Data Technical Test

Partie I : Ingestion et Validation de la donnée [max 3 heures]

Introduction

Chez Kompozite, pour créer de la donnée aujourd'hui les équipes métiers passent souvent par des fichiers tabulaires (csv ou excel), qui passent ensuite par nos outils internes pour être validés et ingérés dans nos base de données.

Cet exercice cherche à reproduire cet enjeu.

Pour cela, il vous est demandé de développer une interface qui accepte un fichier csv en entrée, le lise, le valide et éventuellement le corrige.

Questions:

- Créer une interface permettant d'ingérer des fichiers csv dont les lignes représentent des objets de type Mesh.
- Stocker les données dans une base de données (au choix).
- En cas d'erreur, renvoyer un message d'erreur explicite décrivant le (ou les) problème(s) sur la (les) ligne(s) concernée(s).

Modèle de données d'un objet de type Mesh.

Colonne	Type	Description	Contraintes	Valeurs Possibles	Actions et Transformations attendues
codename	string	Identifier	unique across database, requis		* Vérifier et empêcher l'ajout de doublons.
trame	string	Classement trame	*requis, Enum	* T2 Ra1 M2 E2 * T2 Ra1 M4 E2 * T2 Ra1 M4 E3	* Bloquer la ligne si le string ne correspond à aucune des valeurs possibles données dans la colonne "Valeurs Possibles"
mass_surf	float	Masse Surfacique (kg/m2)	requis, >0		* Transformer le string en float si nécessaire * Bloquer la ligne si la contrainte de positivité

Colonne	Type	Description	Contraintes	Valeurs Possibles	Actions et Transformations attendues
					n’est pas respectée
is_compat_interior_wall	bool	Hauteur de la maille (cm)	requis		* Transformer les strings “vrai”, “faux”, “true”, “false” en valeurs booléennes. * Transformer les strings et/ou nombres “0”, “1”, 0, 1 en valeurs booléennes.
mesh_height	float	Largeur de la maille (cm)	requis, >0		* Transformer le string en float si nécessaire * Bloquer la ligne si la contrainte de positivité n’est pas respectée
mesh_width	float	Masse combustible (MJ/m2)	requis, >0		* Transformer le string en float si nécessaire * Bloquer la ligne si la contrainte de positivité n’est pas respectée
roll_pallet	int	Conditionnement - Nombre de rouleaux par palettes	optionnel		* Transformer le string en entier si nécessaire.
color_names	list of strings		* Enum * 1 item minimum	white, yellow, green, purple, red, blue, orange, magenta, dark, grey, cyan	* Transformer le string en liste * Vérifier que chacun des éléments de la liste correspond bien à une des valeurs possibles données dans la colonne “Valeurs Possibles”. * Bloquer la ligne

Colonne	Type	Description	Contraintes	Valeurs Possibles	Actions et Transformations attendues
					si la contrainte précédente n'est pas respectée (i.e vide ou liste vide non-autorisés).

Livrables:

- L'interface permettant de stocker de la nouvelle donnée.
- Le code source permettant de créer l'interface.

Aide et précisions:

- *requis signifie qu'une valeur doit obligatoirement être présente ⇒ Renvoyer un message d'erreur si la valeur n'est pas présente.
- Le choix de l'interface est ouvert (script python facilement executable, application executable [fichier .exe], API Web locale ou hostée, Appli Web locale ou hostée) - **Il faut choisir la solution avec laquelle vous êtes la plus à l'aise**. Dans tous les cas, il faut penser à la **reproductibilité** des résultats.
- Le choix du langage est ouvert mais python est recommandé.
- L'utilisation de librairies ou frameworks est possible et même recommandé (Ex: Pandas, Pydantic, SQLModel, Polars, SQLAlchemy)
- Pour le stockage le choix de la solution est libre. Vous pouvez par exemple stocker la donnée dans une base de données locale de type SQLite, stocker dans une base de données hébergée chez un cloud provider ou sauvegarder les éléments localement dans des fichiers de type pickle ou json.

Data

- Jeu de données

dummy_meshes_correct.csv

dummy_meshes_with_errors.csv

Partie II : Matching [max 2h]

Introduction

À Kompozite, nous recevons de la donnée de différents clients. Ces clients nous envoient une liste de produits, avec diverses informations comme le nom, la marque(fournisseur), le poids…

Parfois nos clients utilisent les mêmes texte pour les mêmes marques, mais parfois les textes sont différents. Par exemple LAHERA et LAHERA PRODUCTIONS ou MONOCIBEC et FINCIBEC (MONOCIBEC).

C'est important pour Kompozite de pouvoir relier ces marques car les produits de ces clients peuvent alors être traités ensemble et occasionné un gain de temps.

Questions:

Vous avez à votre dispositions 2 listes

- `brands_list_1_short`
- `brands_list_2_short`

Proposez une solution, via un script python, pour aider à l'identification de marque similaires dans les deux fichiers.

Livrables:

- Fichier avec des correspondances probables de marque, idéalement un csv
- Le code utilisé pour le générer le fichier

Aide et précisions:

- Ces listes contiennent des noms de marques de deux clients différents pour les marques commençants par A à K.
- Le but n'est pas d'identifier tous les couples, ni d'être juste à 100%, mais de fournir des propositions qui pourront être validés par un regard humain.
- Une marque d'un fichier peut correspondre à plusieurs marques dans l'autre

brands_list_1_short.csv

brands_list_2_short.csv