

La mise en place

Afin de démarrer un projet PHP. Il faut tout d'abord créer notre dossier et mettre en place composer.

```
mkdir NomDuProjet
cd NomDuProjet
composer init
code .
```

Maintenant il faut versionner le projet et le mettre sur github.

1. Initialiser le dépôt : En cliquant sur l'icone contrôle de source, puis initialiser le dépôt.
2. Créer un fichier `.gitignore` à la racine du projet et rajouter le contenu suivant :

```
vendor
```

3. Réaliser un commit : Toujours sur l'onglet contrôle de source, spécifier un message de commit puis cliquer sur la validation
4. Publier les branches sur github (en cliquant sur Publier les branches). Tout au long de cette étape, une fenêtre de navigateur github vas s'ouvrir afin d'obtenir les permissions.

À la fin de la démarche votre projet devrait disponible et en ligne sur github.

Configuration de composer

Petit rappel : Composer est un outil qui permet de mettre en place l'arborescence d'un projet et d'installer des possible librairies.

Attention, il doit être configuré, pour cela il faut ouvrir le fichier `composer.json` et éditer la ligne : autoload - psr-4 :

```
{
  // ...
  "App\\": "src/"
  // ...
}
```

Il faut lancer la commande : `composer install` afin de mettre en place composer

Vous pouvez faire un commit et publier sur github

Lancer un serveur de développement

Un site professionnel en php contient une seule page php : **Le frontend controller**.

Ce script php est le point de départ de notre site internet, il contiendra **TOUT LE CODE NÉCESSAIRE** au bon fonctionnement de notre site.

1. Créer un fichier dans le dossier `public` nommé : `index.php`
2. Placer le code suivant :

```
<?php

var_dump($_SERVER);
```

3. Lancer un serveur de développement sur notre « frontend controller » avec la commande :

```
php -S localhost:11001 -t public public/index.php
```

Maintenant vous pouvez accéder à votre site internet sur l'adresse : <http://localhost:11001>

IMPORTANT

La commande `php -S` lance un serveur de développement. Vous ne pouvez plus rentrer de commande à l'intérieur du terminal qui vient de lancer le serveur. Pour arrêter un serveur il faut appuyer sur la combinaison de touche CTRL-C

Servir nos fichiers « statique »

Afin de servir nos fichier statique (les fichiers présents dans le dossier `public`), il faut demander à notre « frontend controller » d'analyser l'URI :

si L'URI contient une extension de fichier alors le script php s'arrête et laisse la place au fichier.

Placer dans le fichier `public/index.php` le code suivant

```
<?php

// On test si l'on demande un fichier se terminant par les extensions suivante
if (preg_match("/(\\.css|\\.png|\\.jpeg|\\.js|\\.gif|\\.pdf|\\.mp3|\\.mp4|\\.html|\\.jpg)$/i",
$_SERVER['REQUEST_URI'])) {
    // On demande à PHP de ne pas se lancer. Ainsi ce sera le fichier
    // demandé qui sera retourné dans notre navigateur
    return false;
}

var_dump($_SERVER);
```

Créer le fichier `public/style.css` avec le code suivant :

```
body,
html {
    margin: 0;
    padding: 0;
}
```

Vous pouvez tester on vous rendant sur la page `/style.css`.

BONUS

Pour ceux qui le désire il existe de très très bon tutoriel sur les expression régulière :

[Celui ci par exemple](#)

Arborescence du projet

- Un fichier `composer.json` et `composer.lock` : Se sont les fichier de configuration de composer
- Un dossier `public` : Contient notre site internet (le frontend controller, ainsi que tout les fichiers ncessaire au bon fonctionnement de notre site).
- Un dossier `src` : Contient toutes nos classes
- Un dossier `vendor` : Contient l'autoload et les librairies composer

Pour commencer à utiliser les classes de notre projet nous devons inclure l'autoload de composer :

Dans le fichier `public/index.php`

```
<?php

require __DIR__ . '/../vendor/autoload.php';

// ...
```