

# Documentation du code : Laravel - Calculatrice à notation polonaise

## 1. Routes

### Introduction

Les routes définissent les points d'accès de l'application, en dirigeant les requêtes HTTP vers les actions appropriées dans les contrôleurs. Dans ce code, deux routes sont définies pour gérer l'affichage de la vue et le traitement des calculs.

### Routes définies

#### 1.1 Route GET /test

- **Méthode** : GET
- **Action** : Affiche la page d'accueil de l'application, en l'occurrence la vue welcome.
- **Fonctionnement** : Lorsqu'un utilisateur accède à l'URL /test, la route renvoie simplement la vue welcome, sans effectuer de calculs.

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

#### 1.2 Route POST /test

- **Méthode** : POST
- **Action** : Envoie la requête à la méthode operation du **NombreController** pour traiter les données soumises par le formulaire.
- **Fonctionnement** : Lorsque l'utilisateur soumet des données (en appuyant sur les boutons), les données sont envoyées à cette route, qui déclenche la méthode operation du **NombreController**.

```
Route::post('/test', [NombreController::class, 'operation']);
```

### Conclusion

Les deux routes permettent de gérer l'affichage de la vue principale et le traitement des calculs via un formulaire HTML. La route GET sert à afficher la page, tandis que la route POST est utilisée pour effectuer des calculs via le contrôleur.

## 2. Controller (NombreController)

# Introduction

Le contrôleur **NombreController** est responsable de la gestion des calculs. Il reçoit les données envoyées par le formulaire, les traite et renvoie le résultat à l'utilisateur.

## Méthodes du contrôleur

### 2.1 Méthode operation()

- **But** : Effectuer les calculs en fonction de l'opération envoyée via le formulaire (addition ou soustraction).
- **Description** :
  - Lorsqu'un nombre est soumis via le formulaire, il est ajouté à la session sous la clé 'nombre'.
  - Si l'opérateur choisi est l'addition (+), tous les nombres dans la session sont additionnés.
  - Si l'opérateur choisi est la soustraction (-), le premier nombre de la session est utilisé comme base, et les autres nombres sont soustraits successivement.
  - Le résultat de l'opération est ensuite affiché via echo.

```
public function operation()
{
    Session();
    session()->push('nombre', request('nombre'));
    $scompt = 0;
    if (request("nombre") == "+")
    {
        $stab = session()->get('nombre');

        foreach ($stab as $value => $key){
            $scompt += intval($key);
        }

        echo "Le Resultat est : ".$scompt;
    }

    else if (request("nombre") == "-") {
        $stab = session()->get('nombre');
        $scompt = $stab[0];
        var_dump($stab);
        for ($i = 1; $i <= count($stab)-1; $i++) {

            $scompt -= intval($stab[$i]);
        }
        echo "Le Resultat est = : ".$scompt;
    }

    /*
    else if (request("nombre") == "*")
    {
        $stab = session()->get('nombre');
        $scompt = 1;
        foreach ($stab as $value => $key){
            $scompt *= intval($key);
        }
    }*/

    return view('welcome');

    //session()->flush();
}
```

## Conclusion

Le contrôleur **NombreController** gère la logique métier des calculs et communique avec la session pour stocker les nombres à calculer. Il effectue les calculs d'addition ou de soustraction et affiche le résultat.

### 3. Vue (Formulaire de saisie)

#### Introduction

Ce code HTML crée un formulaire qui permet à l'utilisateur de soumettre des nombres à calculer en notation polonaise. Il interagit avec la route **POST /test** pour envoyer les données au contrôleur et effectuer les calculs.

#### Structure du formulaire

##### 3.1 Balise <form>

- **Action** : Le formulaire envoie les données à la route /test via la méthode **POST**.
- **Sécurité CSRF** : Le token `{{ csrf_field() }}` protège contre les attaques CSRF en incluant un champ caché avec un token unique pour chaque formulaire.
- 

##### 3.2 Champ de saisie

- **Label** : Le label indique à l'utilisateur qu'il doit entrer des nombres à calculer en notation polonaise.
- **Champ input** : Ce champ permet à l'utilisateur d'entrer un nombre qui sera traité par le contrôleur.

##### 3.3 Boutons

- **Bouton Envoyer** : Ce bouton permet d'envoyer les données au contrôleur. Le bouton est étiqueté "Envoyer" et déclenche l'envoi des données sans effectuer de calcul immédiat.
- **Bouton Calculer** : Ce bouton permet de lancer l'opération de calcul une fois que les données ont été envoyées au serveur.

#### Conclusion

Le formulaire HTML permet à l'utilisateur d'entrer des nombres à calculer et de soumettre ces données au contrôleur via la route **POST /test**. Les boutons permettent d'envoyer les données ou de lancer un calcul.

#### Conclusion générale

Ce projet Laravel permet à l'utilisateur de saisir des nombres et de réaliser des calculs simples (addition et soustraction) via un formulaire interactif. Le tout est géré par des routes et un contrôleur qui communiquent avec la session pour effectuer les calculs en utilisant la notation polonaise.