

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

19/03/2024

# Guide d'utilisation

Application web Django

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

ASSEMAT Loris, CHIARLINI Chloé, PIUZZI Florent, VOLLAND Jimmy

## Table des matières

Prérequis .....	2
Machine Virtuelle avec SSH .....	2
Dépôt Git .....	2
Commencer avec l'application.....	2
Lancement avec une commande (One Button Config) .....	2
Configurer l'application .....	2
Création de fausses données (feature preview) .....	3
Tests unitaires.....	3
Administrer un utilisateur.....	3
Avec une ligne de commande.....	3
Depuis l'espace utilisateur (Responsable uniquement) .....	4
Administrer un projet .....	4
Avec une ligne de commande.....	4
Depuis l'espace utilisateur (Responsable uniquement) .....	4
Administrer une tâche .....	4
Avec une ligne de commande.....	4
Depuis l'espace utilisateur (Responsable uniquement) .....	5
Accéder à son espace .....	3

## Prérequis

### Machine Virtuelle avec SSH

Pour pouvoir utiliser l'application il est nécessaire que l'environnement l'hébergeant possède python et Django.

### Dépôt Git

Il est nécessaire de cloner le dépôt git <https://github.com/skomanex/evaldjango/> si cela n'est pas déjà fait.

## Commencer avec l'application

### Lancement avec une commande (One Button Config)

La commande :

- `Python3 manage.py bootapp ip:port`

Permet de lancer l'application à l'ip:port spécifié, ATTENTION, ceci entraine la suppression des données et l'instanciation des données de test, à ne pas utiliser en PROD.

Si cette méthode est choisie les paragraphes suivants ne vous concerne pas.

### Configurer l'application

L'application peut être lancé en se situant dans le dossier contenant le fichier manage.py et en exécutant la commande :

- `python3 manage.py runserver`

Selon la configuration de votre environnement il faut peut-être spécifier l'adresse IP et le port d'écoute pour l'application. D'après la documentation Django la commande runserver démarre un petit serveur web et ne devrait donc pas être utilisé en production.

Dans notre propre environnement de développement nous utilisons :

- `python3 manage.py runserver 192.168.56.12:8000`

A noter :

Dans le fichier settings du projet Django nous avons configurés `ALLOWED_HOSTS = ['*']`

En production le serveur s'occupe de gérer qui sont les hôtes autorisés et doit donc être configuré de manière appropriée.

La commande :

- `python3 manage.py createsuperuser`

Est native à Django et permet la création d'un super utilisateur. Le super utilisateur est un moyen d'accéder au back office Django. Il est recommandé d'en créer un seul et il n'est pas nécessaire au bon fonctionnement de l'application. (Ce pourquoi la création des objets n'est pas couverte en utilisant le back office Django).

## Création de fausses données (feature preview)

L'application propose une nouvelle commande avec deux options :

- `python3 manage.py testdata create`
- `python3 manage.py testdata delete`

La première viens créer les objets présents dans `./projectmanagement/management/fake_data.py` . L'ensemble de ces données sont flaggées avec un attribut 'estTest'. Il est possible d'ajouter ou de retirer des données présentent dans `fake_data.py`.

La deuxième commande supprime l'ensemble des données flaggées avec 'estTest'.

## Tests unitaires

Les tests unitaires sont définis dans `./projectmanagement/tests.py` et peuvent être exécutés avec la commande :

- `python3 manage.py test`

## Accéder à son espace

L'URL :

- `IP:port/projectmanagement/`

Permet d'accéder à la page de connexion où il faut rentrer ses identifiants, le reste de l'application vérifie l'id de l'utilisateur dans les requêtes, il faut donc absolument être connecté, notamment pour accéder aux pages de création de projet/tâches/utilisateurs.

## Administrer un utilisateur

A noter :

- Sauf utilisation de la commande `testdata create` ou utilisation du back office Django le premier utilisateur est forcément à créer avec la ligne de commande.
- Les lignes de commandes ne devrait être utiliser que pour créer le premier utilisateur responsable. Un CRUD est possible avec les commandes sur tous les objets et ce par soucis de cohérence mais est cependant découragé car mauvaise pratique. Le « bon » CRUD doit passer par les canaux mis à disposition au travers de l'application même.

## Avec une ligne de commande

La commande :

- `python3 manage.py appuser create`

Lance la création d'un utilisateur pour l'application, vous êtes guidée pour remplir les champs nécessaires dans l'utilisateur. A la fin du processus, l'utilisateur est créé et pourra donc se connecter en utilisant son email et mot de passe sur la page de login.

La commande :

- `python3 manage.py appuser modify`

Vous guide au travers de la recherche d'un utilisateur particulier pour modifier ses champs.

La commande :

- `python3 manage.py appuser delete`

Vous guide au travers de la recherche d'un utilisateur pour le supprimer.

## Depuis l'espace utilisateur (Responsable uniquement)

### Administrer un projet

A noter :

- Un CRUD est possible avec les commandes sur tous les objets et ce par soucis de cohérence mais est cependant découragé car mauvaise pratique. Le « bon » CRUD doit passer par les canaux mis à disposition au travers de l'application même.

### Avec une ligne de commande

La commande :

- `python3 manage.py appproject create`

Lance la création d'un projet pour l'application, vous êtes guidée pour remplir les champs nécessaires dans le projet. A la fin du processus, le projet est créé.

La commande :

- `python3 manage.py appproject modify`

Vous guide au travers de la recherche d'un projet particulier pour modifier ses champs.

La commande :

- `python3 manage.py appproject delete`

Vous guide au travers de la recherche d'un projet pour le supprimer.

## Depuis l'espace utilisateur (Responsable uniquement)

L'URL :

- `IP:port/projectmanagement/createproject`

Permet de créer un projet, cette URL est accessible directement depuis la page utilisateur

### Administrer une tâche

A noter :

- Un CRUD est possible avec les commandes sur tous les objets et ce par soucis de cohérence mais est cependant découragé car mauvaise pratique. Le « bon » CRUD doit passer par les canaux mis à disposition au travers de l'application même.

### Avec une ligne de commande

La commande :

- `python3 manage.py apptask create`

Lance la création d'une tâche pour l'application, vous êtes guidée pour remplir les champs nécessaires dans la tâche. A la fin du processus, la tâche est créée.

La commande :

- `python3 manage.py apptask modify`

Vous guide au travers de la recherche d'une tâche particulière pour modifier ses champs.

La commande :

- `python3 manage.py apptask delete`

Vous guide au travers de la recherche d'une tâche pour la supprimer.

### Depuis l'espace utilisateur (Responsable uniquement)

L'URL :

- `IP:port/projectmanagement/createtask`

Permet de créer un projet, cette URL est accessible directement depuis la page utilisateur