

# Documentação Completa do Projeto Django\_Platform\_AI

## Visão Geral do Projeto

O **Django\_Platform\_AI** é uma aplicação web desenvolvida com o framework Django, que integra funcionalidades de inteligência artificial para aprimorar a interação com os usuários.

## Tecnologias Utilizadas

- **Django**: Framework web para desenvolvimento backend.
- **PostgreSQL**: Banco de dados relacional utilizado na aplicação.
- **Django Rest Framework (DRF)**: Implementação de APIs REST.
- **S3 da AWS**: Armazenamento de arquivos na nuvem.
- **Selenium e Flask**: Utilizados em integrações e automação.
- **LangChain e OpenAI**: Ferramentas para processamento de linguagem natural (NLP) e IA.

## Estrutura do Projeto

O projeto é modularizado em diferentes aplicativos Django, cada um responsável por uma funcionalidade específica:

- **ai\_chats/**: Gerencia as interações de chat baseadas em IA.
- **companies/**: Lida com operações relacionadas a empresas.
- **core/**: Contém configurações e funcionalidades compartilhadas.
- **files/**: Responsável pelo gerenciamento de arquivos na plataforma.
- **groups\_files/**: Gerencia arquivos associados a grupos específicos.
- **text\_ai/**: Implementa funcionalidades de processamento de linguagem natural.
- **users/**: Gerencia autenticação, perfis e permissões de usuários.

# Guia de Instalação e Execução

## 1. Clonar o Repositório

```
$ git clone https://github.com/Declo/Django_Platform_AI.git
$ cd Django_Platform_AI
```

## 2. Criar e Ativar um Ambiente Virtual

```
$ python -m venv venv
$ source venv/bin/activate # Linux/Mac
$ venv\Scripts\activate    # Windows
```

## 3. Instalar Dependências

```
$ pip install -r requirements.txt
```

## 4. Configurar o Banco de Dados PostgreSQL

- Criar um banco de dados no PostgreSQL.
- Configurar as credenciais no arquivo `.env`.

## 5. Executar Migrações

```
$ python manage.py migrate
```

## 6. Criar Superusuário (Opcional)

```
$ python manage.py createsuperuser
```

## 7. Iniciar o Servidor

```
$ python manage.py runserver
```

A aplicação estará disponível em `http://127.0.0.1:8000/`.

# Rotas Disponíveis

## 1. ai\_chats/

- GET /ai\_chats/: Lista todas as interações de chat.
- POST /ai\_chats/: Cria uma nova interação de chat.
- GET /ai\_chats/{id}/: Exibe detalhes de uma interação específica.
- PUT /ai\_chats/{id}/: Atualiza uma interação existente.
- DELETE /ai\_chats/{id}/: Remove uma interação.

## 2. companies/

- GET /companies/: Lista todas as empresas.
- POST /companies/: Adiciona uma nova empresa.
- GET /companies/{id}/: Exibe detalhes de uma empresa específica.
- PUT /companies/{id}/: Atualiza informações de uma empresa.
- DELETE /companies/{id}/: Remove uma empresa.

## 3. files/

- GET /files/: Lista todos os arquivos.
- POST /files/: Faz upload de um novo arquivo.
- GET /files/{id}/: Exibe detalhes de um arquivo específico.
- PUT /files/{id}/: Atualiza informações de um arquivo.
- DELETE /files/{id}/: Remove um arquivo.

## 4. groups\_files/

- GET /groups\_files/: Lista todos os grupos de arquivos.
- POST /groups\_files/: Cria um novo grupo de arquivos.
- GET /groups\_files/{id}/: Exibe detalhes de um grupo específico.
- PUT /groups\_files/{id}/: Atualiza informações de um grupo.
- DELETE /groups\_files/{id}/: Remove um grupo de arquivos.

## 5. text\_ai/

- POST /text\_ai/process/: Processa texto utilizando IA.

## 6. users/

- GET /users/: Lista todos os usuários.
- POST /users/: Cria um novo usuário.
- GET /users/{id}/: Exibe detalhes de um usuário específico.
- PUT /users/{id}/: Atualiza informações de um usuário.
- DELETE /users/{id}/: Remove um usuário.

## Estrutura do Banco de Dados e Relacionamentos

Cada aplicativo implementa operações CRUD para gerenciar suas respectivas entidades. As relações no banco de dados são estabelecidas utilizando os modelos do Django (`models.py`).

### Exemplo de Relacionamentos

- **Usuários e Empresas:** Um usuário pode estar associado a uma ou mais empresas (Muitos-para-Muitos).
- **Arquivos e Grupos:** Um arquivo pode pertencer a um grupo específico (Um-para-Muitos).
- **Chats e Usuários:** Cada interação de chat está vinculada a um usuário (Muitos-para-Um).

## Considerações Finais

O **Django\_Platform\_AI** é uma plataforma robusta que integra IA e funcionalidades de gerenciamento de arquivos, chats e usuários de forma modular.

Para mais detalhes, consulte o repositório oficial no GitHub: [Django\\_Platform\\_AI](#).