|  |
| --- |
| Document de passation |
| Atelier Redac BD |
| Arthur Bosquette, Victor Carmouze, Mickael Martinez  Version 1.0  10 Pages  27/04/2013  PLIC-BD-COLLABORATIF-SYMFONY-DESSIN-TEXTE-WEB-NODEJS-SOCKET.IO |



Propriétés du document

|  |  |
| --- | --- |
| Auteur | Arthur Bosquette, Victor Carmouze, Mickael Martinez |
| Version | 1.0 |
| Nombre de pages |  |
| Références | PLIC-BD-COLLABORATIF-SYMFONY-DESSIN-TEXTE-WEB-NODEJS-SOCKET.IO |

Historique du document

|  |  |  |  |
| --- | --- | --- | --- |
| Date de révision | Version | Auteur | Changements |
| 26/04/2013 | 1.0 | Groupe DansTaBulle | Version initiale du document |

Source du document

Sommaire

[Symfony 4](#_Toc354826401)

[Installation 4](#_Toc354826402)

[Prérequis 4](#_Toc354826403)

[Dépendances 5](#_Toc354826404)

[Dépôt 6](#_Toc354826405)

[Développement 7](#_Toc354826406)

[Collaboratif 8](#_Toc354826407)

[1) Pour NodeJS 8](#_Toc354826408)

[2) Pour Express 8](#_Toc354826409)

[3) Collaboratif 9](#_Toc354826410)

[4) Base de données 9](#_Toc354826411)

[5) Editeur de Texte 9](#_Toc354826412)

[6) Dessin 10](#_Toc354826413)

# Symfony

## Installation

### Prérequis

Vous devez installer, sur votre machine de développement, les services suivants :

* Un serveur Web (Apache, NGinx)
* PHP
* Un système de gestion de base de données (MySQL, PostGreSQL)
* Git, pour la gestion des dépendances

Voici comment installer un serveur web classique sur votre machine de dev (Apache, PHP, MySQL).

#### Windows

Des logiciels sont là pour vous simplifier la vie, nous avons utilisé wampserver pour Windows.

<http://www.wampserver.com/>

Ce logiciel vous installera Apache 2, PHP 5.x (la dernière version stable), MySQL, PHPMyAdmin (permet de gérer vos bases de données), xDebug (utile pour le debug de scripts PHP) et un peu plus selon la version que vous avez choisi.

Pour Git, il vous faut installer msysgit.

<https://code.google.com/p/msysgit/>

<https://help.github.com/articles/set-up-git>

#### Mac OS X

Téléchargez cette fois-ci MAMP. (Pas la peine d’aller chercher la version PRO, la version de base suffit).

<http://www.mamp.info/en/index.html>

Pour Git, installer le logiciel de GitHub vous permet d’avoir accès à git via la console.

<https://help.github.com/articles/set-up-git>

#### Linux

Beaucoup de paquets sont à installer, je vous laisse faire une petite recherche. N’oubliez pas d’installer toutes les extensions PHP qui vous seront nécessaires.

### Dépendances

Il faut à présent télécharger Symfony.

<http://symfony.com/download>

Prenez la version sans Vendors, ce qui nous permettra de gérer toutes les dépendances de notre système. Les Vendors sont les composants de Symfony, sans eux, Symfony ne peut marcher. C’est pour cela qu’il nous faut un système pour les gérer efficacement.

#### Installer Composer

Composer est un outil pour gérer les dépendances en PHP. Les dépendances, dans un projet, ce sont toutes les bibliothèques dont votre projet dépend pour fonctionner.

Installer Composer est très facile, il suffit d'une seule commande… PHP ! Exécutez cette commande dans la console :

php -r "eval('?>'.file\_get\_contents('http://getcomposer.org/installer'));"

Cette commande va télécharger et exécuter le fichier PHP suivant : <http://getcomposer.org/installer>

Il suffit ensuite d’exécuter la commande suivante pour mettre à jour Symfony et télécharger les bundles manquants/non à jours.

php composer.phar update

Cette commande va lire dans le composer.json à la racine de votre installation Symfony.

#### Cette partie est peu claire ? Il existe des tutoriels détaillant comment composer marche, son utilité, et comment en tirer profit facilement.

* <http://getcomposer.org/doc/00-intro.md>
* <http://symfony.com/fr/doc/2.1/book/installation.html>

### Dépôt

Voici le .gitignore que nous utilisons pour notre dépôt.

# Mac Files

.DS\_Store

# Bootstrap

Symfony/app/bootstrap\*

# Symfony directories

Symfony/vendor/\*

Symfony/logs/\*

Symfony/cache/\*

Symfony/web/uploads/\*

Symfony/web/bundles/\*

Symfony/app/cache/\*

Symfony/app/logs/\*

#Config Files

Symfony/app/config/parameters.yml

#### Besoin de plus de précisions ?

<http://symfony.com/doc/2.0/cookbook/workflow/new_project_git.html>

## Développement

La majorité des informations essentielles au développement sur symfony est présente dans les liens suivants :

* <http://www.siteduzero.com/informatique/tutoriels/developpez-votre-site-web-avec-le-framework-symfony2>
* <http://symfony.com/doc/current/book/index.html>

Ces deux sites couvrant la majorité du développement sous Symfony, il n’est pas nécessaire de s’attarder longtemps dessus. Néanmoins, avant de commencer votre site, consulter les sites suivants :

#### Espace Membre

Ici, le bundle FOS User Bundle vous simplifiera la vie :

* <https://github.com/FriendsOfSymfony/FOSUserBundle>
* <https://github.com/FriendsOfSymfony/FOSUserBundle/tree/master/Resources/doc>

A gérer bien sûr avec composer ;-)

#### Upload de fichiers

* <http://symfony.com/fr/doc/master/cookbook/doctrine/file_uploads.html>

#### Utilisation de PostGreSQL

Certains problèmes peuvent apparaître avec l’utilisation de PostGreSQL. Faites attention aux mots clés de PostGreSQL. Si vous avez des soucis avec des types de données, référez vous à :

* <http://rchea.com/symfony-2-1-et-postgresql/>
* <http://www.d-sites.com/2011/12/03/symfony-2-et-postgresql-mots-reserves/>

#### Design

Pour vous faciliter la vie, utiliser bootstrap pour définir un style de base à votre site. Rien ne vous empêche de surcharger les définitions de celui-ci par la suite.

<http://twitter.github.io/bootstrap/>

#### Gestion des Thumbnails

Avalanche Imagine Bundle vous permet de créer des miniatures de vos images et de les stocker en cache.

* <https://github.com/avalanche123/AvalancheImagineBundle>

# Collaboratif

Nous avons choisi d’utiliser NodeJS car c’est une solution performante qui nous a permis de gérer plus facilement le côté collaboratif et instantané de l’application, notamment grâce à Socket.IO.

## 1) Pour NodeJS

- Télécharger et installer le NodeJS correspondant à votre OS sur le site officiel.

- NPM, le gestionnaire de modules, est normalement inclus dans l’installation. Il permet d’installer facilement un module dans une application, via la ligne de commandes et l’instruction « npm install nomDuModule ». Il suffit de la lancer depuis le répertoire principal du projet.

- Il existe un certain nombre de ressources permettant de réaliser ses premiers pas sur nodeJS, nous allons en donner quelques exemples. Cependant, il est important de faire attention car la plateforme évolue vite et tout n’est pas forcément compatible d’une version à l’autre.

<http://www.nodebeginner.org/>

<http://blog.developpez.com/philippe/p10308/node-js/node_js_se_former_et_developper_gratuite>

<http://www.laurentkempe.com/post/Writing-a-first-Nodejs-application-on-Windows.aspx>

Le deuxième lien contient lui-même quelques autres liens intéressants.

- Il existe un IDE pratique pour le développement, Cloud9.

- Le module Nodemon permet de relancer automatiquement l’application serveur à chaque modification de son code source. C’est très pratique pour éviter d’avoir à le faire à la main.

## 2) Pour Express

- Express est un framework pour NodeJs qui permet la création de sites web.

- Il est possible de configurer un package.json pour indiquer les dépendances et installer les versions souhaitées de tous les modules avec un simple « npm install » dans le répertoire du projet.

Il aura cette forme :

{

"name": "application-name",

"version": "0.0.1",

"private": true,

"scripts": {

"start": "node app.js"

},

"dependencies": {

"express": "3.1.1",

"jade": "\*",

"socket.io": "\*"

}

}

- Attention, un certain nombre de ressources sur le web proposent des exemples de code valables pour la version 2.x, or l’actuelle est la 3.x et tout n’est malheureusement pas compatible.

Pour aider à migrer le code vers la version 3, on peut se tourner vers :

<https://github.com/visionmedia/express/wiki/Migrating-from-2.x-to-3.x>

- Si vous n’avez pas envie d’utiliser le moteur de template par défaut Jade (qui n’est a priori pas très intuitif, mieux vaut utiliser EJS), il est possible d’écrire dans le page.jade un « import page.html ».

- En tous cas, voilà un exemple de source pour apprendre à utiliser Express :

<http://code4fun.fr/creer-un-site-simple-avec-node-js-express-et-jade/>

## 3) Collaboratif

- Le moyen le plus efficace de gérer le collaboratif avec NodeJS est de passer par le fameux Socket.IO qui permet de gérer les évènements en temps réel de façon asynchrone.

- Voilà quelques ressources simples pour apprendre à l’utiliser :

<http://www.atinux.fr/2011/08/28/tutoriel-socket-io-debutant/>

<http://www.web-tambouille.fr/2011/03/8/node-js-partie-3-premiere-application-node-js-et-html5-express-socket-io.html>

- Il est également possible d’utiliser un système de session :

<http://naholyr.fr/2013/01/authentification-websocket-sessions-express-3-socket-io/>

## 4) Base de données

- Si vous utilisez postgreSQL, il existe ce module permettant de l’utiliser avec nodeJS :

<https://github.com/brianc/node-postgres>

Et ce lien où l’utilisation est explicitée par la première réponse :

<http://stackoverflow.com/questions/9205496/play-with-postgres-through-node-js>

## 5) Editeur de Texte

- Pour le cas où, comme nous, vous décideriez de ne recoder que la partie collaborative de l’éditeur pour ne pas perdre de temps, il existe déjà un certain nombre de solutions open sources.

Nous avons choisi CKEditor pour son grande nombre de plugins, ses possibilités avancées de configuration et son aspect graphique acceptable et facilement intégrable dans notre design.

<http://ckeditor.com>

- Nous avons choisi d’envoyer au serveur et de broadcaster aux autres sockets chaque modification du texte par un utilisateur, sans système de lock, mais nous ne le sauvegardons en base que lors de l’enregistrement (le module save a été réécrit par nos soins) et lorsque l’éditeur perd le focus.

- Pour garder une trace de quelle socket correspond à quelle utilisateur, notamment pour l’utiliser lors de la déconnection, nous gardons un tableau avec un objet {socketId, userName}.

## 6) Dessin

En ce qui concerne le dessin collaboratif, voilà quelques ressources intéressantes :

- Upload d’image en HTML5 :  <http://html5demos.com/dnd-upload>

- Résolution de quelques problèmes liés à socket.IO :

<http://stackoverflow.com/questions/10191048/socket-io-js-not-found>

- Apprendre à dessiner avec des canvas :

<http://www.vincent-rousseau.net/content/mini-paint-html5-avec-canvas>