted with ❤ by **GitHub** **view raw**

I assume you’re using at least PHP version 7. The preferred method of running such a class is simple. If a class property is an *invokable* then it has to be surrounded with parenthesis, eg:

```php
<?php

class Calculator
{
    public function __construct(
        private Adder $adder
    ) {}

    public function add(): int
    {
        return ($this->adder)(2, 3);
    }
}

$calculator = new Calculator();
echo $calculator->add(); // prints 5
```

invokable-example-run.php hosted with ❤ by GitHub view raw

If a variable is a callable then simply:

```php
<?php

$adder = new Adder();
echo $adder(3, 4); // prints 7
```

invokable-example-run2.php hosted with ❤ by GitHub view raw

How callable classes are useful in SOLID?

The star of today’s episode is the letter “S” from the “SOLID” acronym, or “Single Responsibility Principle”. I assume you can’t wait to see how callable classes can help you create better and more beautiful code.

The rules are simple. The callable class:

- is supposed to have a single behavior (eg. “add a product to a basket”, “remove a product from a basket”, etc.)
- must have only the “__invoke” method and it cannot have any other public methods
- can have as many private methods as it wants for the internal needs

- its name should reflect its behavior. The lack of specifically named public methods enforces this rule (more on that later)

Good candidates for callable classes

Every class that is supposed to do something specific:

- providers returning a value (TaxAmountProvider, ConfigurationProvider, SomeDataProvider, etc.)
- validators (EmailValidator, VatinValidator, AgeValidator, etc.)
- controllers (LoginAction, HomepageAction, RegistrationFormAction, etc.)
- specifications (CanAddProductSpecification, IsSuperAdminSpecification, etc.)
- services (CalculateTaxService, AddToBasketService, etc.)

Real life example

I would like to demonstrate a real-life example. Let’s take a usual order management service:

```php
<?php

class OrderService
{
    public function addProduct(Order $order, Product $product): void {}
    public function removeProduct(Order $order, Product $product): void {}
    public function addTax(Order $order, Tax $tax): void {}
    public function addShippingCost(Order $order, ShippingCost $cost): {}
    public function empty(Order $order): void {}
}
```

invokable-order-service.php hosted with ❤ by **GitHub** **view raw**

I am pretty sure this class is going to be modified over and over again. How would you define the single responsibility limit of such class? It will always be tempting to extend this class with new features. After all everything is related to the order so it’s cool right?

```php

<?php

class PromotionalOrderService extends OrderService
{
    public function addGratisProduct(Order $order, Product $product): void {}
    public function removeGratisProduct(Order $order, Product $product): void {}
    public function applyDiscount(Order $order, DiscountCalculator $calculator): void {
    public function sendPromotionalEmail(Order $order, User $user): void {}
}
```

invokable-order-service2.php hosted with ❤ by GitHub view raw

Can you see the issue here? The class is going to be a pain to maintain. By the way, from the point of view of an outsider, are you able to tell what a generically named `OrderService` does? You won’t be able to without looking through the entire class, while this one will have at least 1000 lines. Good luck, have fun.

Invokable classes to the rescue

What if we used the wonderful technique of callable classes? We would achieve something like this:

```php
<?php

class AddToOrder {
    public function __invoke(Order $order): void {}
}

class RemoveFromOrder {
    public function __invoke(Order $order, Product $product): void {}
}

class EmptyOrder {
```



```php
}

class AddShippingCostToCart {
    public function __invoke(Order $order, ShippingCost $cost): void {}
}
```

invokable-order-actions.php hosted with ❤ by GitHub view raw

Isn’t this already beautiful?

- real concern separation,
- classes can’t have more than one responsibility (the “S” from “SOLID”) due to the fact there’s only one public method allowed,
- class functionality is obvious without going through the code,
- classes are **much, much easier** to cover with unit tests,
- unit tests can be more precise,
- order actions are endlessly extensible,
- order actions can be separately modified leading to less regressions.

Do you like invokable classes already? I fell in love with them, and I hope you will give them some love too.

Just don’t rush with refactoring. It’s up to you to decide if this technique will be helpful in your use case.

There’s one publication of mine I’m currently working on where I use invokable classes extensively. See for yourself how I use them in practice:

Serving resized S3 images on the fly

We will create a Symfony based application that serves resized images on the fly, directly from AWS S3

medium.com

166 | 7

PHP Php Developers Php Development Laravel Symfony

Enjoy the read? Reward the writer.[Beta]

Your tip will go to .com software through a third-party platform of their choice, letting them know you appreciate their story.

Give a tip

Sign up for DevGenius Updates

By Dev Genius

Get the latest news and update from DevGenius publication Take a look.

By signing up, you will create a Medium account if you don’t already have one. Review our Privacy Policy for more information about our privacy practices.

Get this newsletter