

Module : M205 – Développement back-end

Filière : Développement digital – option web full stack

TP : Gestion de routage

Objectifs :

- Manipuler les routes de base

Exercice 1 :

1. Créer un nouveau projet Laravel
2. Créer la liste des routes pour répondre aux URL suivants :
 - `http://localhost /`
 - `http://localhost /login`
 - `http://localhost /page/1`

Afficher des chaines de caractères comme résultats

3. Soient les deux routes suivantes :

```
Route::get('{n}', function($n) {  
    return 'Je suis la page ' . $n . ' !';  
});  
Route::get('contact', function() {  
    return "C'est moi le contact."  
});
```

Que pensez-vous qu'il va se passer avec <http://localhost/contact>? Tester le.

Ajouter une route **/test**

- a. Ajouter un paramètre dont on souhaite afficher la valeur : **/test/param**
 - b. Premièrement, afficher la valeur du param comme chaîne de caractères ;
 - c. Deuxièmement, utiliser une vue pour cette route et afficher dans la vue la valeur du param :
 - Dans la route appeler la vue de cette façon :

```
return view('welcome', ["param" => $param]);
```
 - Dans la vue ajouter :

```
<div>Param vaut {{ $param }}</div>
```
 - d. Utiliser la méthode **Route ::view**
4. Ajouter une route qui a l'allure suivante :
`/Formation/{formation}/filier/{filier}/groupe/{groupe}/stagiaire/{stagiaire}`
 - a. Retourner une chaîne de caractères affichant les valeurs des paramètres passés.
 - b. Retourner une vue affichant les valeurs des paramètres passés.

- c. Modifier la route pour indiquer que le paramètre {stagiaire} est facultatif. Si {stagiaire} n'est pas indiqué, afficher l'année en cours à sa place.

5. Lister les routes avec la commande artisan.

Exercice 2 :

1. Créer deux routes /profile avec les deux verbes http get et post :

- a. **get** : permet d'afficher une vue comportant le formulaire suivant :

```
<form action="{{ route('profile') }}" method="post">
    {{ csrf_field() }}
    <label for="nom">Nom</label>
    <input type="text" name="nom" />
    <label for="email">Email</label>
    <input type="text" name="email" /><br>
    <input type="submit" value="ok" />
</form>
```

- b. **post** : permet de répondre aux soumissions du formulaire et d'afficher le nom et l'email indiqués par l'utilisateur ; Cette route doit être nommé '**Profile**'

Exercice 3 :

Créer une calculatrice web en laravel en suivant les étapes suivantes :

1. Créer sous le dossier « **view** » une vue appelée « **calculatriceView** » ;

- Ajouter à cette vue le formulaire de calculatrice suivant :

La méthode d'envoi des données du formulaires est « **post** » et l'action est « **/calculer** »

Calculatrice

Le résultat : 14

2. Ajouter une route nommée « **home** » affichant la vue « calculatriceView »

```
Route::view("/home/{result?}", "calculatriceView")->name("home");
```

3. Ajouter une route « **post** » avec l'URI « **/calculer** » qui permet de lire les différentes valeurs auprès du formulaire, effectuer les calculs et retourner le résultat à afficher en home.

```
Route::post("/calculer", function (Request $requete){  
    //Traitement  
    return redirect()->route("home", ["result"=>.....]);  
});
```