

# Création d'un gestionnaire de membres de club sportif

## Évolution de la configuration de l'application

On va aller dans le dossier **config** puis dans le fichier **app.php**

Le fichier de configuration app.php définit les grandes variables de configuration communes à toute l'application.

Tu peux remarquer ce genre de lignes :

```
/*
|-----|
| Application Name |
|-----|
'name' => env('APP_NAME', 'Laravel'),
```

Ici, notre variable de configuration appelée "name" a pour valeur ce que l'on a renseigné dans APP\_NAME de notre .env

Si l'on n'a rien renseigné dans le .env, alors c'est la valeur par défaut qui est ici « Laravel ». Cette manière de faire permettra dans toute l'application d'accéder aux variables de configuration de manière simple.

Nous allons maintenant modifier ce fichier pour renseigner la langue du site (très utile pour que les parties en Anglais de Laravel soient automatiquement en Français !) et le fuseau horaire.

Modifions donc les variables "timezone", "locale" et "fallback\_locale" comme suit :

```
'timezone' => 'Pacific/Tahiti',
```

<https://www.php.net/manual/fr/timezones.pacific.php>

```
'locale' => 'fr',
```

```
'fallback_locale' => 'fr',
```

Cependant, même si on a renseigné la langue à "fr", il faut télécharger les fichiers qui permettent de traduire les phrases de base de Laravel en Français. Pour ça, on va installer le dossier de langues "fr" qu'on va piocher de ce repo GitHub :

<https://github.com/Laravel-Lang/lang>

On fait donc un :

```
composer require laravel-lang/lang:~8.0
```

Puis on a juste à déplacer le dossier de langues "fr" de vendor à resources/lang/fr et de même avec le fichier fr.json qu'on mettra dans resources/lang.

Pour ça on va utiliser les quelques commandes ci-dessous :

```
mkdir resources/lang/fr
mv vendor/laravel-lang/lang/src/fr/* resources/lang/fr
mv vendor/laravel-lang/lang/json/fr.json resources/lang/
ls resources/lang/fr
```

La dernière ligne doit afficher

Mode	LastWriteTime		Length	Name
----	-----		-----	----
-a----	11/10/2022	16:15	684	auth.php
-a----	11/10/2022	16:15	502	pagination.php
-a----	11/10/2022	16:15	828	passwords.php
-a----	11/10/2022	16:15	8127	validation-inline.php
-a----	11/10/2022	16:15	10699	validation.php

## Le routage de base

Nous allons maintenant définir proprement la **page d'accueil de notre projet...**

Pour cela nous allons modifier les règles de routage.

Comme nous l'avons vu dans le TP précédent, le routeur le plus basique de Laravel accepte une URL et retourne une valeur.

La valeur peut être une chaîne, une vue ou un contrôleur.

Toutes les routes sont stockées dans **routes/web.php** :

```
<?php
use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', function () {
    return view('bienvenue');
});
```

**Note** : par défaut `return view('welcome');`

Ce morceau de code signifie que lorsque le routeur Laravel reçoit '/', il renvoie une vue appelée 'welcome' qui est située à resources/views/welcome.blade.php.

Nous pourrions aussi lui faire renvoyer **une chaîne de caractères** :

```
Route::get('/', function () {
    return 'Bonjour étudiant !';
});
```

Ou nous pouvons lui faire renvoyer un **contrôleur**, qui pointe vers une **vue**

```
Route::get('/user', [UserController::class, 'index']);
```

Cette ligne de code signifie que si le routeur reçoit « /user », Laravel ira dans la section UserController et invoquera la fonction index.

## Création de la page d'accueil (index)

Pour cela nous allons modifier les règles de routage que nous avons utilisées pour nos tests. Nous allons les gérer à l'aide d'un contrôleur (Nous avons vu toutefois que nous pouvions nous contenter d'une simple route)

### Contrôleurs de base



Un contrôleur est un endroit où vous organisez les **comportements** de votre application. Par exemple, lorsque le routeur reçoit '/' il renvoie un 'IndexController'.

RQ : Le « IndexController » récupère les informations requises dans la base de données si besoin par l'intermédiaire d'un **modèle**, puis les place dans l'emplacement correspondant d'une **vue**, et enfin renvoie la vue au navigateur. C'est l'utilisation la plus basique d'un contrôleur.

Pour créer notre premier contrôleur. Allez dans le terminal, tapez :

```
php artisan make:controller IndexController
```

```
PS G:\01-BTS-SIO\02-MODULES\Bloc1-an2-S1\GestionClub\GestionClub> php artisan make:controller IndexController
Controller created successfully.
PS G:\01-BTS-SIO\02-MODULES\Bloc1-an2-S1\GestionClub\GestionClub>
```

Allez à  app/Http/Controllers/et vous trouverez un  IndexController.php

Nous allons faire en sorte que le IndexController nous retourne une vue.

```
class IndexController extends Controller
{
    //
    public function index()
    {
        return view(bienvenue);
    }
}
```

## Contrôleurs à action unique

Les contrôleurs à action unique sont utiles lorsque nous n'avons besoin que d'une seule méthode dans la classe du contrôleur :

```
class IndexController extends Controller
{
    public function __invoke()
    {
        return view(bienvenue);
    }
}
```

Maintenant, nous pouvons modifier la route. Nous n'avons plus besoin de spécifier la méthode...

Retournez à web.php et créez un le routeur comme ceci :

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\IndexController;

/*
|-----
| Web Routes
|-----
|
|
*/

Route::get('/', IndexController::class);
```

**Pour les autres pages comme « A propos » et « Contact », nous allons nous passer d'un contrôleur...** Ce sont des pages de données que l'on peut considérer comme statiques (très peu souvent modifiées).

Cela a deux avantages :

- c'est plus rapide à écrire !
- l'affichage de la page sera plus rapide qu'en passant par un contrôleur (car Laravel va charger moins de classes, faire moins d'actions...). On peut donc faire comme ça si on veut des performances plus intéressantes.

Une chose importante à connaître est l'ordre des routes !

Lisez bien ceci pour vous éviter des heures de recherches et de prises de tête.

La règle est : les routes sont analysées dans leur ordre dans le fichier des routes.

À vous de jouer ! créez un sous dossier « pages » dans votre dossier « views » et créez les deux pages « A propos », « Contact » (nous créerons le formulaire un peu plus tard avec l'apprentissage des modèles) et « Mentions légales »...

Mettre à jour le menu de navigation et rajoutez le lien vers les mentions légales dans le footer

## Ressources

<https://apcpedagogie.com/routage-sous-laravel/>

<https://apcpedagogie.com/creer-et-utiliser-un-controleur-laravel/>