



Data**master**  
i n n o v a t i o n

# Évaluation du Développeur

## Full Stack

Janvier 2024

Créé Par  
**L'équipe Data Master**

## I. Modèles Django :

### 1. Modèle Utilisateur

Créez un modèle Django nommé **User** avec les champs suivants :

- id (AutoField, Clé primaire)
- username (CharField)
- email (EmailField)
- phone (CharField)
- address (CharField)
- role (ChoiceField avec les options : admin, simple-user)

### 2. Modèle Tâche

Concevez un modèle Django nommé Task avec les champs suivants :

- id (AutoField, Clé primaire)
- title (CharField)
- description (TextField)
- priority (CharField)
- status (CharField)
- last\_updated\_by (Clé étrangère à l'utilisateur)
- created\_date (DateTimeField, auto\_now\_add=True)
- updated\_date (DateTimeField, auto\_now=True)

### 3. Relation

Établissez une relation entre les modèles User et Task. Utilisez une clé étrangère (ForeignKey) pour associer chaque tâche à son créateur.

## **II. Modèles Django :**

Implémentez des fonctionnalités d'authentification utilisateur, notamment les points de terminaison pour l'inscription (sign-up) et la connexion (sign-in).

## **III. Opérations en Fonction du Rôle Utilisateur :**

### **1. Opérations Administrateur**

Développez des points de terminaison API accessibles aux administrateurs pour :

- Récupérer toutes les tâches avec des filtres optionnels.
- Récupérer une tâche spécifique par ID.
- Mettre à jour n'importe quelle tâche.
- Supprimer uniquement les tâches qu'ils ont créées.
- Créer une nouvelle tâche pour n'importe quel utilisateur.

### **2. Opérations Utilisateur Simple**

Implémentez des opérations API pour les utilisateurs simples, leur permettant de:

- Récupérer toutes les tâches qu'ils ont créées.
- Récupérer une tâche spécifique par ID (limité à leurs tâches).
- Mettre à jour les tâches qu'ils ont créées.
- Supprimer les tâches qu'ils ont créées.
- Créer une nouvelle tâche pour eux-mêmes.

## **IV. Gestion des Erreurs :**

Implémentez une gestion robuste des erreurs :

- Fournissez des messages d'erreur clairs pour différentes situations.
- Utilisez des codes HTTP appropriés pour indiquer le succès ou l'échec.

## **V. Documentation de l'API :**

Créez une documentation complète pour les points de terminaison API :

- Définissez clairement le but de chaque point de terminaison.
- Spécifiez les paramètres requis pour chaque point de terminaison.
- Décrivez le format attendu des réponses.

## VI. Interface Frontend :

Choisissez un framework frontend (React, Vue) ou utilisez du HTML/JS simple pour créer une interface conviviale.

### 1. Écrans d'Authentification (Inscription et Connexion)

Concevez deux écrans pour l'authentification :

- **Écran d'Inscription :**

- Champs de saisie pour le nom d'utilisateur, l'e-mail, le téléphone, l'adresse, le mot de passe et le rôle.
- Un bouton d'inscription pour créer un nouveau compte utilisateur.
- Gestion appropriée des erreurs et messages pour les tentatives d'inscription infructueuses.

- **Écran de Connexion :**

- Champs de saisie pour l'e-mail et le mot de passe.
- Un bouton de connexion pour authentifier l'utilisateur.
- Gestion appropriée des erreurs et messages pour les tentatives de connexion infructueuses.

### 2. Opérations Frontend

Implémentez des fonctionnalités front end permettant aux utilisateurs de :

- Voir la liste des tâches.
- Ajouter de nouvelles tâches :
  - Les administrateurs peuvent ajouter des tâches pour n'importe quel utilisateur.
  - Les utilisateurs simples peuvent ajouter des tâches pour eux-mêmes.
- Mettre à jour les tâches existantes :
  - Les administrateurs peuvent mettre à jour n'importe quelle tâche.
  - Les utilisateurs simples peuvent mettre à jour uniquement les tâches qu'ils ont créées.
- Supprimer des tâches :
  - Les administrateurs peuvent supprimer n'importe quelle tâche.
  - Les utilisateurs simples peuvent supprimer uniquement les tâches qu'ils ont créées.

## VII. Soumission :

- Créez un répertoire GitHub pour le projet.
- Incluez un README.md détaillé avec des instructions pas à pas pour la configuration et l'exécution du projet localement.
- Soumettez le lien vers le répertoire GitHub.