TP #2

VOUS VOUS METTREZ EN BINOME POUR LA SUITE DU TP

Données d'Installations Sportives des Pays de la Loire

1. telecharger les 3 fichiers CSV jeux de données, au format CSV :
   - [Installations] http://data.paysdelaloire.fr/donnees/detail/equipements-sportifs-espaces-et-sites-de-pratiques-en-pays-de-la-loire-fiches-installations
   - [Equipements] http://data.paysdelaloire.fr/donnees/detail/equipements-sportifs-espaces-et-sites-de-pratiques-en-pays-de-la-loire-fiches-equipements
   - [Activités] http://data.paysdelaloire.fr/donnees/detail/equipements-sportifs-espaces-et-sites-de-pratiques-en-pays-de-la-loire-activites-des-fiches-equ
2. manipuler les fichiers CSV comme au TP #1 avec le module 'csv'

Base de Données : mettre en place une base de données alimentée par ces fichiers de données équipement, activités, installations

1. Définir la structure de BD (avec clés primaires & clés étrangères), en fonction de l'API
2. charger les données d'EQUIPEMENTS et d'ACTIVITES en CSV, et alimenter la BD
3. charger les données d'INSTALLATIONS en CSV ou JSON (au choix), et alimenter la BD > RQ : vous pouvez installer MySQL connector pour python (http://dev.mysql.com/doc/connector-python/en/connector-python-installation-binary.html), > ou sinon utiliser sqlite qui est supporté nativement par Python (pour accéder à la base, utiliser par exemple l'outil www.sqllitebrowser.org ou le plugin SQLlite manager de firefox)

Structure des données en objet

A partir d'une requête à la BD par exemple => construire une structure d'objets (grappe d'objets) qui représente ces données objets qui peuvent être manipulés ensuite en objets

Recherche

Faire évoluer votre application pour afficher des données filtrées des équipements sportifs (avec une recherche sur certaines données que l'utilisateur sélectionnera), par ex : pouvoir afficher les lieux & salles correspondant à un sport donné

Mettez en place toute la gestion de tests unitaires

1. vous utiliserez les tests unitaires proposés nativement par Python https://docs.python.org/3/library/unittest.html

Documenter et renforcer la gestion d'erreurs de votre application

2. utilisez des exceptions à chaque fois qu'une erreur peut se déclencher => cf. librairie Python §exceptions
3. ajoutez la documentation de code (dostring) : (https://docs.python.org/3/library/doctest.html#module-doctest)

Realiser une application web pour fournir des API REST (services)

4. vous utiliserez pour cela le framework BOTTLE http://bottlepy.org/docs/dev/index.html
5. vous fournirez les réponses pour chaque ressource dans le format de représentation qui vous convient (XML ou JSON par exemple)
6. vos API permettront de récupérer et modifier toutes les informations d'installations, équipements, et activités et d'effectuer quelques recherches => vous définirez pour cela au préalable les URI de vos ressources

Réaliser une 2nde application web pour interroger vos API

7. C'est une application web simple qui pourra s'exécuter dans un navigateur (en HTML / Javascript / JQuery) Cette application offre une interface à l'utilisateur pour lui faciliter la saisie et l'affichage des résultats.
8. lorsque vous aurez des API qui fonctionnent avec un client de test, vous pourrez fournir vos services pour que d'autres binomes les testent