

Informatique de gestion

645-2 Project

**Health Tourism**

Etudiants : Siméon Bobylev, Vincent Huck, Dany Maillard

Professeur : Alexandre Cotting

Rendu, le : 05 juillet 2015

Avant-propos

Contexte

Objectifs

Table des matières

[Liste des figures iv](#_Toc426546963)

[1. Outils et technologies 1](#_Toc426546964)

[1.1 GitHub 1](#_Toc426546965)

[1.2 Symfony 1](#_Toc426546966)

[1.3 JetBrains PhpStorm 2](#_Toc426546967)

[1.4 Composer 2](#_Toc426546968)

[1.5 WampServer 3](#_Toc426546969)

[1.6 Android Studio 3](#_Toc426546970)

[1.7 Microsoft Azure 3](#_Toc426546971)

[2. Implémentation 3](#_Toc426546972)

[2.1 Environnement de développement 3](#_Toc426546973)

# Liste des figures

[Figure 1 – Dépendances dans composer.json 2](#_Toc426543256)

# Outils et technologies

## GitHub

L’utilisation de GitHub pour ce projet nous a permis de bénéficier des avantages suivants :

* Travail sur le même projet à plusieurs simplifié
* Gestionnaire de version (Git)
* Répartition des tâches via les Issues
* Code review pour chaque commit

Notre projet sur GitHub : <https://github.com/ZHB/HEVS-Android-tourism>

Il existe un outil simple qui permet gérer les différentes versions de code entre GitHub et votre poste sans connaître toutes les commandes spécifiques, il s’agit de [GitHub Windows](https://windows.github.com/). Très simple à prendre en main, il permet à n’importe qui de travailler à plusieurs sur un projet publié sur GitHub.

La plupart des IDE (NetBeans, PhpStorm, etc.) peuvent être configurés avec Git mais nécessitent tout de même quelques connaissance de base.

Pour les plus avancés, il existe aussi la version [BASH](https://msysgit.github.io/) pour Windows. Cette dernière offre beaucoup plus de possibilités, surtout lorsqu’il s’agit de gérer les conflits en différentes versions mais nécessite par conséquent une connaissance accrue du produit.

## Symfony

Nous avons choisi d’utiliser un framework PHP, nommé [Symfony](https://symfony.com/), pour le développement de la partie administrative et ceci pour les raisons suivantes :

* Utilisé par de nombreux professionnels
* Documentation complète et bien rédigée
* Comporte de nombreux bundles et librairies
* Temps de développement optimisé (mais courbe d’apprentissage plus longue)

Pour débuter avec le framework, nous vous conseillons de commencer par lire [The Symfony Book](https://symfony.com/doc/current/book/index.html). Les premiers chapitres, qui traitent des notions fondamentales, ainsi que les chapitres Controller, Routing, Templates et Databases and Doctrines sont primordiaux.

Pour ceux qui connaissent déjà le framework peuvent aller plus loin en lisant [The Symfony Cookbook](https://symfony.com/doc/current/cookbook/index.html) et [Symfony Best Practices](https://symfony.com/doc/current/best_practices/index.html).

## JetBrains PhpStorm

Nous avons choisi d’utiliser [PhpStorm](https://www.jetbrains.com/phpstorm/) comme IDE pour la partie Symfony pour les raisons ci-dessous :

* Payant mais gratuit pour les étudiants (création du compte avec @students.hevs.ch)
* Commandes pour Composer et Symfony
* Plugin Symfony (auto complétion)
* Convivial

## Composer

[Composer](https://getcomposer.org/) est un outil qui permet de gérer les dépendances (librairies) d’un projet Symfony ou sein de n’importe quel autre projet PHP. Aujourd’hui, il fait partie des incontournables dans le monde du développement PHP.

Nous vous conseillons de lire la documentation du site officiel qui explique bien comment installer et utiliser cet outil.

Exemple des dépendances dans le fichier composer.json :

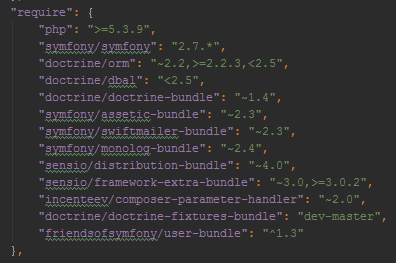


Figure 1 – Dépendances dans composer.json

Chaque ligne veut dire que la librairie en question est inclue dans tout le projet.

Un autre avantage, en plus de la gestion des dépendances, est le fait qu’il n’y a plus besoin de se transmettre la totalité des librairies entre les divers développeurs. Il suffit d’avoir les 2 fichiers : composer.lock et le fameux composer.json, ensuite à l’aide de la commande *composer install*, l’outil de va se débrouiller pour récupérer toutes les fichiers et inclure les librairies dans votre projet. Pour cela, il faut bien évidemment être connecté à Internet.

## WampServer

[WampServer](http://www.wampserver.com/) est un logiciel qui permet de simuler un serveur web sur sa machine locale. WAMP est acronyme signifiant Windows Apache MySQL PHP, le tout centralisé dans un seul et unique logiciel à installer sur sa machine de développement.

Les sites web doivent se trouver dans le répertoire par défaut *C:\wamp\www*.

## Android Studio

<https://developer.android.com/sdk/index.html>

## Microsoft Azure

<http://azure.microsoft.com/fr-fr/>

# Implémentation

## Environnement de développement

Installer Wamp

Installer PhpStorm

Installer Composer

Installer GitHub Windows

Récupérer le projet depuis GitHub

Installer les dépendances via Composer