

Le but de ce test technique est de tester vos capacités à vous adapter à un nouveau projet et un nouveau framework. Durant votre stage vous allez utiliser un projet open source implémenté en python appelé Django.

Vous allez devoir développer une application basique en utilisant le système de base de données de votre choix.

Toute la documentation nécessaire est disponible à l'adresse suivante <https://docs.djangoproject.com/fr/3.0/intro/install/> ainsi que sur les pages du tutoriel Django 1 à 7.

Des fonctionnalités optionnelles sont indiquées **en vert**. Elles ne sont pas nécessairement expliquées dans le tutoriel mais sont dans la documentation officielle Django et des pistes d'implémentation sont facilement accessibles sur internet.

Les mots clés utilisés proviennent de la documentation Django, si vous ne comprenez pas un terme n'hésitez pas à vous référencer au tutoriel officiel.

Vous devrez créer les Models d'un magasin de vente de bagel. Vous rendrez le projet complet y compris les migrations, ainsi qu'un fichier contenant les différents paquets à installer « requirements.txt ». Vous nous indiquerez quel système de base de données vous avez utilisé et si une installation est nécessaire (sqlite, mysql, postgresql, ...).

Le projet sera testé dans un environnement virtuel python, avec la liste de commande suivante :

- pip install -r requirements.txt
- python manage.py migrate
- python manage.py createsuperuser
- python manage.py runserver

Vous devrez créer au minimum un Model représentant :

1. Une table de clients (différent du model User par défaut) contenant au minimum les champs suivants :
 - Un champ texte nom/prénom
 - Un champ texte email
 - Un champ date de création du compte (à la seconde près au moins)
2. Une table de bagel contenant au minimum les champs suivants :
 - Un champ texte type de pain
 - Un champ nombre représentant le stock restant
3. Une table de compléments à mettre dans le bagel :
 - Un champ texte nom du complément (graine de sésame, crème, saumon, ...)
 - Un champ nombre représentant le stock restant

4. Une table commande contenant au minimum les champs suivants :
 - Un champ texte adresse
 - Un champ date de commande (à la seconde près au moins)

En complément (ces champs peuvent être rajoutés aux Models existants ou peuvent faire l'objet de nouveaux Models) :

1. Chaque commande est associée à un seul client
2. Chaque commande doit contenir au moins un bagel mais n'a pas de limite en dehors des stocks disponibles
3. Chaque bagel dans une commande contient au moins 1 complément et un maximum de 5 compléments dans la limite des stocks disponibles.

Une fois les Models créés vous devrez ajouter les Models suivants au site de l'admin afin qu'ils puissent être édités par l'admin

- Clients
- Bagel
- Compléments
- Commandes (surtout pour la visualisation, l'édition des commandes dans l'admin peut être fastidieux et sera plutôt implémentée dans la suite de l'exercice)

Nous rajouterons les Clients depuis l'admin.

Nous essaierons de rajouter/supprimer des bagels/compléments depuis l'admin.

Un plus : ils ne pourront pas être supprimés s'ils sont liés à des commandes.

Vous pouvez tester les fonctions dans le shell Django en lançant : `python manage.py shell`

Pour créer une commande vous devrez implémenter une fonction python qui prend en paramètre un identifiant de client, une adresse, une liste de dictionnaires décrivant les bagels et leurs compléments sous forme de liste d'identifiants. Par exemple `add_order(client_id=1, address='1 street name', bagels=[{'id': 1, 'toppings': [1, 2, 3]}])`, devra créer une commande pour le client 1 avec le bagel 1 et les 3 compléments 1, 2, 3 pour le bagel 1, le nom des variables devra être cohérent, vous n'êtes pas obligés d'utiliser les clés primaires par défaut de Django (vous pouvez utiliser le mail pour le client par exemple). Vous devrez vous assurer que les bagels et compléments sont bien disponibles en fonction du stock disponible et renvoyer une erreur si le stock est insuffisant. **Attention la commande entière doit être complète sinon elle ne devra pas être créée.**

Vous devrez décrémenter le stock des bagels et compléments correspondant à la quantité utilisée.

Pour supprimer une commande vous devrez implémenter une fonction python qui prend en paramètre un identifiant de commande.

Pour afficher des statistiques vous devrez implémenter une fonction qui permet de récupérer la liste des bagels les plus vendus avec le nombre de commandes, avec en option une plage de date afin d'affiner la recherche qui renverra les bagels les plus vendus durant cette période.

Soyez créatifs, n'hésitez pas à rajouter des champs dans les Models comme par exemple les allergènes et préférences des clients afin de rejeter les commandes ne correspondant pas aux attentes des clients, rajouter un système d'authentification...

Encore une fois vous serez jugés sur votre capacité à vous adapter, n'hésitez pas à nous rendre un projet incomplet nous discuterons ensemble des pistes d'améliorations.