

# Rotte con "chiusure"

- Come si è visto, nei casi semplici, ogni rotta viene associata con un *callback* che è una "chiusura" (funzione anonima)
  - questo ci risparmia il disturbo di trovare un nome al callback!
- Esistono però delle convenzioni utili sui nomi (e i contenuti) dei callback
- Nell'esempio sopra, si ripete uno schema: ogni callback non fa altro che servire una *Page* (web), sostanzialmente statica (a meno dei parametri)
- Altre pagine simili da servire allungherebbero a dismisura la lista delle *Route* e dei relativi callback/closure (qui brevi ma, in altri casi, lunghi)
- È più ordinato ricondurre tutti i callback a metodi di una classe che chiameremo *PageController* ("controller delle pagine")
- Si genera uno (schema, da riempire, di) **controller** con il tool *artisan*:

```
<!-- web.php -->

<?php
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('home'); });

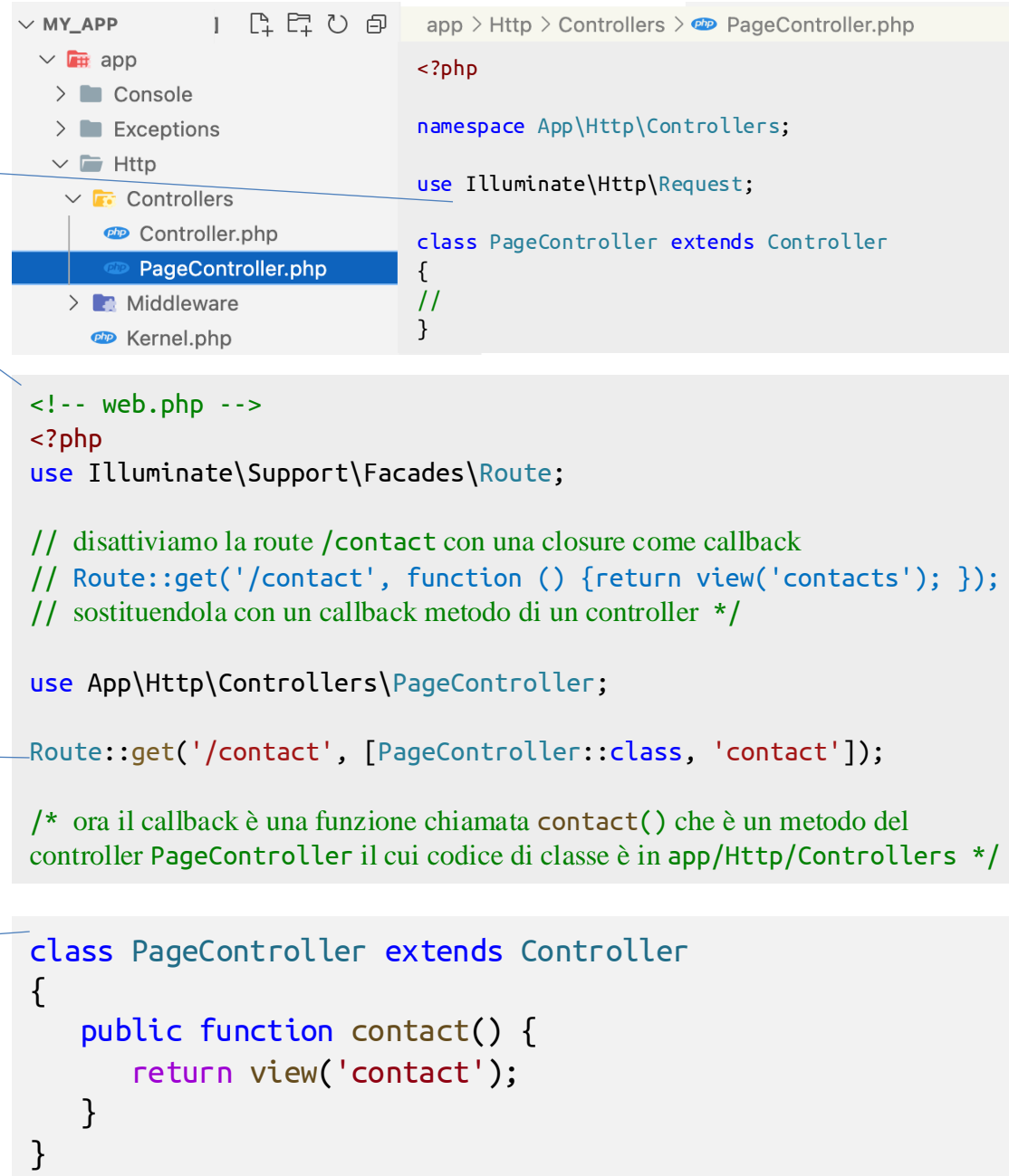
Route::get('/contact', function () {
    return view('contact'); });

Route::get('/about', function () {
    return view('about'); });
```

```
my_app $ php artisan make:controller PageController
Controller created successfully
```

# PageController

- Codice PHP "boilerplate", generato con *artisan*
- In *web.php*, in ogni route da definire, si inserisce come callback un metodo del *PageController*, specificando come qui a destra
- La route qui associa al path *'/contact'*, il callback *contact()*
- Questo è definito come metodo della classe *PageController* e fa ciò che faceva la closure soppiantata, cioè restituisce la view *contact.blade.php*



# Route e Controller

- Un **controller** serve quindi a riunire in una sola classe dei callback associati a path riferibili a un gruppo omogeneo di "*risorse*" su cui opera la Web app

```
<!-- web.php -->
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\PageController;

Route::get('/', [PageController::class, 'home']);
Route::get('/contact', [PageController::class, 'contact']);
Route::get('/about', [PageController::class, 'about']);
```

```
<?php
// file app/Http/Controllers/PageController.php

namespace app\Http\Controllers;
use Illuminate\Http\Request;

class PageController extends Controller
{
    public function home() {
        return view('welcome'); }
    public function contact() {
        return view('contact'); }
    public function about() {
        return view('about'); }
}
```

- Qui le risorse associate ai path e su cui operano i callback di *PageController* sono pagine Web statiche (le view servite dall'app ai clienti)
- In altri casi le risorse saranno *User*, *Book*, *Car*, *Team* ... cui corrisponderanno i controller *UserController*, *BookController*, *CarController*, *TeamController* ...
- Probabilmente i metodi di questi controller non si limiteranno a invocare solo una view, ma conterranno anche "business logic"

# Componenti Laravel finora

Finora abbiamo introdotto tre componenti software principali per Laravel:

- *controllers*
- *views*
- *routes*

Mancano i...

- *models*

