



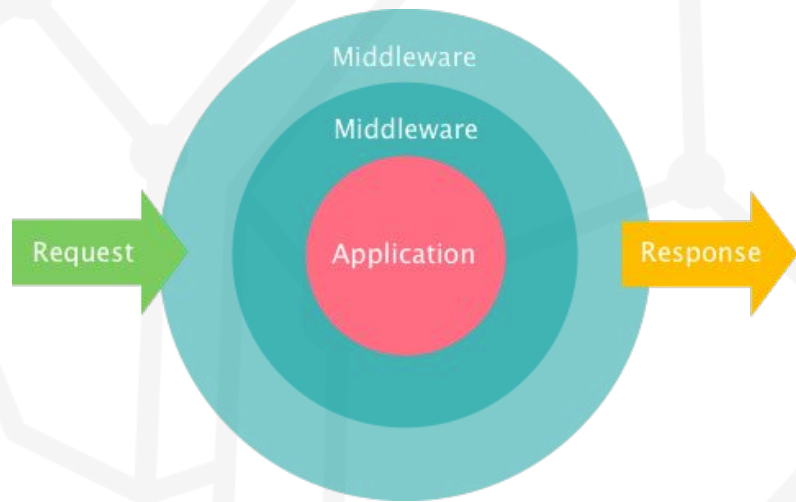
Laravel

Middleware

I Middleware

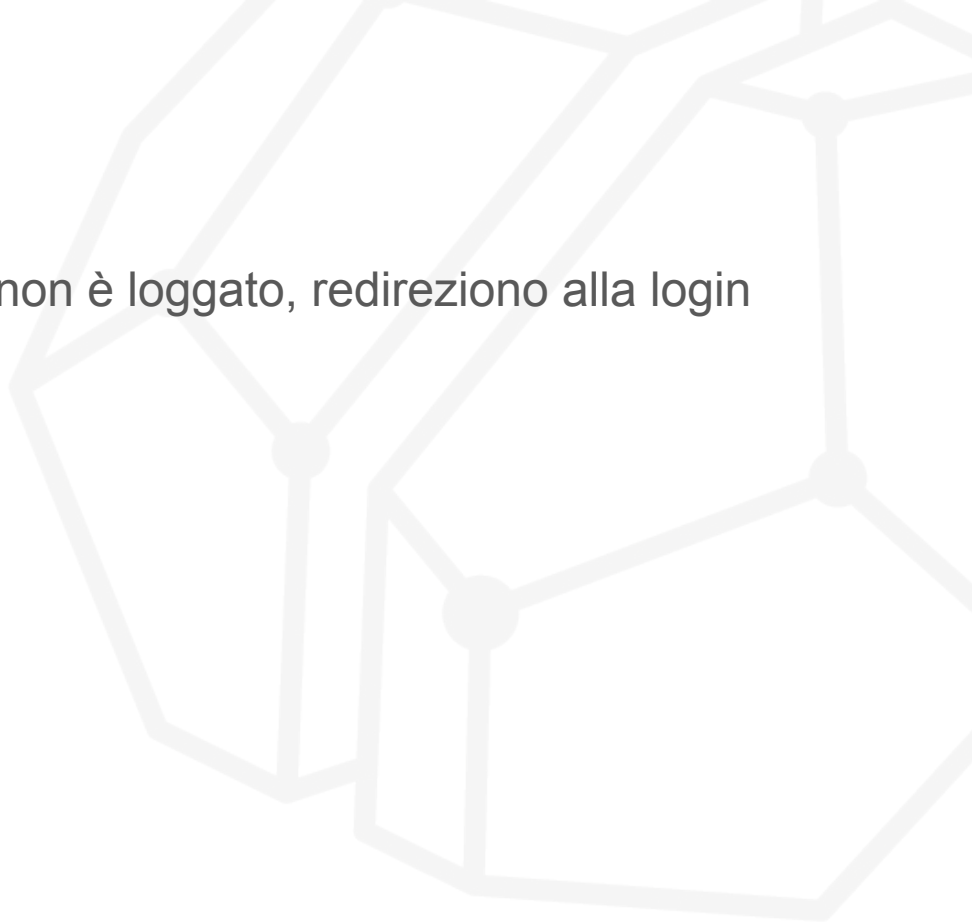
Il Middleware è un layer aggiuntivo che si frappone tra il client e il cuore dell'applicazione, nello specifico tra richieste/risposte e i controller

- può modificare le risposte in uscita
- può modificare le richieste in ingresso
- può in generale agire in maniera generica adottando logiche e controlli su richieste, sessioni ecc.



Esempi di Middleware

- controllo autenticazione: se l'utente non è loggato, reindirizzano alla login
- controllo indirizzo ip sorgente (filtro)
- inserimento di custom headers
- gestione CORS



Creazione di un Middleware

In Laravel, è possibile creare un Middleware attraverso il comando

```
php artisan make:middleware NomeDelMiddleware
```

Verrà creato un file nel percorso `app/Http/Middleware`

Implementazione di un middleware

La funzione `handle()` gestisce il cuore del middleware stesso.

La closure `$next` si occupa di gestire l'avanzamento della richiesta nei vari layer.

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

Middleware per modificare la response

Recuperiamo la risposta da `$next` e aggiungiamo un header HTTP custom

```
public function handle($request, Closure $next)
{
    $response = $next($request);
    $response->headers->set('X-Powered-by', 'Laravel REST API!');
    return $response;
}
```

Middleware per controllare una request

Ad esempio, potrei voler controllare l'indirizzo ip di provenienza della request; abilitare solo un indirizzo all'accesso ed emettere un errore negli altri casi



```
public function handle($request, Closure $next)
{
    if ($request->ip() != '127.0.0.1') {
        abort(403);
    }
    return $next($request);
}
```

Abilitare il middleware su tutte le rotte

Per abilitare il middleware su ogni richiesta effettuata, quindi sull'intero progetto, è necessario aggiungere la classe del middleware nella proprietà `$middleware` della classe `app/Http/Kernel.php`

Registrare un middleware

Per aggiungere un middleware su rotte specifiche o su un controller, è necessario prima registrarlo all'interno di `app/Http/Kernel.php` inserendolo nell'array `$routeMiddleware` e associandogli un "alias"

```
protected $routeMiddleware = [  
    // ...  
    'my_middleware' => \App\Http\Middleware\MyMiddleware::class,  
];
```

Assegnare un middleware a rotte specifiche

Può essere necessario abilitare un middleware per rotte specifiche. Il Middleware deve essere già registrato in `app/Http/Kernel.php`

```
Route::get('admin/profile', function () {  
    //  
})->middleware('auth');  
// Es. file routes/api.php
```

Link alla documentazione:

<https://laravel.com/docs/6.x/middleware#assigning-middleware-to-routes>

Abilitare un middleware in un controller

Può essere necessario abilitare un middleware per tutte le rotte contenute in un controller. Il Middleware deve essere già registrato in `app/Http/Kernel.php`

```
class UserController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
        $this->middleware('log')->only('index');
        $this->middleware('subscribed')->except('store');
    }
} //Es. file app/Http/Controllers/UserController.php
```



Q & A