ENSEIRB-MATMECA

Filière informatique, $2^{ème}$ année

Projet de Fin d'Année

Documentation

Application hybride pour rediriger les utilisateurs lors des pannes de transport en commun sur Bordeaux Métropole

Pierre ROUX
Romain RAMBAUD
Victor SAINT GUILHEM
Mehdi BOUNAKHLA
Mohammed RIHANI
Nathan REAVAILLE

`Clients` Hassen BENSALEM, Raphaël CHERRIER
`Encadrant` Toufik AHMED

Mars 2016

1 Organisation et forme du code

Ce document, les README.md du serveur et de l'interface utilisateur et finalement le rapport constituent la ducumentation de l'application. Ce document traite de l'organisation et de a forme du code.

1.1 Serveur

Le code du serveur s'organise de la façon suivante :

```
+-- App/ ....................... Répertoire principal du serveur
|   +-- Project/ ................
    | +-- Controller/ .......... fichier de l'application flask
     | +-- login.py ......... fichier de gestion des connection
        |   +-- notification.py .. fichier d'envoie des notification
        |   +--path.py .......... fichier de gestions des trajets utilisateurs
        |   +--user.py .......... fichier de gestion des données utilisateurs
    | +-- Librairies/ .......... Librairies du serveur
     | +-- celeryLib ........ Tasks de celery et creation de l'app
        |   +-- dataCollectlib ...
        |   +-- disturbanceLib ... Module de collecte/detection/traitement des pannes
        |   +-- graphLib ......... Graphe du reseau
    | +-- models/ .............. modeles de la base de données utilisateur
    | +-- path.py .......... trajets
        | +-- user.py .......... utiliseteurs
        | +-- subPath.py ....... sous trajets
        | +-- tweet.py ......... tweets
        | +-- disturbance.py ... pannes
        |  +-- transportStop.py . arrets
| +-- runcelery.py .......... fichier d'execution de l'application flask
| +-- runserveur.py ......... fichier d'execution de l'application celery
| +-- serveur.config ........ fichier de configuration de flask / celery
| +-- serveur_test.py ....... fichier de test de l'application flask
```

Pour installer toutes les dépendances dans l'environnement virtuel :

```
pip install -r pfa/requirements.txt
```

Notre application en plus du framework Flask se base sur Celery. Pour pouvoir l'utiliser il vous faut intaller RabbitMQ. Si vous utilisez ubuntu ou debian vous pourrez l'installer en tapant la commande :

```
sudo apt-get install rabbitmq-server
```

Sinon les autres : `http://www.rabbitmq.com/download.html`

Pour lancer le serveur Flask :

```
python pfa/runserver.py
```

Pour lancer Celery :

```
python pfa/runcelery.py worker -B
```

Le l'API est maintenant disponible à l'adresse http ://localhost :8000

1.2 Interface utilisateur

Le rendu de l'interface utilisateur est organisé de la façon suivante :

```
+-- www/ ..................... répértoire principal du projet Cordova
|   +-- API/ ................. fichiers json de l'application
    |   +-- lines.json ....... représentation du réseau de la TBC
    |   +-- posts.json ....... tous les trajets
    |   +-- user.json ........ uilisateur
|   +-- bower_components/ .... éléments Polymer utilisés
|   +-- css/ ................. style propre au projet Cordova
|   +-- img/ ................. images utilisées sur l'application
|   +-- js/ .................. js propre au projet Cordova
|   +-- node_modules/ ........ repértoire de dépendance propre à Polymer
|   +-- bower.json ........... json propre à Polymer
|   +-- ressources/ .......... fonctions utilisables par les clients
|   +-- index.html ........... instance de jeu
|   +-- main.html ............ fonctions auxilières utiles à game
|   +-- post-card.html ....... élément personalisé représentant un trajet
|   +-- post-list.html ....... élément personalisé représentant une liste de trajets
|   +-- user-page.html ....... élément personalisé représentant la page de profile
|   +-- user-login.html ...... élément personalisé représentant la page de connexion
|   +-- cache-config.json .... configuration du cache
|   +-- apps.js .............. js associé au cache
+-- README.md ................ guide d'utilisation
+-- install.sh ............... script générant l'exécutable pour mobile
```

Pour résumer, le dossier www/ représente le repértoire principal du projet Cordova (contient la web application et des fichiers propre à Cordova et à Polymer). Le README.md est un guide d'utilisation du code client, il représente une sorte de tutoriel pour l'installation de l'environnement et l'utilisation de Apache Cordova dans la génération de l'exécutable mobile (Android ou iOS) de notre application. Il contient aussi des liens utiles, par exemple le drive qui contient notre exécutable pour Android (apk)...

Pour l'utilisation, il suffit d'intégrer le repertoire www/ dans un projet Apache Cordova. Et pour visualiser le résultat :

— Sur le navigateur, il suffit de lancer la commande suivante dans le répertoire www/ et aller sur le port 8000 :

```
$python -m SimpleHTTPServer
```

— Sur un mobile (android ou iOS), il faut compiler le projet Cordova

L'exécutable généré pour Android (apk) est disponible sur google drive, il est accessible depuis le git du projet :

`https://github.com/piroux/pfa29_client/tree/master/www`

1.3 Commentaires

Ils sont rédigés en anglais et constitués de la manière suivante :

```
/**
 * Description rapide de la fonction.
 * @Name  : fonction_nom
 * @Param : type - description du paramètre 1
 * @Param : type - description du paramètre 2
 * @Return: type - description du retour
 * [@TODO  : Elements à ajouter à la fonction.]
 * [@WARNING:Dangers relatifs à l'usage de la fonction.]
 **/
```