Middleware

Sidrit Trandafili

@sidis405

@strandafili

- Sviluppatore PHP 15+ anni
- Laravel da 4 anni
- Back-end
  - php
  - java
  - node
  - python
- Front-end
  - VueJS
  - Vanilla JS
  - TailwindCSS
  - Sass
- Devops + Sysops da 13 anni

Cosa sono i Middleware?

Middleware

> *I middleware sono un meccanismo che serve a filtrare programmaticamente le richieste HTTP che riceve la vostra applicazione*

Laravel include un middleware di default che verifica se l'utente è autenticato. Se non lo è, il middleware reindirizzerà il visitatore alla schermata login. Se l'utente invece è, alla richiesta gli sarà permesso di continuare l'esecuzione come per intenzione.

Middleware

Esempi di altri usi di middleware:

- Un middleware può essere usato per verificare la chiave API inclusa in una richiesta.
- Per limitare il numero di richieste per secondo alla vostra applicazione.
- Cambiare la lingua del sito, basato sulla locale.
- Oppure per abilitare modalità 'under maintenance'.

Andremo a vedere come creare, registrare ed usare middleware nel nostro progetto. Creeremo un middleware per mettere in 'Manutenzione' certi url o l'intero progetto

Middleware

Per creare un middleware possiamo usare artisan

```
php artisan make:middleware Manutenzione
```

Il contenuto del nuovo middleware verrà inserito in:

app/Http/Middleware/Manutenzione.php

```php
<?php

namespace App\Http\Middleware;

use Closure;

class Manutenzione
{
    /**
     * Handle an incoming request.
     */
    public function handle($request, Closure $next)
    {
        return $next($request);
    }
}
```

Middleware

Noi vogliamo bloccare la richiesta. Dunque un'opzione sarebbe creare una HttpException con codice 503.

Questo ci garantisce che Laravel restituirà la view resources/views/errors/503.blade.php

```php
<?php

namespace App\Http\Middleware;
use Symfony\Component\HttpKernel\Exception\HttpException; // <---- importiamo la classe

use Closure;

class Manutenzione
{
    /**
     * Handle an incoming request.
     */
    public function handle($request, Closure $next)
    {
        throw new HttpException(503); // <----- Exception 503
    }
}
```

Middleware

Adesso che il middleware è stato configurato, dobbiamo far capire all'applicazione che il middleware esiste.

Tale processo avviene in app/Http/Kernel.php

Se vogliamo eseguire sempre il middleware, lo dobbiamo inserire nell'array $middleware del Kernel

```
protected $middleware = [
    ...
    \App\Http\Middleware\Manutenzione::class
];
```

In questo caso l'utente vedrà sempre il messaggio di manutenzione

Se vogliamo eseguire selettivamente il middleware ed attivarlo solo per certe routes, lo dobbiamo piazzare nell'array $routeMiddleware del Kernel, con nome e path FQDN della classe.

```
protected $routeMiddleware = [
    ...
    'manutenzione' => \App\Http\Middleware\Manutenzione::class,
    ...
];
```

Middleware

Prendiamo l'esempio del post

```
    Route::get('posts/{post}', 'PostController@show');
```

Adesso per allacciare un middleware a questa route in particolare

```
    Route::get('posts/{post}', 'PostController@show')->middleware('manutenzione');
```

Per allacciare più middleware ad una route

```
    Route::get('posts/{post}', 'PostController@show')
        ->middleware(['manutenzione', 'auth', 'logger-richieste']);
```

Per allacciare più middleware ad un gruppo di routes

```
Route::group(['middleware' => ['manutenzione']], function () {

    Route::get('posts/{post}', 'PostController@show');
    Route::get('posts/{post}/modifica', 'PostController@edit');
});
```

Middleware

E se non volessimo usarle nel file delle routes ma nel controller invece per un controllo più granulare?

```php
<?php

namespace App\Http\Controllers;

use App\Post;
use Illuminate\Http\Request;

class PostsController extends Controller
{

    function __construct()
    {
        $this->middleware('manutenzione');
    }
```

Solo su certe routes:

```php
$this->middleware('manutenzione')->only('create', 'store', 'destroy');
```

A tutte le routes, eccetto certe:

```php
$this->middleware('manutenzione')->except('index', 'edit', 'update');
```