

Documentation

Installation

Télécharger le dossier *framework* et vous le mettez dans votre dossier *www* sous le nom que vous voulez.

Structure du framework

- `config/` : dossier contenant la configuration des routes et de la base de données
- `src/` : dossier contenant le code source de l'application/site
- `src/App/` : dossier avec le code spécifique de l'application (découpé en MVC)
- `src/Library` : dossier contenant le code spécifique au framework
- `helpers.php` : contient quelques fonctions utilitaires
- `index.php` : moteur du framework

Les fichiers pouvant être modifiés sont ceux se trouvant dans les dossiers `config/` et `src/App/`. Tous les autres fichiers **ne doivent pas être modifiés**.

Créer une page

1. Créer une route dans le fichiers *config/routes/php*
2. Créer un contrôleur
3. Créer une vue

Créer une nouvelle route

Le fichier des routes ressemble à ceci :

```
return [  
    '/' => [  
        'App\Controllers\HomeController',  
        'index'  
    ],  
    '/ma-route' => [  
        'App\Controllers\NomDuController',  
        'methode'  
    ],  
];
```

Pour créer une nouvelle route, il faut ajouter un élément dans ce tableau. La clé représente le chemin (la partie de l'url qui se trouve derrière *index.php*) et sera associée à un tableau avec le nom **complet** du contrôleur et la méthode à appeler.

Créer un contrôleur

Pour créer un nouveau contrôleur, créer un nouveau fichier dans le dossier *src/App/Controllers*.

```
namespace App\Controllers;  
  
use Library\Core\AbstractController;
```

```
class HomeController extends AbstractController
{
    public function index(): void
    {
        // Dans le template homepage.phtml, les variables $name et $age vont exister
        // et contenir les valeurs 'Toto' et 42
        $this->display('homepage', [
            'name' => 'Toto',
            'age' => 42
        ]);
    }
}
```

Cette classe doit hériter de la classe *AbstractController* du framework. La méthode *display* de l'*AbstractController* permet d'afficher une vue. Le premier paramètre représente le nom du template et le second paramètre les données à passer à cette vue.

Créer une vue

Template

Pour créer une nouvelle vue, créer un nouveau fichier dans le dossier *src/App/Views* au format ".phtml". Ce fichier ne contiendra que le code spécifique de la page.

Layout

Il faut également mettre à jour le fichier *layout.phtml* contenant tout le html commun à toutes les pages.

NB : Attention de ne pas supprimer ce fichier qui est utilisé dans le framework.

Génération des url

Dans vos liens, Il faut utiliser la fonction *url* qui permet de générer l'url vers une route spécifique.

```
<a class="nav-link active" aria-current="page" href="<?= url('/') ?>">Accueil</a>
```

Base de données

Configuration

Configurer la base de données dans le fichier *config/database.php*.

Créer un modèle

Les modèles contiennent les requêtes SQL de votre application. Généralement on crée un modèle par table ("PostModel", "UserModel"...).

Les fichiers modèles doivent être créés dans le dossier *src/App/Models*.

```
namespace App\Models;

use Library\Core\AbstractModel;

class PostModel extends AbstractModel
```

```

{
    public function findAll(): array
    {
        return $this->db->getResults(
            'SELECT id, title, content, created_at
            FROM posts
            ORDER BY created_at DESC'
        );
    }

    public function find(int $id): array
    {
        return $this->db->getResults(
            'SELECT id, title, content, created_at
            FROM posts
            WHERE id = :id
            ORDER BY created_at DESC', [
                'id' => $id
            ]
        );
    }
}

```

Le modèle doit hériter de la classe *AbstractModel* du framework. Ce modèle contient une propriété *db* qui est une instance de *Connection* avec laquelle les requêtes SQL pourront être effectuées.

Récupérer des données

La méthode *getResults* de la classe *Connection* permet d'effectuer une requête SQL et de récupérer les résultats (cf exemple ci-dessus).

Exécuter une requête SQL autre que select

La méthode *execute* de la classe *Connection* permet d'effectuer une requête SQL de type insert, update ou delete.

```

public function create(string $title, string $content): void
{
    return $this->db->execute('INSERT INTO posts(title, content) VALUES(:title,
:content)', [
        'title' => $title,
        'content' => $content
    ]);
}

```