Symfony

Comme indiqué sur la page d'accueil, Symfony est avant tout **un ensemble de composants PHP réutilisables**.

Le framework Symfony en lui-même vient rassembler plusieurs dizaines de ces composants, dans une structure (arborescence) précise. On construit ensuite notre application dans cette structure.

Versions

Symfony adopte également le système de versioning `semver`, et présente une nouvelle version majeure tous les 2 ans.

Pendant ces 2 années, on aura 5 versions mineures : de 0 à 4.

La version mineure n°4 sera donc la dernière sous-version d'une version majeure (3.4, 4.4, etc...), et elle sortira en même temps que la version majeure suivante.

Ainsi, les versions 3.4, 4.4, etc...sont appelées des versions **LTS** ou Long-Term Support : un support sur la correction de bug et de failles de sécurité est assuré sur ces versions pendant 3 ans pour les corrections de bugs et 4 ans pour les failles de sécurité.

Installation en version 5.4 LTS

Pour installer un projet Symfony avec la version 5.4, donc la LTS, on va utiliser le binaire de Symfony (ou Symfony CLI) :

```
symfony new my_project_directory --version=5.4 --webapp
```

> **REMARQUE**
>
> On peut créer 2 types d'application Symfony : des applications destinées à la création de microservices, API, ou applications console par exemple, et des applications web complètes, incluant un moteur de template (Twig). Dans ce cas, on ajoute l'option `--webapp` à

> l'instruction de ligne de commande permettant de créer l'application. La base d'installation reste la même (package Composer `symfony/skeleton`)

A l'exécution de cette commande, on verra défiler les différents composants issus de packagist.org, qui sont intégrés automatiquement par le framework dans nos dépendances.

Environnements

Dans une application Symfony, l'environnement par défaut dans lequel nous allons travailler est `dev`, par défaut.

L'environnement se présentera sous forme d'une variable d'environnement `APP_ENV`.

Il y a 3 environnements prévus par Symfony par défaut : `dev`, pour la phase de développement, `test` pour les tests unitaires, fonctionnels, etc... et `prod` pour le déploiement en production, qui nous permet d'optimiser la configuration de l'application pour qu'elle soit plus rapide.