Rapport du projet de *Web Data Management*

Shendan Jin & Olivier Marty

22 février 2016

1 Présentation du projet

Le but de ce projet est d'utiliser différentes sources d'informations sur internet qui concernent l'état d'un moyen de transport (lignes de métro, de train, stations vélo vide ou pleine, ou encore un bouchon sur l'autoroute (non implémenté)) afin de prévenir l'utilisateur lorsque celui-ci va les utiliser.

Pour cela on se connecte à son agenda (Google Calendar), et on analyse ses emails (Gmail) pour prévoir ses déplacements, mais d'autres sources pourrait être ajoutées, de façon modulaire. Pour chaque événement, on géolocalise l'adresse avec l'API d'OpenStreetMap, puis on cherche les moyens de transport utiles pour se rendre à ces coordonnées. En cas de problème sur l'un de ces moyens de transport, l'utilisateur est notifié par le medium de son choix (email, ou sms via l'API de free mobile).

2 Description globale

Le projet est divisé en trois principales composantes :

2.1 Les événements

Les événements auquel l'utilisateur va se rendre sont représentés par un objet contenant un identifiant unique, la date, l'addresse, et une description. Ces événements sont récupérés à partir de deux sources : Google Calendar et Gmail. Cependant les seuls mails aboutissant à la détection d'un événement sont ceux de la forme :
"Rendez-vous
le 22/02/2016 08h45
à Université Paris Diderot
pour le cours de WDM"

2.2 Les sources d'état du trafic

Une information concernant l'état d'un moyen de transport est représenté par un objet contenant un identifiant unique, un message, et un attribut booléen qui détermine si le moyen de transport présente un problème.

Les données proviennent de différentes sources :

- **État des lignes de la RATP** Les données sont récupérées en scrapant l'url `http://ratp.fr/meteo/`. En effet la ratp ne fournit aucune API publique dynamique. L'API statique `http://data.ratp.fr/explore/dataset/offre-transport-de-la-ratp-format-gtfs/` a cependant été utilisée pour obtenir la localisation des stations et les lignes les traversant.
- **État des lignes Transilien** Les données sont récupérées en scrapant l'url `http://www.transilien.com/info-trafic/temps-reel`. L'API SNCF (`https://ressources.data.sncf.com/explore/dataset/osm-mapping-idf/`) est par contre utilisée pour déterminer la postition des stations et les lignes les traversant.
- **Vélos en libre service JCDecaux** Les données sont récupérées via l'API vls de JCDecaux (voir `https://developer.jcdecaux.com/#/opendata/vls`).

2.3 La boucle principale

Cette boucle articule les différents modules entre eux. Elle consomme les événements depuis les sources, qu'elle rafraichit toutes les 30 secondes. Une demi-heure avant qu'un un événement arrive à échéance, on cherche la localisation de l'événement, puis on cherche les sources qui peuvent être utiles pour ce rendre à cet événement (stations proches). Le module source récolte alors l'état de ces sources, et une notification est envoyée pour chaque problème soulevé.

3 Programmes python

Afin d'éxecuter un programme, il faut lancer la commande `python3 fichier.py`. Il faudra peut-être installer les dépendances python : `pip3 install -r requirements.txt`

main.py C'est le programme principal. Lors du premier lancement, il demandera l'autorisation de se connecter aux API Google Calendar et Gmail.

demo.py Lance une démonstration qui ignore les événements mais lance les notifications à partir d'une adresse.

find.py Fournit une aide pour trouver les identifiants utilisés en interne pour désigner une source. Il propose une recherche interactive dans les listes téléchargées depuis les différentes API.

ratp_preprocessing.py Les données statiques fournies par la ratp n'était pas dans un format qui nous convenait (pour utiliser des termes de SGBDR, il faut faire plusieurs jointures pour trouver les lignes qui passent par un arrêt !), en plus de peser plus de 500Mo ! Ce fichier extrait de ces données la liste des stations, leurs localisations et les lignes les traversant, et le fichier final (`ratp.csv`) ne pèse alors plus que que 500Ko. Les données brutes sont disponibles à l'adresse `http://data.ratp.fr/explore/dataset/offre-transport-de-la-ratp-format-gtfs/`.

map.py Génère un fichier cvs contenant la liste des identifiants associés à des coordonnées d'intérêt, pour visualisation. Une carte présentant ces données est disponible à l'adresse `https://www.google.com/maps/d/edit?mid=z6ibLBE5MDrk.kudB9LIy9Cws&usp=sharing`