

Projet

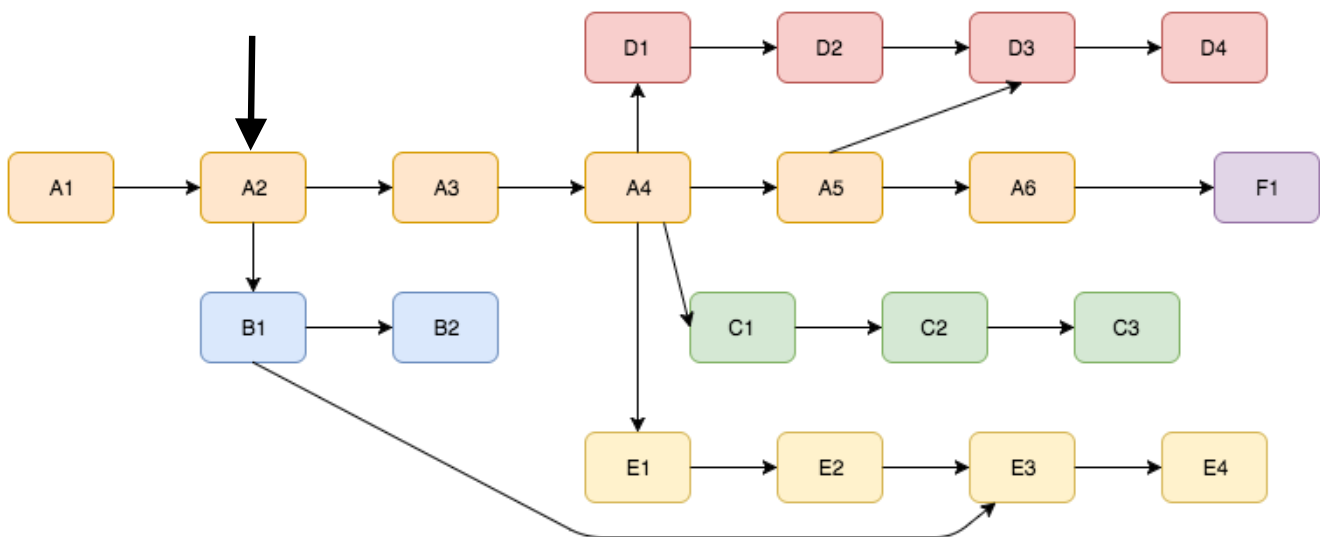
Le projet consiste à **recupérer** un **ensemble de données** provenant de la ville de Montréal et d'**offrir des services** à partir de ces données. Il s'agit de données ouvertes à propos d'**installations pour faire des activités sportives**.

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 et un maximum de 120 XP. Si vous faites le travail en équipe de 2 personnes, vous devez accumuler un minimum de 200 XP et un maximum de 220 XP. Vous n'êtes pas tenu de développer toutes les fonctionnalités mais lorsque vous choisissez une fonctionnalité, vous devez la faire au complet.

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 15xp

Trois listes de données doivent être obtenues à l'aide de requêtes HTTP et leur contenu doit être stocké dans une base de données SQLite.

La liste des piscines et installations aquatiques en format CSV :

<https://data.montreal.ca/dataset/4604afb7-a7c4-4626-a3ca-e136158133f2/resource/cbdca706-569e-4b4a-805d-9af73af03b14/download/piscines.csv>

La liste des patinoires en format XML :

<https://data.montreal.ca/dataset/225ac315-49fe-476f-95bd-a1ce1648a98c/resource/5d1859cc-2060-4def-903f-db24408bacd0/download/129-patinoire.xml>

La liste des aires de jeux d'hiver (glissades) en format XML :

http://www2.ville.montreal.qc.ca/services_citoyens/pdf_transfert/L29_GLISSADE.xml

Vous devez organiser la structure de vos données de façon à minimiser le nombre de requêtes faites à la BD pour les fonctionnalités que vous réaliserez dans votre projet. N'hésitez pas à changer la modélisation des données.

A2 5xp

L'importation de données du point A1 est faite automatiquement chaque jour à minuit à l'aide d'un BackgroundScheduler.

A3 5xp

Le système écoute les requêtes HTTP sur le port 5000. La route « /doc » fait apparaître la documentation de tous les services REST. La documentation est en format HTML, généré à partir de fichiers RAML. Intégrez la fonctionnalité du point A2 à l'application Flask créée au point A3.

A4 10xp

Le système offre un service REST permettant d'obtenir la liste des installations pour un arrondissement spécifié en paramètre. Les données retournées sont en format JSON.

Ex. GET /api/installations?arrondissement=LaSalle

A5 10xp

Une application JavaScript/HTML permet de saisir un arrondissement à partir d'un formulaire HTML. Lorsque l'utilisateur lance la recherche, une requête asynchrone contenant l'arrondissement saisi est envoyée à la route définie en A4. Lorsque la réponse asynchrone revient, l'application affiche la liste des installations 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 d'installation. La liste de toutes les

installations est prédéterminée dans une liste déroulante et l'utilisateur choisira une installation parmi cette liste. Lorsque l'utilisateur lance la recherche, une requête asynchrone est envoyée à un service REST que vous devez créer à cet effet. Lorsque la réponse asynchrone revient, l'application affiche l'information connue sur cette installation.

B1 10xp

Le système détecte les nouvelles installations 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 stockée dans un fichier de configuration en format YAML.

B2 10xp

Les noms des nouvelles installations sont publiés automatiquement sur un compte Twitter.

C1 10xp

Le système offre un service REST permettant d'obtenir la liste des installations dont les données ont été mises à jour en 2021.

Pour chaque installation, on indique toute l'information connue.

La liste est triée en ordre croissant du nom de l'installation.

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'une glissade. Le client doit envoyer un document JSON contenant les modifications à apporter à la glissade. Le document JSON doit être validé avec json-schema.

D2 5xp

Le système offre un service REST permettant de supprimer une glissade.

D3 20xp

Modifier l'application faite en A5 afin de pouvoir supprimer ou modifier les glissades retournées par l'outil de recherche. L'application doit invoquer les services faits en D1 et D2 avec des appels asynchrones et afficher une confirmation en cas de succès ou un message d'erreur en cas d'erreur. Il faut développer la même fonctionnalité pour les piscines et installations aquatiques.

D4 20xp

Le système offre une **procédure d'authentification** du type « **Basic Auth** » et permet de **restreindre l'accès aux fonctionnalités de suppression** de D3 uniquement à un **utilisateur prédéfini**. Assurez-vous de spécifier clairement les données d'authentification pour la correction.

Attention : Une authentification de type **Basic Auth** n'est pas la même chose que l'authentification avec salt et hachage vu en classe. Une recherche sera nécessaire.

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 d'arrondissements 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'une **nouvelle installation est détectée**, un **courriel est automatiquement envoyé** à tous les utilisateurs qui surveillent l'arrondissement contenant cette installation.

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 asynchrone** **invoquera** un service REST pour **supprimer le profil** de l'utilisateur.

F1 20xp

Le système est **entièrement déployé** sur la **plateforme infonuagique Heroku**.

Remise

Le répertoire de travail contenant les fichiers doit être archivé dans un fichier zip et nommé selon les codes permanents des auteurs. L'archive doit être remise **par Moodle** (la date de remise est sur Moodle). **Aucun retard ne sera accepté** et une pénalité sera appliquée pour une archive non conforme sans les codes permanents.

~~Si vous voulez travailler en équipe, vous devez mettre votre logiciel sous gestion de sources dans un dépôt **privé** git (ex. github, bitbucket) et m'en donner un accès en lecture. Cette mesure sert à m'assurer que les coéquipiers ont partagé l'effort équitablement. Un coéquipier n'ayant pas offert une participation~~

~~significative n'aura pas les points pour le travail. Il est exigé que chaque membre de l'équipe contribue à un minimum de 100 XP.~~

Un fichier 'correction.md' rédigé en markdown doit être placé à la racine du projet. Vous devez y inscrire tous les points que vous avez développé et comment tester chaque point. Si un point a été développé mais qu'il n'est pas présent dans ce fichier, il ne sera pas corrigé. Pour le point F1, nous n'avez qu'à fournir l'URL de votre application.

Technologies

Les technologies permises sont celles disponibles dans l'environnement virtuel. Le code Python doit respecter PEP8. Tous les fichiers sources doivent être encodés en UTF8.

Pondération

Fonctionnalités (XP) : 60%

Qualité du projet : 40%

La qualité du projet est évalué en conformité avec le document « Évaluation de la qualité du logiciel ».

Environnement virtuel

Vous n'êtes pas dans l'obligation d'utiliser l'environnement virtuel mis à votre disposition. Par contre, la correction aura lieu dans cet environnement virtuel. Il est de votre responsabilité de vous assurer que votre travail soit compatible avec cet environnement.

pycodestyle

Il est attendu que l'exécution de pycodestyle sur votre code source ne soulève aucune erreur et aucun avertissement.

Évaluation de la qualité du logiciel

Un document nommé "Évaluation de la qualité du logiciel" est disponible sur Moodle. Tous les éléments dans ce document sont des exigences à ce travail.

Base de données

Vous devez modéliser vous même votre base de données. Un répertoire doit être créé avec le nom « /db » et il doit contenir un script de création de tables avec le nom « db.sql ».