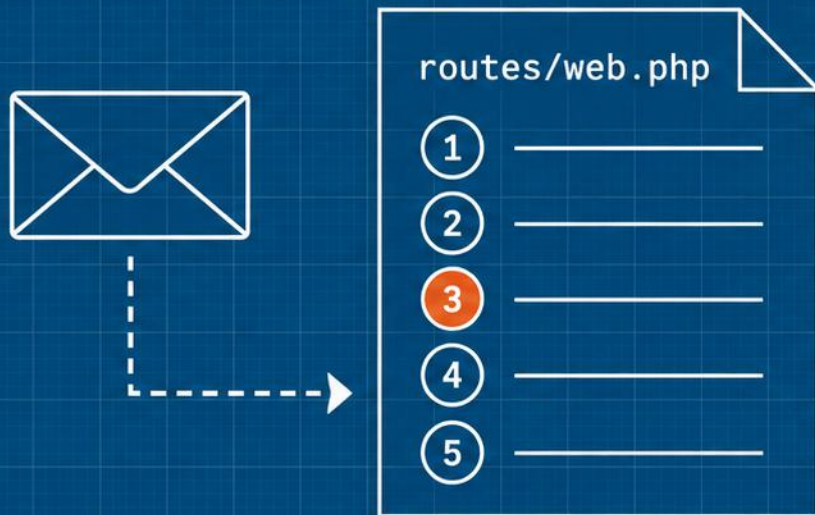


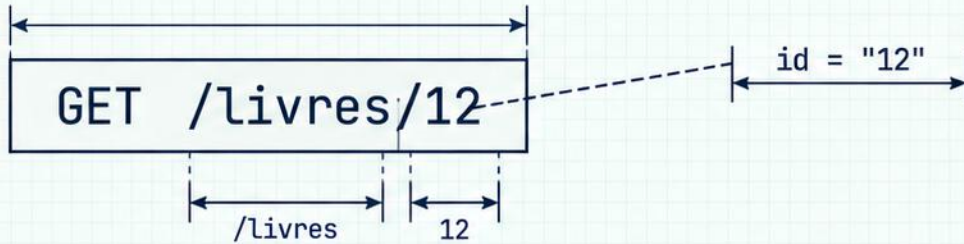
Bloc Laravel · chapitre 7

Routing

Laravel parcourt routes/web.php de haut en bas et garde la première route qui correspond à l'adresse et au verbe.



Une route, quatre morceaux



```
1 Route::get('/livres/{id}', [BookController::class, 'show'])->name('books.show');
```

1

1. Le verbe :
quelle méthode HTTP

2

2. Le chemin :
partie fixe et paramètre

3

3. L'action :
quel code répond

4

4. Le nom :
pour fabriquer l'adresse
(facultatif)

Dans ce cours, une route qui interroge un modèle ou prépare une vue avec des données appelle un contrôleur

Une fonction anonyme est une fonction sans nom, écrite directement à l'endroit où elle sert.

```
1 Route::get('/ping', function () {  
2     return 'pong';  
3 });
```

```
1 Route::get('/posts', [PostController::class, 'index']);
```



web.php

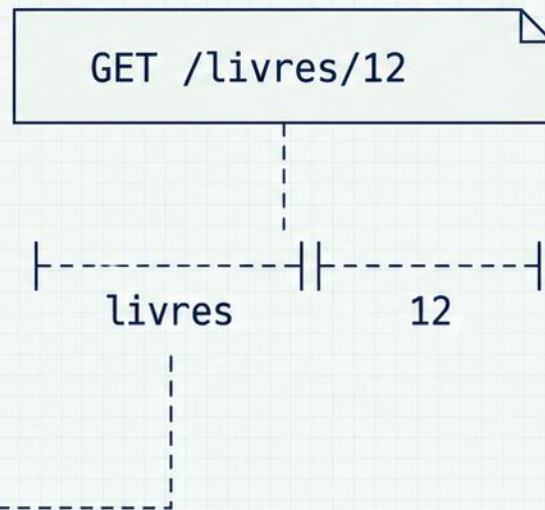
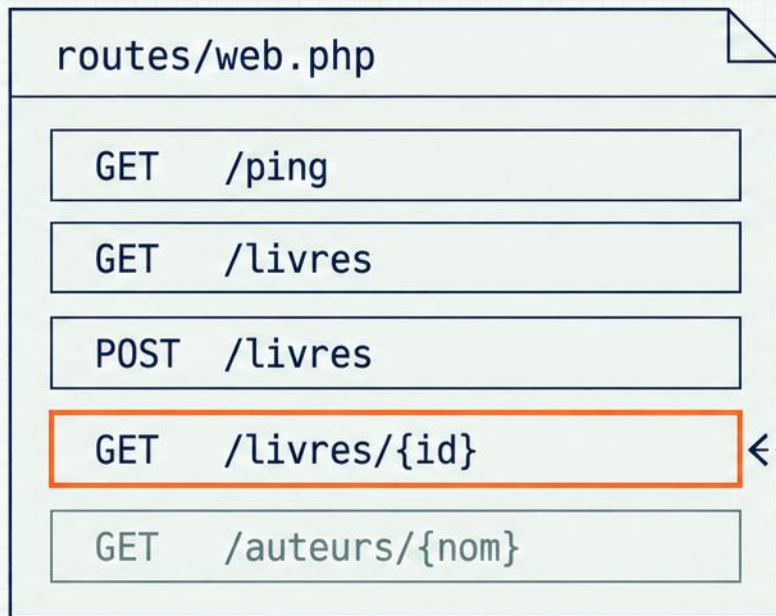


PostController.php

La première route qui répond oui deux fois l'emporte

Laravel découpe d'abord l'adresse en segments, les morceaux compris entre deux barres obliques.

1. Autant de segments, les mêmes mots fixes ?
2. Le bon verbe ?



La table de routage, pour cinq requêtes

L'ensemble des routes, parcouru dans cet ordre, s'appelle la table de routage.

Requête	Route retenue	Réponse
GET /livres	GET /livres	Liste des livres
POST /livres	POST /livres	Livre enregistré
GET /livres/12	GET /livres/{id}	Fiche du livre 12
GET /livres/12/avis	aucune	404
DELETE /livres	aucune pour ce verbe	405

404 : aucune route pour ce chemin. 405 : le chemin existe, pas pour ce verbe.

Un paramètre obligatoire

GET /livres/12

Dans '/livres/{id}', le segment {id} s'appelle un paramètre de route.

```
1 Route::get('/livres/{id}', function (string $id) {  
2     return "Fiche du livre $id";  
3 });
```

```
1 $ curl -s http://premier-test-laravel.test/livres/abc  
2 Fiche du livre abc
```

Laravel transmet le segment sous forme de chaîne de caractères, sans vérifier qu'il s'agit d'un nombre.

Plusieurs paramètres

GET /agenda/2026/10

```
1 Route::get('/agenda/{year}/{month}', function (int $year, int $month) {  
2     return "Agenda du mois $month de l'année $year";  
3 })->whereNumber(['year', 'month']);
```

```
1 $ curl -s http://premier-test-laravel.test/agenda/2026/10  
2 Agenda du mois 10 de l'année 2026
```

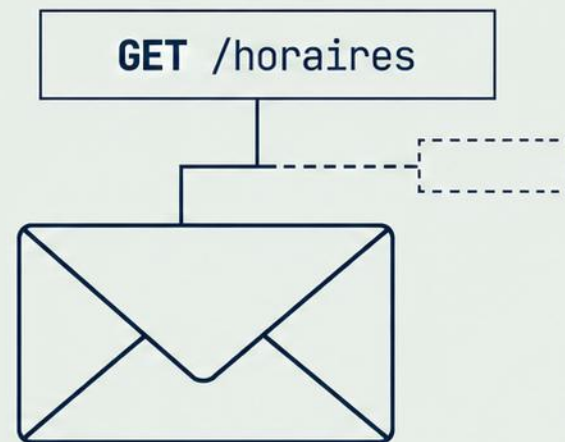
Laravel donne les valeurs à la fonction dans l'ordre où les accolades apparaissent dans le chemin, sans regarder le nom des variables.

Un paramètre optionnel

Un point d'interrogation placé juste après le nom, comme {day?}, rend le paramètre optionnel. La fonction doit alors prévoir une valeur par défaut.

```
1 Route::get('/horaires/{day?}', function (string $day = 'lundi') {  
2     $hours = match ($day) { ... };  
3     return "Le $day : $hours";  
4 });
```

```
1 $ curl -s http://premier-test-laravel.test/horaires  
2 Le lundi : de 9 h à 18 h  
3 $ curl -s http://premier-test-laravel.test/horaires/samedi  
4 Le samedi : de 9 h à 12 h
```



Chaque contrainte fixe une condition que la valeur d'un paramètre doit remplir

Contrainte	Accepte	Refuse
<code>whereNumber('amount')</code>	des chiffres	12.5
<code>whereAlpha('name')</code>	des lettres sans accent	Hergé
<code>whereIn('currency', ['usd', 'gbp'])</code>	une valeur de la liste	chf
<code>where('slug', '[a-z0-9-]+')</code>	le motif donné	Romans

Une valeur refusée : la route ne correspond pas, Laravel essaie la suivante, et répond 404 si aucune ne convient.

Le piège de l'ordre

```
1 Route::get('/recettes/{id}', function (string $id) {  
2     return "Recette $id";  
3 });  
4  
5 Route::get('/recettes/nouvelle', function () {  
6     return 'Formulaire de nouvelle recette';  
7 });
```

```
1 Recette 12  
2 Recette nouvelle
```

`{id}` accepte aussi nouvelle : la seconde route n'est jamais consultée.

routes/web.php

GET /recettes/nouvelle

1

Route::get('/recettes/{id}', ...)

id = "nouvelle"

NON CONFORME

2

Route::get('/recettes/nouvelle', ...)
jamais consultée

Déclarez les routes fixes avant les routes à paramètre qui commencent de la même façon

Avant

routes/web.php

① `Route::get('/recettes/{id}', ...)` ✓

`id = "nouvelle"`

② `Route::get('/recettes/nouvelle', ...)` jamais consultée

GET /recettes/nouvelle



Recette nouvelle



route fixe d'abord

Après

routes/web.php

① `Route::get('/recettes/nouvelle', ...)` ✓

② `Route::get('/recettes/{id}', ...)` jamais consultée

GET /recettes/nouvelle



Formulaire de nouvelle recette

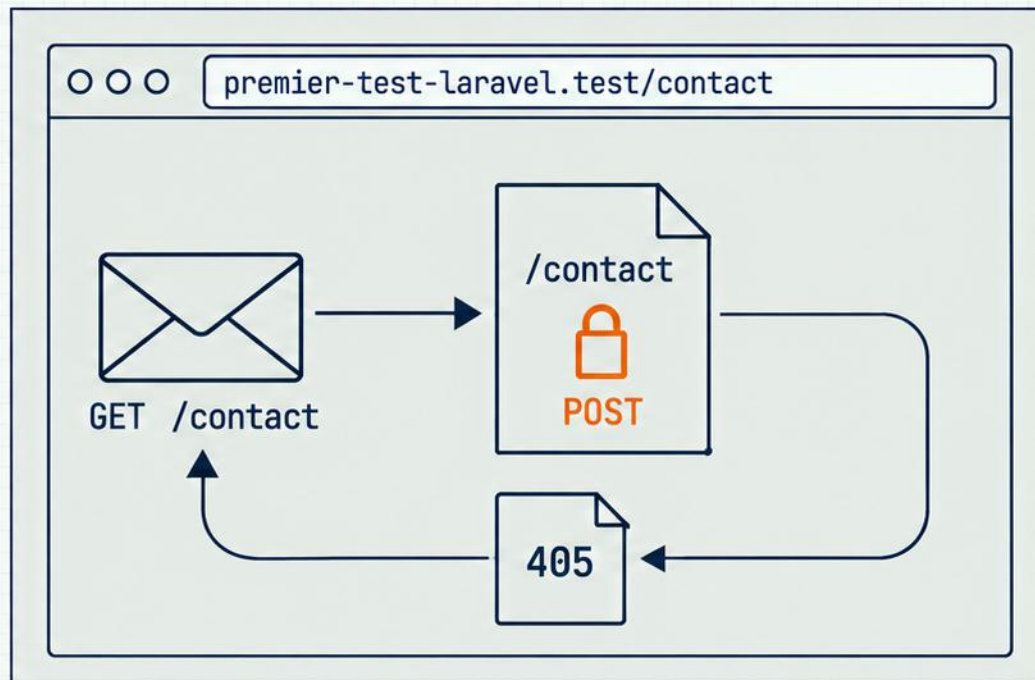
VÉRIFIÉ

```
1 Route::get('/recettes/{id}', function (string $id) { ... }->whereNumber('id'));
```

Ou gardez l'ordre et limitez {id} aux chiffres.

Le chemin désigne la ressource, le verbe indique ce qu'on veut en faire

Verbe	Usage	CRUD
GET	lire	Read
POST	créer, envoyer un formulaire	Create
PUT ou PATCH	modifier	Update
DELETE	supprimer	Delete

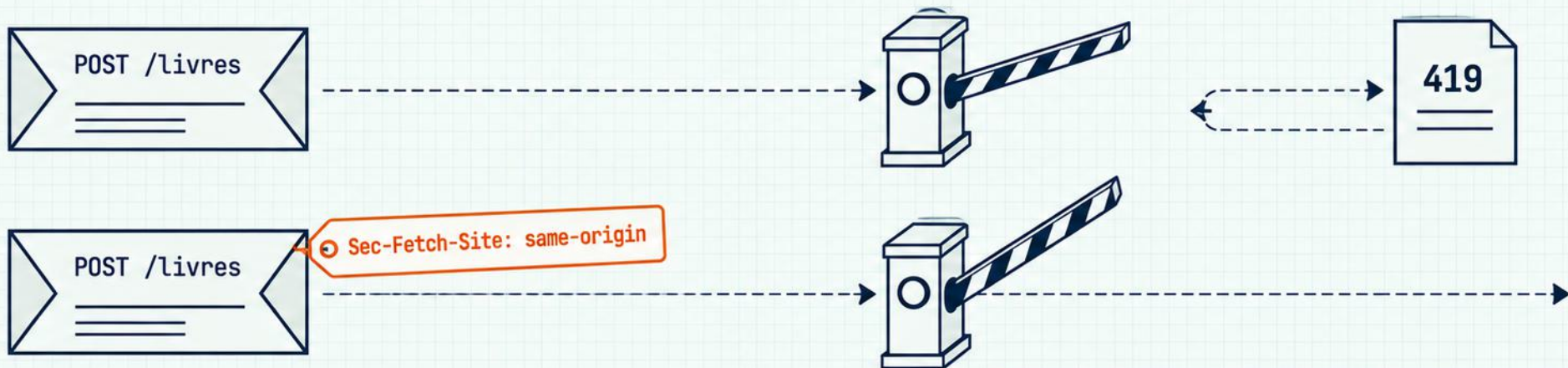


Le navigateur envoie toujours GET : `/contact`, déclarée avec `Route::post`, répond 405.

Laravel vérifie qu'une requête POST vient de votre propre site

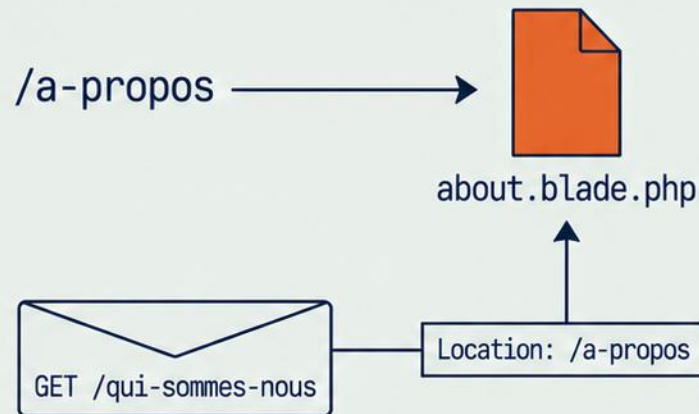
```
1 $ curl -s -X POST http://premier-test-laravel.test/livres
2 ... <title>Page Expired</title> ...
3 $ curl -s -X POST -H "Sec-Fetch-Site: same-origin" http://premier-test-laravel.test/livres
4 Livre enregistré
```

Cette vérification protège contre la CSRF, la falsification de requête entre sites.



Deux routes sans action : Route::view et Route::redirect

```
1 Route::view('/a-propos', 'about');  
2  
3 Route::redirect('/qui-sommes-nous', '/a-propos');  
4 Route::permanentRedirect('/presentation', '/a-propos');
```



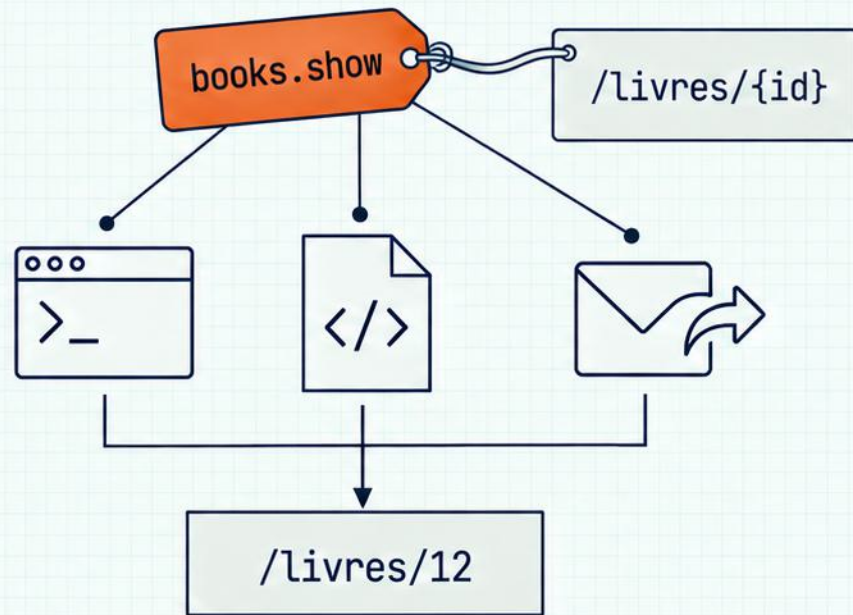
Une redirection est une réponse qui demande au navigateur d'ouvrir une autre adresse, donnée dans l'en-tête Location.
redirect : 302, temporaire. permanentRedirect : 301, définitive.

Une route nommée : route() produit son adresse à partir du nom

Une route nommée est une route qui a reçu un nom avec `->name()`.

```
1 Route::get('/livres/{id}', function (string $id) { ... })->name('books.show');
```

```
1 route('books.show', 12)
2 {{ route('books.show', 12) }}
3 redirect()->route('books.show', 12)
```



Dans le terminal, une vue Blade ou l'action d'une route, les trois donnent `http://premier-test-laravel.test/livres/12`.

Écrivez les liens et les redirections avec route() et le nom de la route

```
1 Route::get('/catalogue/livres/{id}', function (string $id) { ... })->name('books.show');
```

```
1 <a href="/livres/12">Lien écrit à la main</a>  
2 <a href="http://premier-test-laravel.test/catalogue/livres/12">Lien écrit avec route()</a>
```



Le lien écrit avec route() suit le nouveau chemin. Le lien écrit à la main mène à un 404.

À vous : routes, paramètres, ordre, noms

45 minutes, en séance ou chez vous.

- Ex. 1 à 6 : paramètres, POST, redirection.
- Ex. 7 à 9 : ordre, Route::view, routes nommées.
- Ex. 10, sur papier : réponse, 404 ou 405 ?

