Bourgmayer Vincent

02/01/2016

Spécification

Technologie et opportunités.

Pour la mise en œuvre de l’outil de tableau de bord centralisé, il convient d’obtenir un grand nombre d’information. Ces informations sont d’un point de vue personnel sont longues et contraignantes à collecter. Dans certains cas elles peuvent demander des connaissances très pointues et relever du parcours du combattant. J’ai découvert le projet : <http://fr.openfoodfacts.org/> .

C’est un projet de « wikipédia » de produit alimentaire mondial. La base de données fonctionne sur MongoDB, elle est libre et gratuite et un début d’API a été mis en place, … Ces données sont renseigné par un ensemble de client et mise à disposition. On y trouve quasiment le contenu des étiquettes que l’on peut obtenir par simple un scan du code barre ou entrant manuellement le code (du code barre) du produit. Il y a déjà près de 50 000 produits renseigné dans la base. Bien que parfois incomplet, cela mettra du temps un peu de temps à se mettre en place.

L’application smartphone pour scanner un produit est développée en utilisant phoneGap et apache Cordova qui sont également gratuit. Après étude et test des outils du projet, je les ai trouvé satisfaisants et avec un très fort potentiel. Cependant cela reste brouillon, vu que le projet est celui d’une association à but non lucratif reposant sur un travail bénévole. Il tente de combler le manque de transparence de la grande distribution. Bien évidemment il ne saurait convenir parfaitement car il ne fonctionnera pas avec des petits producteurs locaux, cependant il serait un précieux alliés pour la mise en place d’un tel projet.

Le projet pourrait ainsi être mis en œuvre avec une base de donnée MongoDB, Apache Cordova et PhoneGap et aussi l’association AngularJS/NodeJS pourrait une rapidité et une fluidité extraordinaire.

Cela réduirait l’utilisation de langage à : HTML, CSS et JavaScript. Il existe une API pour connecter NodeJS à MongoDB qui s’appelle Mongoose ainsi qu’un module pour lier AngularJS et Apache Cordova/Phone Gap. La construction de graphique se ferait via HighChart et l’exploitation de tableau avec l’excellent (mais compliqué) DataTable.

Concernant la modularité de l’outil, celle-ci doit reposer sur un fonctionnement sécuritaire. Un compte étant anonyme, un utilisateur souhaitant par exemple connecter sa consommation d’autoroute (et donc obtenir les informations de son nombre de passage, de la durée de ses trajets, de sa vitesse moyenne, etc…) pourra générer une clef à usage unique et de validité temporaire qu’il offrira à l’entreprise gérant l’autoroute qui elle pourra envoyer les données (avec un flux RSS par exemple ?) envoyé les données de manière ANONYME (en omettant ainsi les informations identifiant le client) de consommation dans le tableau de bord. Grâce à une API ouverte. Il resterait au entreprise la liberté de facturé ou autorisé la mise en place de cette connexion. Cependant elle gagnerait en transparence, la sécurité des données seraient conserver, les entreprises pourraient choisir quelles sont les informations qu’elles décident de partager, etc…

Après un temps de recherche, et plusieurs tentative infructuense pour installer MongoDB et NodeJS sur mon matériel dans le but de réaliser des tests, j’ai finalement choisi d’essayer elastic search.

J’ai découvert le module « express » pour Node pour créer rapidement le squelette d’une application.

J’ai finalement réussi à créer une base de donnée mongoDB, et à l’utiliser avec node.js( via mongosse). Mon test grossier fonctionne pour l’insertion de donnée .

**Création de compte anonyme :**

**Idée 1 : Utilisation de Captcha :**

La page de création de compte contiendrait un champ pour saisir un login (qui doit être unique).

Un bouton générer un mot de passe placer sous le champs de login serait le seul autre élément avec lequel interragir. En cliquant sur ce bouton, un CAPTCHA apparaitrait ainsi qu’un champs pour saisir le code inscrit dans le captcha. Le mot de passe serait le code du CAPTCHA. Une fois saisie correctement par l’utilisateur (entre 1 et 3 tentatives par CAPTCHA) une redirection affichant le CAPTCHA en plus grand indiquerait qu’il s’agit du mot de passe de l’utilisateur.

Avantage :

* Pas besoin d’E-mail
* Purement anonyme

Inconvenient :

* Le CAPTCHA n’est pas une solution sur à 100% pour éviter les robots
* Pas très user-friendly
* Pas de possibilité de récupérer son password si on l’a oublié.
* Problème d’accès au non-voyant, …

**Idée 2 : Utilisation d’adresse E-mail jetable**

En utilisant des outils comme : jetable.org, yopmail ou trashmail.com on peut générer des adresse e-mails bidon, le temps de la connexion. Certain proposent une redirection vers une vrai adresse mail et gardent des traces (ou non) et d’autre sont des adresse publiques, accessible par tout le monde.

Avantage :

* Ne laisse théoriquement pas de trace
* Permet l’anonymat

Inconvénient :

* Dans certains cas, laisse des traces => anonymat partielle
* Ne bloque pas les robots
* Pas du tout user friendly
* En cas d’adresse publique : pas de possiblité de communiquer un mot de passe perdu