

# Serveur Social

Définition du protocole & architecture





# Fonctionnalités



**Créer des profils** avec la possibilité de mettre un **alias** et un **statut ouvert ou fermé**



**S'abonner à d'autres utilisateurs** ou **rejoindre des fils** de discussion pour **être notifié**



**Avoir une discussion par thèmes** et sous-thèmes pour former une **arborescence** et ajouter des **tags pour améliorer la visibilité**



**Faire un post** pour avoir une communication à grande échelle de la part des élus



**Disposer d'une messagerie one to one** pour avoir une remontée des informations vers les élus





# Terminologie

**01**

## Groupe

Sujet principale de discussions  
Ex : Social, Numérique, Sport, ...

**02**

## Topic

Fil de discussion dans un groupe  
Ex : "L'impact de l'IA sur la société"



**03**

## Commentaire

Message d'un topic  
Ex: "Et sur l'environnement ?"

**04**

## Abonnement

Suivre l'activité d'un utilisateur, d'un  
topic ou d'un groupe



# Gestion des utilisateurs

## Users

GET

/users/me

users\_me\_list



PATCH

/users/me

users\_me\_partial\_update



GET

/users/{id}

users\_read



GET

/users/{id}/groups

users\_groups\_list



## Friends

GET

/friends/

friends\_list



GET

/friends/requests

friends\_requests\_list



POST

/friends/{id}

friends\_create



POST

/friends/{id}/accept

friends\_accept\_create



DELETE

/friends/{id}/delete

friends\_delete\_delete



POST

/friends/{id}/refuse

friends\_refuse\_create



# Gestion des discussions



## Groups



GET

/groups/

groups\_list ▾ 🔒

POST

/groups/create

groups\_create\_create ▾ 🔒

GET

/groups/{id}

groups\_read ▾ 🔒

GET

/groups/{id}/members

groups\_members\_list ▾ 🔒

## Topics



GET

/topics/

topics\_list ▾ 🔒

POST

/topics/create

topics\_create\_create ▾ 🔒

GET

/topics/{id}

topics\_read ▾ 🔒

DELETE

/topics/{id}/delete

topics\_delete\_delete ▾ 🔒

## Tags



GET

/tags/

tags\_list ▾ 🔒

POST

/tags/assign/{topic\_id}

tags\_assign\_create ▾ 🔒

POST

/tags/create

tags\_create\_create ▾ 🔒

# Gestion des interactions

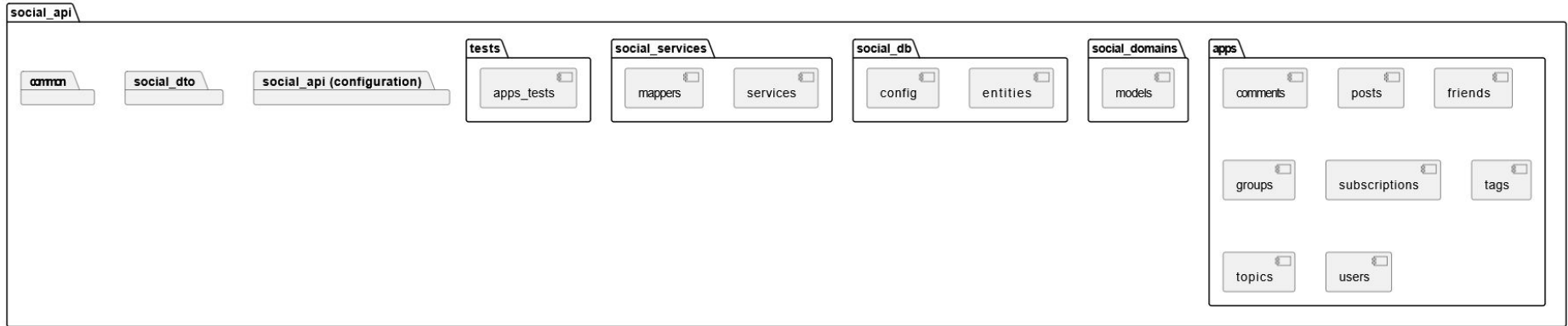
## Subscriptions

POST	/subscriptions/groups/{id}	subscriptions_groups_create	✓	🔒
DELETE	/subscriptions/groups/{id}/unsubscribe	subscriptions_groups_unsubscribe_delete	✓	🔒
POST	/subscriptions/topics/{id}	subscriptions_topics_create	✓	🔒
DELETE	/subscriptions/topics/{id}/unsubscribe	subscriptions_topics_unsubscribe_delete	✓	🔒
POST	/subscriptions/users/{id}	subscriptions_users_create	✓	🔒
DELETE	/subscriptions/users/{id}/unsubscribe	subscriptions_users_unsubscribe_delete	✓	🔒

## Comments

DELETE	/comments/{id}/delete	comments_delete_delete	✓	🔒
POST	/comments/{id}/like	comments_like_create	✓	🔒
GET	/topics/{id}/comments	topics_comments_list	✓	🔒
POST	/topics/{id}/comments/add	topics_comments_add_create	✓	🔒

# Architecture backend :



# Architecture backend :



Dossier racine demperm/src/serveur/social: .gitignore, README et fichier de configuration du projet (.python-version, uv.lock et pyproject.tom) et tous les autres dossiers du projet.

social/social\_api: contient les dossiers concernant les différent aspects de l'API, un manage.py permettant de gérer le serveur et un \_\_init\_\_.py pour être sur que Python reconnaisse tout le travail à l'intérieur de ce dossier comme un même package.

social/social\_api/apps: contient les dossiers dédié à toutes les applications du serveurs (comments, posts, friends, groups, subscriptions, tags, topics et users), chacun ayant en commun un fichier urls.py, views.py, \_\_init\_\_.py.

social/social\_api/social\_api: contient différent fichiers de configuration et de paramétrage pour ASGI, Django, les urls utilisées et WSGI et un fichier \_\_init\_\_.py.

social/social\_api/social\_domains/models: dossier permettant de définir les modèles nécessaire pour toutes les applications.

social/social\_api/social\_db: dossier regroupant la configuration de la db, les entités, les accès,...





# Architecture backend :



social/social-api/social-dto: dossier contenant les objets de transfert de données entre le serveur et l'API.

social/social-api/social\_services: dossier contenant les services et mappers entre les dto et les modèles.

social/social-api/common: dossier contenant les regroupant les éléments en commun entre les différentes parties de l'API et éviter les duplications, il contient des fichiers comme utils.py ou permissions.py.

social/social-api/tests: dossier contenant les tests.



# Architecture DB :

