Symphony Memo

# Install

<http://symfony.com/download>

Tout est expliqué et clair

# Commandes utiles et importantes :

Une fois le fichier symfony téléchargé au dessus placé dans le dossier de nos projets :

Créer un nouveau projet :

Matthieu@Matthieu-PC MINGW64 /c/wamp/www

$ php symfony new Nom\_du\_Projet

Lancer le serveur de Symfony pour commencer à tester le website :

Matthieu@Matthieu-PC MINGW64 /c/wamp/www/stop\_gaspi

$ php bin/console server:run

# Notes to self :

## Note 1

Un href vers une action d’un controller doit impérativement être composé de son « path » (Précisé dans le controller en comm au dessus de l’action appelée.

*/\*\*  
 \** ***@Route****("/entity-remove", name="delete\_route\_name")  
 \*  
 \*/*

Ici au dessus de l’action appellee dans le controller

<a href=”{{ path(‘delete\_route\_name’) }}”>Appel Lucky Controller</a>

Ici dans le HTML pour faire le lien

La route configurée dans le comm., ici « /entity-remove » n’est utilisée qu’en URL, on ne peut la passer dans le HREF

## Note 2

L’ajout d’une wildcard à la route (paramètre entre {}) au dessus de l’action du controller force a avoir un paramètre dans l’url sinon exception générée.

## Note 3

Chose que l’on utilisera beaucoup :

Fonction de redirection avec des paramètres à envoyer au template :

// renders app/Resources/views/hello/greetings/index.html.twig

return $this->render('hello/greetings/index.html.twig', array(

'name' => $name

));

## Note 4

Création des entités et de l’ORM  à partir de notre base stop\_gaspi existante :

Une fois qu’on a bien configuré notre base dans le fichier de config : app/config/parameters.yml

On execute les commandes suivantes pour créer toutes les entités dont nous avons besoin à partir de notre base :

$ php bin/console doctrine:mapping:import --force AppBundle xml

$ php bin/console doctrine:mapping:convert annotation ./src

La commande ci-dessous génère les Getters & Setters pour nos entités !

$ php bin/console doctrine:generate:entities AppBundle

notre base :

**Note 5**

Une fois qu’on a fait tout ça, on peut créer la vue et controller associée à une entité avec la commande suivante : Par la suite de cette commande on choisit l’entité à « CRUDER », si on veut ajouter un droit d’écriture sur cette entité, etc…

Matthieu@Matthieu-PC MINGW64 /c/wamp/www/stop\_gaspi

$ php bin/console generate:doctrine:crud