

ming-ming-api-cc0464

Este guia apresenta de forma coesa as etapas necessárias para configurar o projeto Django **Ming-ming-tests-django**, integrando as bibliotecas **branca**, **django** e **folium**, além de preparar a aplicação **entregas**.

1. Pré-requisitos

- **Python 3** instalado.
- **pip** disponível no sistema.

2. Instalação de Dependências

Execute no terminal:

```
pip install -q branca django folium
```

3. Criação do Projeto Django

No diretório raiz do seu projeto, inicie o projeto com:

```
django-admin startproject magalu .
```

4. Criação da Aplicação "entregas"

Ainda no diretório do projeto, crie o aplicativo responsável pela lógica de rotas e visualização de mapas:

```
python3 manage.py startapp entregas
```

5. Estrutura de Diretórios

Organize templates e arquivos estáticos com os comandos:

```
mkdir -p entregas/templates/entregas
mkdir -p entregas/static/entregas/css
mkdir -p entregas/static/entregas/js
mkdir -p entregas/static/entregas/images
```

Por fim, criação de um arquivo `urls.py` dentro da pasta `entregas`.

6. Configurações no `settings.py`

Abra `magalu/settings.py` e ajuste:

1. **INSTALLED_APPS**:

```
INSTALLED_APPS = [
    # ... outras apps do Django
    'entregas.apps.EntregasConfig',
]
```

2. **Idioma da interface de administração** (para Português do Brasil):

```
LANGUAGE_CODE = 'pt-br'
```

7. Verificação de Configuração

Certifique-se de que não há erros:

```
python3 manage.py check
```

8. Aplicação de Migrações

Aplique as migrações padrões do Django:

```
python3 manage.py migrate
```

9. Criação do Superusuário

Para acessar o Django Admin, crie um superusuário:

```
python3 manage.py createsuperuser
```

10. Execução do Servidor de Desenvolvimento

Inicie o servidor local e verifique no navegador:

```
python3 manage.py runserver
```

Acesse <http://127.0.0.1:8000/> ; se a página inicial do Django aparecer, a configuração inicial foi concluída com sucesso.

11. Configurações Adicionais: Hosts Permitidos e Origens Confiáveis para CSRF

Em alguns cenários (por exemplo, ao usar GitHub Pages ou integração via HTTPS em ambiente local), pode ocorrer o erro **403 - CSRF Verification Failed** com a mensagem:

Origin checking failed – <https://localhost:8000> does not match any trusted origins.

Para corrigir, adicione ao seu `settings.py` :

```
# Hosts permitidos (útil também para DEBUG=False)
ALLOWED_HOSTS = ['localhost', '127.0.0.1']

# Origens confiáveis para CSRF (inclua sempre o esquema)
CSRF_TRUSTED_ORIGINS = [
    'http://localhost:8000',
    'https://localhost:8000',
]
```

Observação: esse ajuste foi necessário apenas na integração via GitHub (HTTPS) em ambiente de desenvolvimento local.
