

2 minutes

Limiter les méthodes HTTP

Limiter les méthodes HTTP

Les principaux verbes, également appelés méthodes, HTTP sont les suivant :

GET : indique que l'on veut récupérer une ressource à l'URI spécifiée. Les requêtes GET doivent uniquement être utilisées pour obtenir des données (qui peuvent être une page HTML).

POST : indique que l'on souhaite envoyer une entité vers l'URI spécifiée. La plupart du temps, le serveur enregistre les données envoyées dans une base de données.

PUT : indique que l'on souhaite remplacer une entité par celle envoyée vers l'URI. On l'utilise le plus souvent pour remplacer complètement une entité enregistrée dans une base de données.

PATCH : indique que l'on souhaite modifier partiellement une entité avec les changements envoyés vers l'URI. On l'utilise le plus souvent pour modifier partiellement une entité enregistrée dans une base de données.

DELETE : indique que l'on souhaite supprimer la ressource indiquée par l'URI.

Dans le cadre du Routing Symfony, nous pouvons limiter les méthodes utilisables sur une route.

Pour ce faire, il suffit de préciser le **paramètre nommé methods** :

```
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class DefaultController
{
    #[Route(path: '/', name: 'index', methods: ['GET'])]
    public function index(Request $request)
    {
        dd($request);
        return new Response('<h1>Hello World !</h1>');
    }

    #[Route(path: '/blog', name: 'blog', methods: ['PUT', 'POST'])]
    public function blog()
    {
        return new Response('<h1>Blog</h1>');
    }
}
```

Copier

Avec cette configuration, la route index ne peut plus être accédée qu'avec une requête GET et la route blog avec des requêtes PUT ou POST.

Essayez de vous rendre sur <https://127.0.0.1:8000/blog>.

Vous aurez l'erreur : No route found for "GET https://127.0.0.1:8000/blog": Method Not Allowed (Allow: PUT, POST).

L'erreur signifie qu'aucune route n'est trouvée pour une requête GET sur le chemin /blog et que seules les méthodes PUT et POST sont autorisées sur cette route.

Limiter le protocole

La plupart du temps vous voudrez en production limiter l'accès à vos routes au seul protocole HTTPS (à savoir la version sécurisée du protocole HTTP, utilisant HTTPS).

Pour ce faire, il suffit de modifier le fichier de configuration que nous avons mis en place dans le dossier config/routes :

```
controllers:
  resource: '../..../src/Controller/'
  type: annotation
  defaults:
    schemes: [https]
```

Copier

Avec cette configuration, toutes les routes ne seront par défaut accessibles qu'en HTTPS.

C'est le plus courant, bien qu'il soit possible également de configurer route par route en utilisant le paramètre nommé schemes :

```
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class DefaultController
{
    #[Route(path: '/', name: 'index', methods: ['GET'],
schemes: ['https'])]
    public function index(Request $request)
    {
        dd($request);
        return new Response('<h1>Hello World !</h1>');
    }

    #[Route(path: '/blog', name: 'blog', methods: ['PUT',
'POST'])]
    public function blog()
    {
        return new Response('<h1>Blog</h1>');
    }
}
```

Copier

Ici la route index est maintenant accessible qu'en HTTPS.