

Documentation Backend - Système de Gestion des Achats

Table des matières

1. Vue d'ensemble
2. Architecture
3. Installation et configuration
4. Modèles de données
5. API Endpoints
6. Authentification
7. Services externes
8. Déploiement
9. Tests

Vue d'ensemble

Ce projet Django REST Framework implémente un système de gestion des demandes d'achat avec workflow de validation multi-niveaux.

Fonctionnalités principales

- Gestion des utilisateurs avec rôles (Employé, Moyens Généraux, Comptabilité, Direction)
- Système de demandes d'achat avec workflow de validation
- Authentification JWT avec cookies HttpOnly
- Notifications email automatiques
- Upload de fichiers (Cloudinary)
- Dashboard avec statistiques

Stack technique

- Backend: Django 5.2 + Django REST Framework
- Base de données: PostgreSQL
- Authentification: JWT (SimpleJWT)
- Email: Mailjet
- Stockage fichiers: Cloudinary
- Cache/Session: Cookies HttpOnly

Architecture

Structure du projet

```
simplonservice/
  core/ # Application principale
    models.py # Modèles de données
    views.py # Vues API
    serializers.py # Sérialiseurs DRF
    urls.py # URLs de l'API
    authentication.py # Classes d'authentification
    jwt_views.py # Vues JWT personnalisées
    jwt_serializers.py # Sérialiseurs JWT
    admin.py # Interface admin
    signals.py # Signaux Django
  services/
```

```
■ ■■■■ email_service.py # Service d'envoi d'emails  
■■■ simplonservice/  
■■■ settings.py # Configuration  
■■■ urls.py # URLs racine
```

Installation et configuration

Prérequis

- Python 3.9+
- PostgreSQL
- Compte Mailjet
- Compte Cloudinary

Variables d'environnement

```
SECRET_KEY=your_secret_key  
DEBUG=False  
DATABASE_NAME=your_db_name  
DATABASE_USER=your_db_user  
DATABASE_PASSWORD=your_db_password  
DATABASE_HOST=your_db_host  
DATABASE_PORT=5432  
ALLOWED_HOSTS=your-domain.com,localhost  
CORS_ALLOWED_ORIGINS=https://your-frontend.com  
CSRF_TRUSTED_ORIGINS=https://your-frontend.com  
JWT_COOKIE_DOMAIN=your-domain.com  
MAILJET_API_KEY=your_mailjet_key  
MAILJET_SECRET_KEY=your_mailjet_secret  
CLOUDINARY_API_KEY=your_cloudinary_key  
CLOUDINARY_API_SECRET=your_cloudinary_secret  
CLOUDINARY_CLOUD_NAME=your_cloudinary_name  
FRONTEND_URL=https://your-frontend.com  
COMPANY_NAME=Your Company  
DEFAULT_FROM_EMAIL=noreply@your-domain.com  
DEFAULT_FROM_NAME=Your Company
```

Installation

```
pip install -r requirements.txt  
python manage.py migrate  
python manage.py createsuperuser  
python manage.py runserver
```

Modèles de données

CustomUser

Utilisateur personnalisé avec rôles et informations supplémentaires.

Champs principaux:

- role: employee, mg, accounting, director
- department: département de l'utilisateur
- phone: numéro de téléphone
- created_by: utilisateur qui a créé ce compte

PurchaseRequest

Demande d'achat avec workflow de validation.

Statuts possibles:

- pending: En attente (validation MG)
- mg_approved: Validée par Moyens Généraux
- accounting_reviewed: Étudiée par Comptabilité
- director_approved: Approuvée par Direction
- rejected: Refusée

Workflow:

1. Employé → Moyens Généraux
2. MG → Comptabilité
3. Comptabilité → Direction
4. Direction → Approuvé

RequestStep

Historique des étapes de validation d'une demande.

Attachment

Pièces jointes liées aux demandes (devis, factures, etc.).

PasswordResetCode

Codes de réinitialisation de mot de passe (5 chiffres, expire en 5min).

UserActivity

Journal des activités utilisateurs (création, modification, etc.).

API Endpoints

Authentification

```
POST /api/auth/login/  
POST /api/auth/refresh/  
POST /api/auth/logout/  
GET /api/auth/me/  
POST /api/auth/change-password/  
POST /api/auth/password-reset/request/  
POST /api/auth/password-reset/verify/  
POST /api/auth/password-reset/confirm/
```

Utilisateurs

```
POST /api/auth/register/  
GET /api/users/  
GET /api/users/stats/  
GET /api/users/{id}/  
PUT /api/users/{id}/
```

Demandes d'achat

```
GET /api/requests/
```

POST /api/requests/
GET /api/requests/{id}/
PUT /api/requests/{id}/
POST /api/requests/{id}/validate/
PUT /api/requests/{id}/update-rejection/

Pièces jointes

GET /api/attachments/
POST /api/attachments/
DELETE /api/attachments/{id}/

Dashboard

GET /api/dashboard/

Authentification

Système JWT avec Cookies

- Access Token: 1h, stocké en cookie HttpOnly
- Refresh Token: 7 jours, rotation automatique
- Support connexion par username ou email

Classes d'authentification

- CookieJWTAuthentication
- CookieJWTMiddleware
- EmailOrUsernameModelBackend

Exemple frontend fetch avec credentials: include

Services externes

Service Email (Mailjet)

- send_welcome_email()
- send_password_reset_code()
- send_purchase_request_notification()

Stockage Cloudinary

Configuration dans settings.py

Permissions et sécurité

Rôles: employee, mg, accounting, director

Configuration développement vs production

DEBUG = True en dev, False en prod

Déploiement

Checklist: env vars, migrations, collectstatic, tests

Tests

Tests: auth, CRUD, workflow, services, serializers

Maintenance et monitoring

Logs et surveillance (emails, DB, Cloudinary)

Troubleshooting courant

- Authentification: cookies, CORS
- Email: clés, from_email
- Upload: config Cloudinary

Contact: équipe dev

Version: 1.0

Dernière mise à jour: 19 Septembre 2025