# Laravel: Créer et utiliser un middleware

[sommaire]

1. ## Objectifs

- Etre capable de créer un middleware sous Laravel
- Savoir utiliser un middleware sous Laravel

- ## Présentation

- ## Créer middleware

  - La façon la plus simple de créer un middleware est d'utiliser Artisan. Appeler la commande `make:middleware` a pour effet de créer une structure de base de middleware dans le dossier réservé à cet usage : **app/Http/Middleware**.
  - **`php artisan make:middleware EnsureTokenIsValid`**
  - Cette commande placera une nouvelle classe **EnsureTokenIsValid** dans le répertoire **app/Http/Middleware**.
  - Dans ce middleware, nous n'autoriserons l'accès à la route que si l'entrée token fournie correspond à une valeur spécifiée. Sinon, nous redirigerons les utilisateurs vers l'URI home:

```
<?php

namespace App\Http\Middleware;

use Closure;

class EnsureTokenIsValid
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
```

apcpedagogie.com

```
    {
        if ($request->input('token') !== 'my-secret-token') {
            return redirect('home');
        }

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

- ## Enregistrement d'un middleware

  - Maintenant que nous avons créé un middleware, nous devons informer l'application que le middleware existe. Si vous souhaitez qu'un middleware s'exécute à chaque demande, accédez à **`app/Http/kernel.php`** et ajoutez le **middleware** au Kernel.

- ## Activité

  - **Étape 1: créer un middleware**

    - Dans cette étape, nous devons créer un middleware personnalisé dans un projet basé sur laravel. Alors ouvrons votre invite de commande et exécutons la commande ci-dessous:
    - `php artisan make:middleware CheckStatus`

    - Après avoir créé avec succès le middleware, accédez à **app/http/kernel.php** et enregistrez votre **middleware** personnalisé ici:

```
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    ....
```

apcpedagogie.com

```
    /**
     * The application's route middleware.
     *
     * These middleware may be assigned to groups or used individually.
     *
     * @var array
     */
    protected $routeMiddleware = [
        ....
        'checkStatus' => \App\Http\Middleware\CheckStatus::class,
    ];
}
```

- Après avoir enregistré avec succès votre middleware dans le projet laravel, accédez à **app/http/middleware** et implémentez votre logique ici: **app/Http/Middleware/CheckStatus.php**

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class CheckStatus
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle(Request $request, Closure $next)
    {
        if (auth()->user()->status == 'active') {
            return $next($request);
        }
            return response()->json('Your account is inactive');
    }
}
```

- **Étape 2: Ajouter une route**

apcpedagogie.com

- Dans cette étape, nous allons simplement créer une route et utiliser un **middleware** personnalisé ici. Filtrez chaque requête http et protégez les routes:

```
//routes/web.php

 Route :: middleware (['checkStatus']) -> group (function () {

Route :: get ('home', ' HomeController @ home ');

});
```

- **Étape 3: Ajouter une méthode dans le contrôleur**
  - Nous allons maintenant créer un langage de nom de méthode et vérifier: **app/Http/Contrôleurs/HomeController.php**

```
<?php

 namespace App\Http\Controllers;

 use Illuminate\Http\Request;
 class HomeController extends Controller
 {
     public function home()
     {
         dd('You are active');
     }
 }
```

**Source:**

- https://www.tutsmake.com/laravel-6-custom-middleware-tutorial-example/

---

**Cours Laravel**

Source : https://apcpedagogie.com/laravel-creer-et-utiliser-un-middleware · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com