

Projet

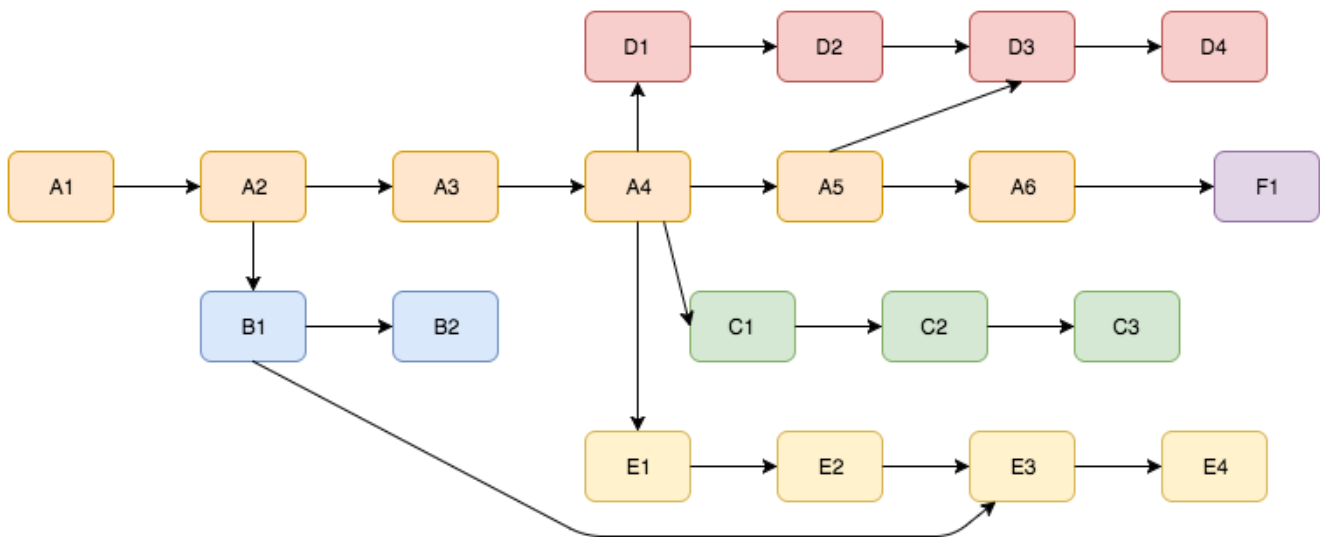
Le projet consiste à récupérer un ensemble de données provenant de la ville de Montréal et d'offrir des services à partir de ces données. Il s'agit de données ouvertes à propos d'établissements ayant reçu des constats d'infraction lors d'inspections alimentaires.

Vous avez la possibilité de construire un projet selon vos préférences. Les diverses fonctionnalités vous offrent un certain nombre de points d'expérience (XP) et vous ouvre le chemin vers d'autres fonctionnalités à développer. Si vous faites le travail individuellement, vous devez accumuler un minimum de 100 XP. Si vous faites le travail en équipe de 2 personnes, vous devez accumuler un minimum de 200 XP. Vous n'êtes pas tenu de développer toutes les fonctionnalités.

Le point de départ est la fonctionnalité A1. Les fonctionnalités ont des dépendances entre elles. Par exemple, il faut avoir complété A1 avant de pouvoir faire A2; il faut avoir complété D2 et A5 avant de pouvoir faire D3.

Fonctionnalités

Voici une carte représentant les dépendances entre les fonctionnalités :



Voici les exigences pour chaque fonctionnalité :

A1 20xp

La liste des contrevenants est obtenue en format XML à l'aide d'une requête HTTP et son contenu est stocké dans une base de données MongoDB.

Les données sont accessibles à partir de l'adresse suivante :

<http://donnees.ville.montreal.qc.ca/dataset/inspection-aliments-contrevenants/resource/92719d9b-8bf2-4dfd-b8e0-1021ffcaee2f>

A2 5xp

L'importation de données du point A1 est faite automatiquement chaque jour à minuit.

A3 5xp

Le système écoute les requêtes HTTP sur le port 3000. La route « /doc » fait apparaître la documentation de tous les services REST. La documentation est en format HTML, généré à partir de fichiers RAML.

A4 10xp

Le système offre un service REST permettant d'obtenir la liste des contrevenants ayant commis une infraction entre deux dates spécifiées en paramètre. Les dates sont spécifiées selon le format ISO 8601. Les données retournées sont en format JSON.

Ex. GET /contrevenants?du=2016-05-08&au=2016-05-15

A5 15xp

Une application JavaScript/HTML permet de saisir deux dates à partir d'un formulaire HTML. Lorsque l'utilisateur lance la recherche, une requête Ajax contenant les deux dates saisies est envoyée à la route définie en A4. Lorsque la réponse Ajax revient, l'application affiche la liste des contrevenants dans un tableau. L'application est disponible sur la page d'accueil du serveur (route « / »).

A6 10xp

L'application du point A5 offre un mode de recherche par nom du restaurant. La liste de tous les contrevenants est prédéterminée dans une liste déroulante et l'utilisateur choisira un restaurant parmi cette liste. Lorsque l'utilisateur lance la recherche, une requête Ajax est envoyée à un service REST que vous devez créer à cet effet. Lorsque la réponse Ajax revient, l'application affiche l'information des différentes infractions du restaurant.

B1 15xp

Le système détecte les nouveaux contrevenants depuis la dernière importation de données, en dresse une liste sans doublon et l'envoi par courriel automatiquement. L'adresse du destinataire du courriel est stocké dans un fichier de configuration en format YAML.

B2 10xp

Les noms des nouveaux contrevenants sont publiés automatiquement sur un compte Twitter.

C1 10xp

Le système offre un service REST permettant d'obtenir la liste des établissements ayant commis une ou plusieurs infractions. Pour chaque établissement, on indique le nombre d'infractions connues. La liste est triée en ordre décroissant du nombre d'infractions.

C2 10xp

Le système offre un service permettant d'obtenir exactement les mêmes données que le point C1 mais en format XML. L'encodage de caractères doit être UTF-8.

C3 5xp

Le système offre un service permettant d'obtenir exactement les mêmes données que le point C1 mais en format CSV. L'encodage de caractères doit être UTF-8.

D1 15xp

Le système offre un service REST permettant de modifier l'état d'un contrevenant. Le client doit envoyer un document JSON contenant les modifications à apporter au contrevenant. Le document JSON doit être validé avec json-schema.

D2 5xp

Le système offre un service REST permettant de supprimer un contrevenant.

D3 20xp

Modifier l'application faite en A5 afin de pouvoir supprimer ou modifier les contrevenants retournés par l'outil de recherche. L'application doit invoquer les services faits en D1 et D2 avec des appels Ajax et afficher une confirmation en cas de succès ou un message d'erreur en cas d'erreur.

D4 20xp

Le système offre une procédure d'authentification du type « Login with Twitter » et permet de restreindre l'accès aux fonctionnalités de modification et suppression de D3 uniquement à un utilisateur prédéfini.

E1 10xp

Le système offre un service REST permettant à un utilisateur de se créer un profil d'utilisateur. Le service reçoit un document JSON contenant :

- le nom complet de l'utilisateur;
- l'adresse courriel de l'utilisateur;
- une liste de noms d'établissements à surveiller.

Le document JSON doit être validé avec json-schema.

E2 5xp

Le système offre une interface graphique en HTML pour invoquer le service fait en E1.

E3 15xp

Lorsqu'un nouveau contrevenant est détecté, un courriel est automatiquement envoyé à tous les utilisateurs qui surveillent cet établissement.

E4 15xp

Le courriel envoyé au point E3 contient un lien pour se désabonner au profil. Le lien amènera à une page HTML qui demandera une confirmation à l'utilisateur. Si l'utilisateur confirme le désabonnement, une requête Ajax invoquera un service REST pour supprimer le profil de l'utilisateur.

F1 20xp

Le système est entièrement déployé sur la plateforme infonuagique Heroku.

Remise et correction

La correction du projet se fera en deux étapes. La première étape consiste en une démonstration des fonctionnalités de votre logiciel. Lorsque votre projet est prêt, vous prenez rendez-vous avec l'enseignant pour lui présenter votre projet. Du temps sera prévu à cet effet en fin de session, mais si votre projet est prêt plus tôt, il sera possible de se faire corriger plus rapidement. La deuxième étape est une remise du code source du projet par Moodle. Le code sera ensuite évalué selon plusieurs critères. La remise finale du code doit être faite au plus tard le 9 décembre 2016.

Technologies

Voici les technologies imposées :

- Node.js 6.5 ou plus
- express.js 4 ou plus
- MongoDB 3.2 ou plus
- HTML5

Vous pouvez utiliser les bibliothèques et packages Javascript de votre choix. Même chose pour HTML/CSS. Il est permis d'utiliser des templates CSS tant que la licence d'utilisation est respectée.

Pondération

Fonctionnalités (XP) : 60%

Qualité du projet : 40%

La qualité du projet comporte les éléments suivants :

- la propreté du code (indentation, longueur des fonctions, duplication de code, style uniforme, nomenclature, commentaires);
- la qualité de la documentation (fautes de langue, documentation complète (incluant les cas d'erreur) et pertinente, readme avec une procédure d'installation et d'exécution);
- le respect des standards du web (REST, encodage de caractères);
- la gestion des erreurs.

Travail d'équipe

Si le travail est fait en équipe de 2 personnes, il est exigé que chaque membre de l'équipe contribue à un minimum de 100 XP. L'équipe doit utiliser un dépôt privé git (github ou bitbucket) et m'en donner l'accès afin que je puisse vérifier la participation de tous les membres de l'équipe.