Exemples de Requêtes API

Ce document contient des exemples de requêtes pour tous les endpoints de l'API.

Table des Matières

Authentification

1. Inscription

```bash
curl -X POST "http://localhost:8000/auth/register" \
  -H "Content-Type: application/json" \
  -d '{
    "mail": "user@example.com",
    "mot_de_passe": "password123"
  }'
```

Réponse:

```json
{
  "id": 1,
  "mail": "user@example.com",
  "created_at": "2024-12-17T10:30:00"
}
```

2. Connexion

```bash
```

```bash
curl -X POST "http://localhost:8000/auth/login" \
  -H "Content-Type: application/json" \
  -d '{
    "mail": "user@example.com",
    "mot_de_passe": "password123"
  }'
```

Réponse:

```json
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "bearer",
  "user": {
    "id": 1,
    "mail": "user@example.com"
  }
}
```

3. Informations Utilisateur

```bash
curl -X GET "http://localhost:8000/auth/me" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

4. Changer le Mot de Passe

```bash
curl -X POST "http://localhost:8000/auth/change-password" \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "ancien_mot_de_passe": "password123",
    "nouveau_mot_de_passe": "newpassword456"
  }'
```

5. Changer l'Email

```bash
```

```bash
curl -X POST "http://localhost:8000/auth/change-mail" \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "nouveau_mail": "newemail@example.com",
    "mot_de_passe": "password123"
  }'
```

Upload

1. Upload Fichier Excel

bash

```bash
curl -X POST "http://localhost:8000/upload/excel" \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -F "file=@/path/to/your/file.xlsx"
```

Réponse:

json

```json
{
  "message": "Fichier importé avec succès",
  "filename": "materiels.xlsx",
  "lignes_inserees": 125,
  "date_import": "2024-12-17",
  "id_date_import": 5
}
```

2. Historique des Uploads

bash

```bash
curl -X GET "http://localhost:8000/upload/history?skip=0&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Réponse:

json

```
{
  "total": 15,
  "skip": 0,
  "limit": 10,
  "data": [
    {
      "id_upload": 15,
      "filename": "materiels_dec.xlsx",
      "upload_date": "2024-12-17T14:30:00",
      "user_mail": "user@example.com"
    }
  ]
}
```


3. Liste des Dates d'Importation

bash

```
curl -X GET "http://localhost:8000/upload/dates" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Réponse:

json


```
{
  "total": 5,
  "dates": [
    {
      "id_date": 5,
      "date_complet": "2024-12-17"
    },
    {
      "id_date": 4,
      "date_complet": "2024-12-10"
    }
  ]
}
```

Statistiques

1. Statistiques Complètes

bash

```
curl -X GET "http://localhost:8000/statistics/?id_date_import=5&skip_type=0&limit_type=10&skip_region=0&limit_region=
  -H "Authorization: Bearer YOUR_TOKEN"
```

Réponse (exemple):

json

```
{
  "nouveau_materiel": 25,
  "materiel_perdu": 3,
  "top_5_districts_pannes": [
    {
      "code": "630601",
      "district": "MOROMBE",
      "nombre_pannes": 42,
      "taux_pannes": 35.5,
      "total_materielle": 118
    }
  ],
  "pannes_par_type_materiel": [
    {
      "type": "Imprimante",
      "nombre_pannes": 64
    },
    {
      "type": "Ordinateur",
      "nombre_pannes": 31
    }
  ],
  "materiels_par_region": [
    {
      "code": "630601",
      "region": "ATSIMO ANDREFANA",
      "total_materiels": 250,
      "taux_fonctionnel": 82.4
    }
  ],
  "etat_6_dernieres_importations": [
    {
      "date_importation": "2024-11-15",
      "fonctionnels": 450,
      "non_fonctionnels": 95
    },
    {
      "date_importation": "2024-12-17",
      "fonctionnels": 475,
      "non_fonctionnels": 98
    }
  ],
  "resume_global": {
```

```
    "total_materiels": 573,
    "materiels_fonctionnels": 475,
    "materiels_en_panne": 98,
    "taux_fonctionnement": 82.9,
    "taux_en_panne": 17.1
  }
}
```

2. Dashboard (Dernière Importation)

bash

```
curl -X GET "http://localhost:8000/statistics/dashboard" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Matériels

1. Liste Tous les Matériels

bash

```
curl -X GET "http://localhost:8000/materiels/all?id_date_import=5&skip=0&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Réponse:

json

```
{
  "total": 573,
  "skip": 0,
  "limit": 10,
  "data": [
    {
      "id_snapshot": 1234,
      "id_physique": 456,
      "etat": "Fonctionnel",
      "nom_materiel": "Imprimante 1",
      "type": "Imprimante",
      "code": "630601",
      "region": "ATSIMO ANDREFANA",
      "district": "MOROMBE",
      "commune": "A",
      "date_import": "2024-12-17"
    }
  ]
}
```


2. Matériels par Commune

bash

```
curl -X GET "http://localhost:8000/materiels/by-commune?id_date_import=5&commune=Ambahita&skip=0&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

3. Nouveaux Matériels Entre Deux Dates

bash

```
curl -X GET "http://localhost:8000/materiels/nouveaux?date_ancienne=4&date_nouvelle=5&skip=0&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Réponse:

json

```
{
  "total": 25,
  "date_ancienne": 4,
  "date_nouvelle": 5,
  "skip": 0,
  "limit": 10,
  "data": [
    {
      "id_snapshot": 1500,
      "id_physique": 789,
      "etat": "Fonctionnel",
      "nom_materiel": "Routeur 1",
      "type": "Routeur",
      "code": "620301",
      "region": "ANOSY",
      "district": "BETROKA",
      "commune": "Nouvelle Commune",
      "date_import": "2024-12-17"
    }
  ]
}
```


4. Détails d'un Matériel

bash

```
curl -X GET "http://localhost:8000/materiels/1234" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Réponse:

json

```
{
  "id_snapshot": 1234,
  "id_physique": 456,
  "etat": "Non fonctionnel",
  "nom_materiel": "Imprimante 2",
  "type": "Imprimante",
  "code": "610201",
  "region": "ANDROY",
  "district": "BEKILY",
  "commune": "Ambahita",
  "date_import": "2024-12-17",
  "motif": "ba. Problème cartouche",
  "achat_consommable": "ENY",
  "compatibilite_consommable": "Mety taminy"
}
```


5. Recherche par Code

```bash
# Avec date spécifique
curl -X GET "http://localhost:8000/materiels/search/by-code?code=630601&id_date_import=5&skip=0&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN"

# Sans date (toutes les dates)
curl -X GET "http://localhost:8000/materiels/search/by-code?code=630601&skip=0&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN"
```

Exemples Python

Exemple Complet avec Requests

```python
```

```python
import requests

BASE_URL = "http://localhost:8000"

# 1. Se connecter
login_response = requests.post(
    f"{BASE_URL}/auth/login",
    json={
        "mail": "user@example.com",
        "mot_de_passe": "password123"
    }
)

token = login_response.json()["access_token"]
headers = {"Authorization": f"Bearer {token}"}

# 2. Upload fichier
with open("materiels.xlsx", "rb") as f:
    upload_response = requests.post(
        f"{BASE_URL}/upload/excel",
        files={"file": f},
        headers=headers
    )
    print(upload_response.json())

# 3. Récupérer statistiques
stats_response = requests.get(
    f"{BASE_URL}/statistics/dashboard",
    headers=headers
)
print(stats_response.json())

# 4. Lister matériels
materiels_response = requests.get(
    f"{BASE_URL}/materiels/all",
    params={"id_date_import": 5, "skip": 0, "limit": 20},
    headers=headers
)
print(materiels_response.json())
```

Exemple avec httpx (async)

python

```python
import httpx
import asyncio

async def main():
    async with httpx.AsyncClient() as client:
        # Login
        login_response = await client.post(
            "http://localhost:8000/auth/login",
            json={
                "mail": "user@example.com",
                "mot_de_passe": "password123"
            }
        )

        token = login_response.json()["access_token"]
        headers = {"Authorization": f"Bearer {token}"}

        # Get statistics
        stats = await client.get(
            "http://localhost:8000/statistics/dashboard",
            headers=headers
        )

        print(stats.json())

asyncio.run(main())
```

Codes d'Erreur Courants

400 Bad Request

```json
{
  "detail": "Cet email est déjà utilisé"
}
```

401 Unauthorized

```json
```

```
{
  "detail": "Email ou mot de passe incorrect"
}
```

404 Not Found

json

```
{
  "detail": "Date d'importation non trouvée"
}
```

422 Unprocessable Entity

json

```
{
  "error": "Validation Error",
  "message": "Les données fournies ne sont pas valides",
  "details": [...]
}
```

500 Internal Server Error

json

```
{
  "error": "Internal Server Error",
  "message": "Une erreur inattendue s'est produite"
}
```