

API et autres

Pas de problème, on reste **minimaliste et efficace** : juste le partage de fichiers/dossiers, avec une intégration simple pour gérer les utilisateurs depuis ton app existante.

1. Intégration des Utilisateurs via l'API REST de Seafile

Seafile propose une **API REST complète** pour gérer les utilisateurs, les groupes et les permissions. C'est la solution la plus propre pour automatiser la création de comptes depuis ton app.

a. Documentation de l'API

- **Lien officiel** : [Seafile API v2.1](#)
 - **Authentification** : Utilise un **token admin** (à générer depuis l'interface web).
-

b. Étapes pour Utiliser l'API

Étape 1 : Générer un Token Admin

1. Connecte-toi à Seafile en tant qu'admin (`https://seafile.tacampagne.fr`).
2. Va dans **Profil** → **Générer un token API**.
3. Copie ce token (ex: `abc123...xyz`), il servira pour toutes les requêtes API.

Étape 2 : Créer un Utilisateur via l'API

Exemple en `curl` :

```
curl -v -H "Authorization: Token abc123...xyz" \  
-H "Content-Type: application/json" \  
-d '{"email": "user1@tacampagne.fr", "password": "motdepasse123", "name": "Jean Dupont"}' \  
https://seafile.tacampagne.fr/api2/accounts/
```

- **Réponse réussie :**

```
{"email": "user1@tacampagne.fr", "name": "Jean Dupont"}
```

Étape 3 : Créer un Groupe et Ajouter des Utilisateurs

1. **Créer un groupe** (ex: "Membres Campagne") :

```
curl -v -H "Authorization: Token abc123...xyz" \  
-H "Content-Type: application/json" \  
-d '{"name": "Membres Campagne"}' \  
https://seafile.tacampagne.fr/api2/groups/
```

2. **Ajouter un utilisateur au groupe :**

```
curl -v -H "Authorization: Token abc123...xyz" \  
-X PUT \  

```

```
https://seafile.tacampagne.fr/api2/groups/Membres%20Campagne/members/user1@tacampagne.fr/
```

Étape 4 : Partager un Dossier avec un Groupe

1. **Récupérer l'ID de la bibliothèque (library) :**

```
curl -H "Authorization: Token abc123...xyz" \  
https://seafile.tacampagne.fr/api2/repos/
```

- Exemple de réponse :

```
[  
  {  
    "id": "a1b2c3d4-5678-90ef-ghij-klmnopqrstuv",  
    "name": "Documents Campagne",  
    "permission": "rw"  
  }  
]
```

2. **Partager le dossier avec le groupe :**

```
curl -v -H "Authorization: Token abc123...xyz" \  
-H "Content-Type: application/json" \  
-d '{"share_type": "group", "group": "Membres Campagne", "permission": "rw"}' \  
https://seafile.tacampagne.fr/api2/repos/a1b2c3d4-5678-90ef-ghij-  
klmnopqrstuv/shares/
```

c. Exemple en Python (pour ton app)

Si ton app est en Python, voici un script pour créer des utilisateurs depuis une liste :

```
import requests  
import json  
  
# Config  
SEAFILE_URL = "https://seafile.tacampagne.fr"  
API_TOKEN = "abc123...xyz" # Remplace par ton token  
HEADERS = {"Authorization": f"Token {API_TOKEN}", "Content-Type": "application/json"}  
  
# Liste des utilisateurs (ex: extraite de ton app)
```

```
users = [
    {"email": "user1@tacampagne.fr", "password": "motdepasse1", "name": "Jean Dupont"},
    {"email": "user2@tacampagne.fr", "password": "motdepasse2", "name": "Marie Martin"}
]

# Créer les utilisateurs
for user in users:
    response = requests.post(
        f"{SEAFILE_URL}/api2/accounts/",
        headers=HEADERS,
        data=json.dumps(user)
    )
    if response.status_code == 200:
        print(f"Utilisateur {user['email']} créé avec succès.")
    else:
        print(f"Erreur pour {user['email']}: {response.text}")

# Créer un groupe et ajouter les utilisateurs
group_name = "Membres Campagne"
requests.post(
    f"{SEAFILE_URL}/api2/groups/",
    headers=HEADERS,
    data=json.dumps({"name": group_name})
)

for user in users:
    requests.put(
        f"{SEAFILE_URL}/api2/groups/{group_name}/members/{user['email']}/",
        headers=HEADERS
    )
```

2. Alternative :

Synchronisation via Base de Données

Si tu préfères éviter l'API (ou si ton app tourne sur le même serveur), tu peux **interagir directement avec la base de données MariaDB** de Seafile. **Attention** : Cette méthode est **moins recommandée** (risque de corruption si mal utilisée), mais elle est possible.

a. Structure de la Base de Données

- **Table EmailUser** : Stocke les utilisateurs.

```
SELECT * FROM EmailUser;
```

Champ	Description
email	Email de l'utilisateur
password	Mot de passe (hashé)
name	Nom complet
is_staff	1 = admin, 0 = utilisateur normal

- **Table** `GroupUser` : Stocke les groupes.
- **Table** `GroupUser_Group` : Lie les utilisateurs aux groupes.

b. Exemple de Requête SQL

Pour ajouter un utilisateur :

```
INSERT INTO EmailUser (email, password, name, is_staff)
VALUES ('user1@tacampagne.fr', 'motdepasse_hashé', 'Jean Dupont', 0);
```

Problème : Le mot de passe doit être **hashé** (Seafile utilise un algorithme spécifique). Pour le générer, utilise l'API ou le script Python ci-dessus.

3. Automatisation avec un Script Bash

Si tu veux un script **simple et robuste** pour synchroniser tes utilisateurs depuis un fichier CSV (ex: exporté depuis ton app) :

```
#!/bin/bash
API_TOKEN="abc123...xyz"
SEAFILE_URL="https://seafile.tacampagne.fr"
CSV_FILE="users.csv" # Format: email,password,name

while IFS=, read -r email password name; do
    # Créer l'utilisateur
    curl -s -H "Authorization: Token $API_TOKEN" \
        -H "Content-Type: application/json" \
        -d "{\"email\": \"$email\", \"password\": \"$password\", \"name\": \"$name\"}" \
        "$SEAFILE_URL/api2/accounts/" | grep -q "email"

    if [ $? -eq 0 ]; then
        echo "Utilisateur $email créé."
        # Ajouter au groupe "Membres Campagne"
        curl -s -X PUT -H "Authorization: Token $API_TOKEN" \
            "$SEAFILE_URL/api2/groups/Membres%20Campagne/members/$email/"
    else

```

```
echo "Erreur pour $email."  
fi  
done < "$CSV_FILE"
```

4. Bonnes Pratiques

1. Gère les mots de passe :

- Génère des mots de passe aléatoires pour les utilisateurs et envoie-les par email (ex: avec `sendmail` ou un service comme Mailgun).
- Ou utilise un **lien d'invitation** (Seafile permet d'envoyer des invitations par email).

2. Sécurise ton token API :

- Ne le stocke **jamais en clair** dans ton code. Utilise des variables d'environnement ou un gestionnaire de secrets (ex: `dotenv`).
- Restreins les permissions du token (ex: uniquement création d'utilisateurs).

3. Logs et monitoring :

- Active les logs de l'API Seafile pour déboguer :
`docker-compose logs -f seafile | grep api2`

4. Teste en staging :

- Déploie une instance Seafile de test pour valider ton script avant de l'exécuter en production.
-

5. Exemple Complet : Intégration avec ton App

Scénario :

- Ton app gère une liste de contacts (ex: `contacts.csv`).
- Tu veux :
 1. Créer un utilisateur Seafile pour chaque contact.
 2. Les ajouter au groupe "Membres Campagne".
 3. Partager un dossier "Documents Campagne" avec ce groupe.

Étapes :

1. **Exporte tes contacts** depuis ton app au format CSV :

```
email,password,name
```

```
user1@tacampagne.fr,motdepasse1,Jean Dupont
```

```
user2@tacampagne.fr,motdepasse2,Marie Martin
```

2. **Exécute le script Python** (ou Bash) pour créer les utilisateurs et les groupes.
3. **Partage le dossier** via l'API (comme montré plus haut).

6. Résumé des Commandes Utiles

Action	Commande curl
Lister les utilisateurs	<pre>curl -H "Authorization: Token ABC" "\$SEAFILE_URL/api2/accounts/"</pre>
Créer un utilisateur	<pre>curl -H "Authorization: Token ABC" -d '{"email": "...", "password": "...", ...}' "\$SEAFILE_URL/api2/accounts/"</pre>
Créer un groupe	<pre>curl -H "Authorization: Token ABC" -d '{"name": "Groupe"}' "\$SEAFILE_URL/api2/groups/"</pre>
Ajouter un utilisateur à un groupe	<pre>curl -X PUT -H "Authorization: Token ABC" "\$SEAFILE_URL/api2/groups/Groupe/members/email/"</pre>
Partager un dossier	<pre>curl -H "Authorization: Token ABC" -d '{"share_type": "group", "group": "Groupe", "permission": "rw"}' "\$SEAFILE_URL/api2/repos/ID_DU_DOSSIER/shares/"</pre>

7. Prochaines Étapes

1. Veux-tu que je t'aide à adapter le script Python/Bash pour ton format de données spécifique ?
2. As-tu besoin d'un exemple pour générer des mots de passe aléatoires et les envoyer par email ?
3. Veux-tu tester l'API en direct avec un exemple de `curl` pour vérifier que tout fonctionne ?

Dis-moi ce qui te serait le plus utile, et on avance ! ☐☐

Revision #1

Created 2026-03-03 17:29:32 UTC by Nicolas Farrie

Updated 2026-03-03 17:29:45 UTC by Nicolas Farrie