UNIVERSITE D’EVRY – VAL D’ESSONNE

Département d’informatique



# Brady ABDEREMANE

le 16 janvier 2017

DOCUMENTATION TECHNICO-FONCTIONNELLE SUR LE SITE LIBRECOACH

*Responsable du projet*  Laurent POLIGNY

Année universitaire 2016-2017

Table des matières

Résumé 3

Introduction 4

1. Configuration de l’environnement 5
   1. Téléchargement et installation des outils 5
   2. Outils de développement (optionnel) 5
   3. Exécution du site web 5
2. Fonctionnement général du site 6
   1. Point particulier 6
3. Structure technique du site 7

Résumé

Ce document a pour objectif de fournir à de futurs développeurs une documentation technique détaillée des différentes parties du projet LibreCoach.

La première partie concerne la configuration des environnements nécessaires au bon fonctionnement du web service et le site web. La configuration décrite dans ce document est surtout liée au système d’exploitation Microsoft Windows car la procédure est plus complexe que sur un système basé sur Linux.

La seconde partie contient une documentation de l’API et décrit le fonctionnement du site web. Cette version du document décrit le fonctionnement de la réécriture de la nouvelle version de l’API qui est plus allégée car elle se concentre sur les fonctions liées à la gestion des données.

## Introduction

LibreCoach est un site internet d’entraides permettant de donner un coup de pouce aux clients qui souhaitent obtenir un aide pour accomplir leurs projets. Il permet :

* L’inscription et la connexion sécurisée des clients
* La gestion d’annonce (ajouter, modifier, supprimer)
* Le client propose ces disponibilités
* La connexion du coach selon sa spécialité (travail, santé, professionnel, famille,…)
* La gestion des clients et des coach dans une page administrateur (laissé libre pour que vous puissiez ajouter des coachs).

Sur le site, toutes les annonces sont gérer par les coachs, les coachs ne peuvent visualiser que les annonces liées à sa spécialité. Ensuite le coach propose un rendez-vous aux clients pour parler du projet plus en détail. Si le client accepte le rendez-vous, le coach prend en charger le projet, si le client refuse le coach pourra lui re-proposer un rendez-vous plus tard.

1 Configuration de l’environnement

1.1 Téléchargement et installation des outils

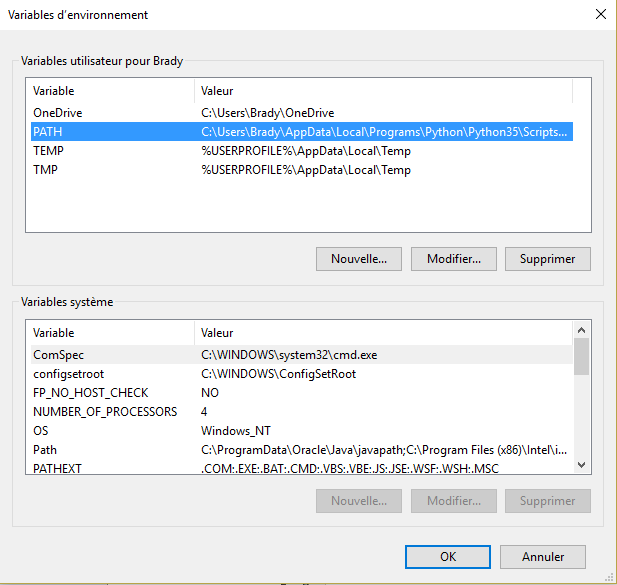
Certains modules de l’API nécessitent d’être recompilés, il faut impérativement installer Python si vous êtes sur Windows ainsi que Python 2.

**• Python :** [**https://www.python.org/downloads**](https://www.python.org/downloads) **- langage de programmation**

**• Django :** [**https://www.djangoproject.com/download/**](https://www.djangoproject.com/download/) **- Framework web de python**

**• Bootstrap : getbootstrap.com/ - Framework front-end**

Remarque : Par la suite, nous aurons besoin d’utiliser Python en ligne de commande. Pour les utilisateurs Windows, il faut modifier les variables d’environnement.



1.2 Outils de développement (optionnel)

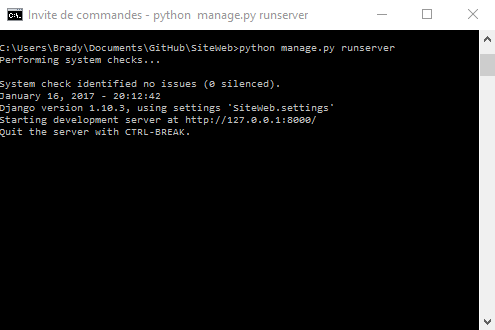
Pour faciliter le développement sur le site web, il est conseillé d’utiliser des modules pour Symfony2 :

• Pour JetBrains PyCharm Community : https://www.jetbrains.com/pycharm/download/ Logiciel de developpement

• Pour Notepad++ : <https://notepad-plus-plus.org/fr/download/v6.6.9.html> Editeur de texte

1.4 Exécution du site web

Avant de lancer le site, il est impotant de démarrer le serveur web de develppement (sur la machine loclae).



2 Fonctionnement général du site

- le client peut s'inscrire ou se connecter(sécurisé)

- le client peut créer une annonce, le modifier ou le supprimer

- le client propose ses disponibilités

- le coach se connecte via un espace sécurisé

- les coachs sont spécialisés selon des rubriques(travail, sante, professionnel,famille,...)

- le coach peut voir toutes les annonces selon sa spécialité uniquement

- le coach propose un rendez-vous par mail( + type de rendez-vous)

- Si le client accepte le rendez-vous, le coach prend en charge le projet (met pris\_en\_charge a true), sinon le coach re-propose un rendez-vous et laisse le pris\_en\_charge a false.

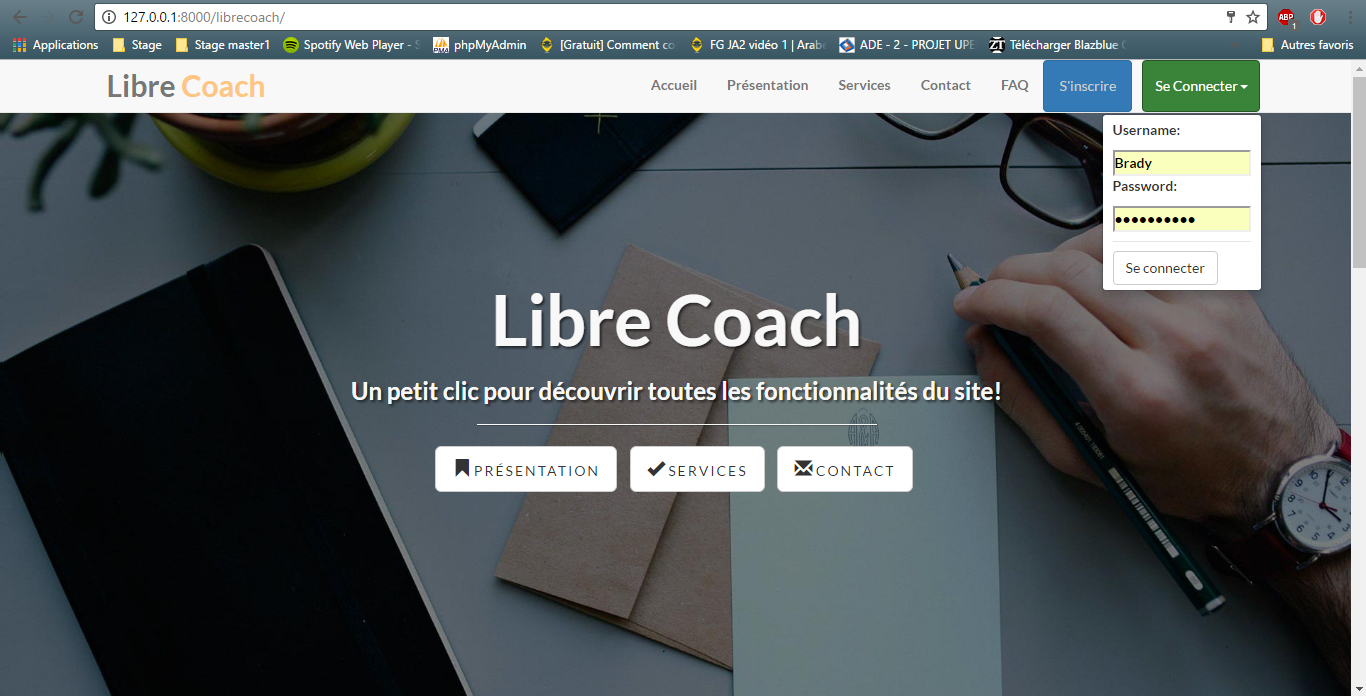
2.1 Point particulier

Noter le site dans la page d’accueil n’est pas encore operationnel.

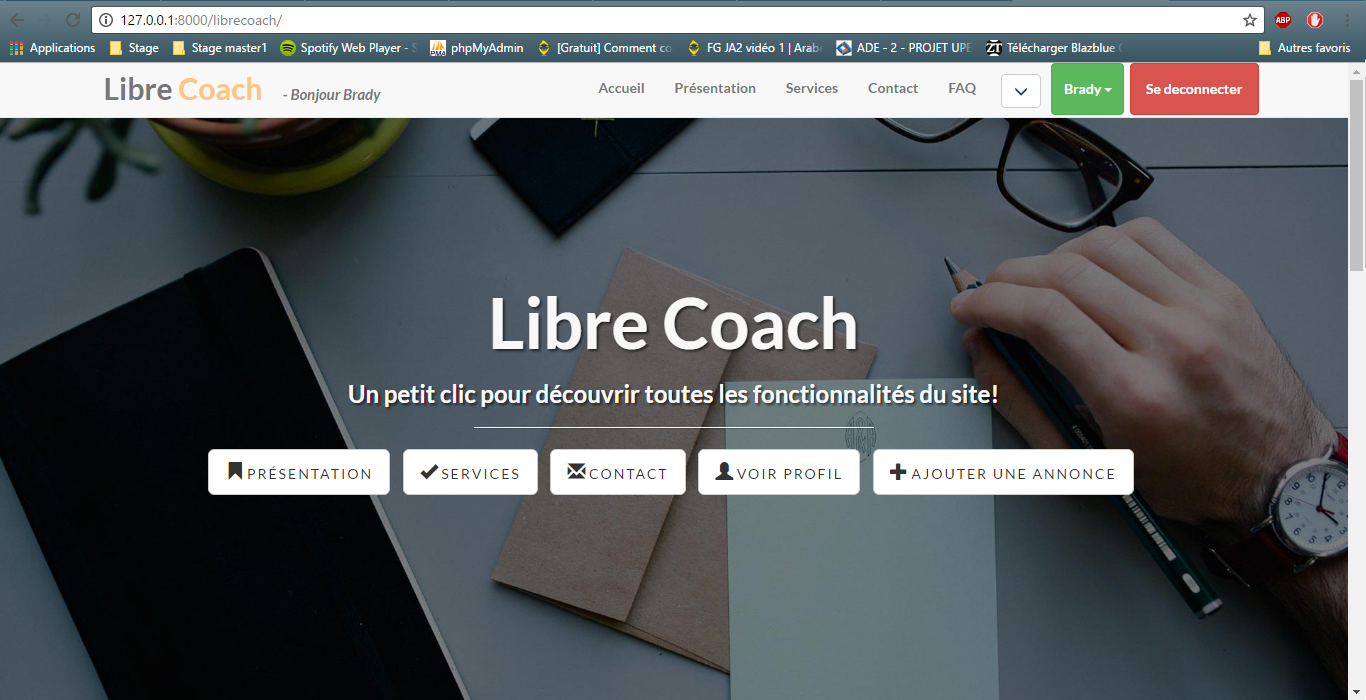
Pas besoin de réécrire le mot de passe pour créer un compte.

La page administration est publique pour créer des coachs mais sera par la suite privée.

Client non connecté :



Client connecté :



3 Structure technique du site

**Structure de la base de données via ORM (object-relationnal mapping)**

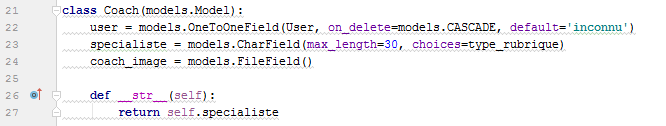
Django propose un système pour bénéficier des avantages d'une base de données SQL sans devoir écrire ne serait-ce qu'une seule requête SQL !

Lorsque vous créez un modèle dans votre application Django, le Framework va automatiquement créer une table adaptée dans la base de données qui permettra d'enregistrer les données relatives au modèle.

Dans notre classe model.py nous avons la class Annonce.

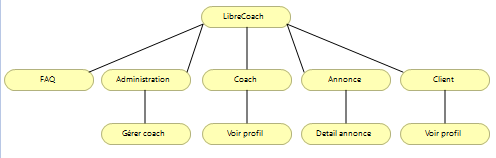


Et la class Coach



**En quelques mots, annonce est relié à user, et rubrique (qui est coach en réalité) et coach est associé à user via(OneToOneField) et la classe User est créée automatiquement via les imports.**

L’arborescence des fichiers (et une description du rôle de chacun : profites-en pour le mettre en commentaire au début de chaque fichier si ça n'est pas déjà fait).



FAQ : TemplateView (page statique)

Administration : Gère les coachs et les utilisateurs (sauf la création)

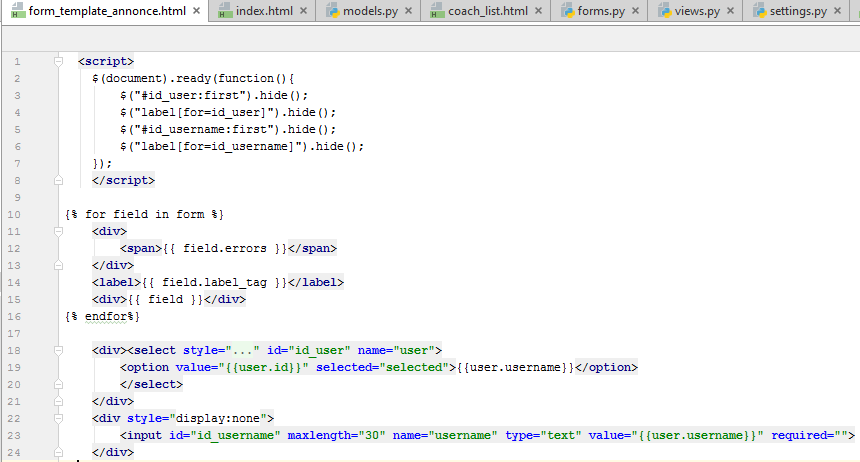
Coach : Gère les annonces lié à sa spécialité + affiche les conformation personnelles

Annonce : créer une annonce

Client : Voir profil + toutes ses annones

**Aspectes techniques particuliers**

Pour la création d’une annonce : nous avons utilisé des vues génériques d’édition (CreateView), néanmoins, pour mettre automatiquement l’annonce au client en question j’ai utilisé une méthode(peu commode) utilisant le JQuery pour cacher le champ «User » en ajoutant manuellement un champ «User » automatiquement remplir au nom du client.



REST API Serializer JSON

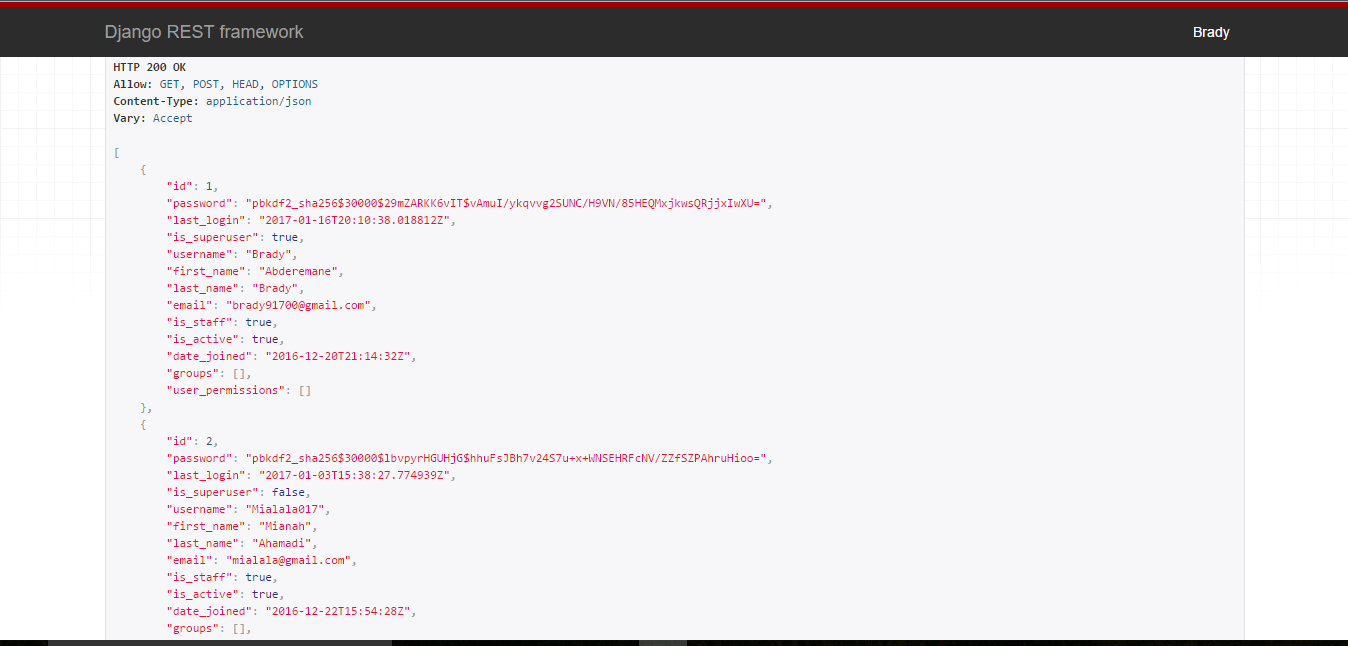
Télécharger REST API : pip install djangorestframework (dans le terminal)

Tout se trouve dans le fichier serializers.py et les fonctions de views.py sont :

- AnnonceList(APIView)

- UserList(APIView)

- CoachList(APIView)



Exemple d’un fichier JSON(Client)

Si par exemple les prochains développeurs souhaitent faire une application IOS, il peut utiliser les fichiers JSON.