

Symfony

Créer un nouveau projet Symfony avec Laragon

Après avoir démarré Laragon, cliquer dans le menu : créer un site web rapidement → Symfony.

Nommer le dossier. Laragon lance alors l'invite de commande et installe symfony/website-skeleton. Mieux vaut être patient, ça prend un certain temps.

Quand c'est fini, on a les messages suivants :

* Run your application:

1. Go to the project directory
2. Create your code repository with the git init command
3. Download the Symfony CLI at <https://symfony.com/download> to install a development web server

* Read the documentation at <https://symfony.com/doc>

C'est une bonne idée d'initialiser un dépôt git, c'est sûr.

Pour le 3, c'est juste pas possible parce que le serveur de développement fait maintenant partie du package de base symfony, sauf quand on installe avec Laragon. A la place, on peut utiliser le serveur apache de laragon (de toute façon il tourne) en installant un hôte virtuel.

C'est pas compliqué, dans le dossier du projet, taper l'instruction suivante dans le terminal de commande :

```
composer require symfony/apache-pack
```

On a aussi :

* You're ready to send emails.

* If you want to send emails via a supported email provider, install the corresponding bridge.

For instance, composer require mailgun-mailer for Mailgun.

* If you want to send emails asynchronously:

1. Install the messenger component by running `composer require messenger`;
2. Add 'Symfony\Component\Mailer\Messenger\SendEmailMessage': amqp to the

config/packages/messenger.yaml file under framework.messenger.routing
and replace amqp with your transport name of choice.

* Read the documentation at <https://symfony.com/doc/master/mailer.html>

J'ai pas encore creusé ça.

Et pour finir :

* Modify your DATABASE_URL config in .env

* Configure the driver (mysql) and
server_version (5.7) in config/packages/doctrine.yaml

* Write test cases in the tests/ folder

* Run php bin/phpunit

***** NOTE: Now, you can use pretty url for your awesome project :) *****

(Laragon) Project path: C:/laragon/www/nom_projet

(Laragon) Pretty url: http://nom_projet.test

Et on autorise le terminal de commande à apporter des modifications, afin de recharger le serveur apache.

Les trucs sympas à installer en plus :

composer require annotations

Pour pouvoir mettre les routes directement dans le contrôleur :

```
class LuckyController extends AbstractController
{
    /**
     * @Route("/Lucky/number")
     */
    public function number(): Response
    {
```

composer require frameworkbundle ? Normalement il y est, ça m'est arrivé une fois qu'il n'y soit pas, pour avoir la classe AbstractController.

```
composer require symfony/maker-bundle --dev
Pour créer des commandes, des contrôleurs, des entités (doctrine entity), même
des crud...
```

exemple :

```
php bin/console make:controller BrandNewController
```

```
created: src/Controller/BrandNewController.php
```

```
created: templates/brandnew/index.html.twig
```