Criando Variáveis Globais no Django

Introdução

No desenvolvimento de aplicações Django, muitas vezes é necessário criar variáveis que possam ser acessadas em diferentes partes do sistema. Essas variáveis, conhecidas como "variáveis globais", são úteis para armazenar configurações ou dados que permanecem consistentes durante a execução da aplicação. Nesta apostila, você aprenderá a configurar variáveis globais no Django, além de entender boas práticas para seu uso.

---

1. O que são variáveis globais?

Variáveis globais são aquelas que podem ser acessadas por qualquer parte da aplicação sem a necessidade de serem passadas como argumentos. No Django, elas podem ser usadas para armazenar:

- Configurações customizadas (como URLs externas ou chaves de API);
- Constantes compartilhadas;
- Dados específicos para sessões ou contextos globais.

---

2. Métodos para Criar Variáveis Globais no Django

Existem várias abordagens para criar variáveis globais no Django. A seguir, apresentamos as principais.

2.1 Usando o arquivo settings.py

O Django já possui um local central para configurações: o arquivo settings.py. Você pode definir variáveis globais diretamente nele.

Passos:

1. **Abra o arquivo settings.py do seu projeto.**
2. **Adicione sua variável global:**

python
Copiar código
```
# Exemplo de variável global
API_KEY = "123456789abcdef"
MAX_UPLOAD_SIZE = 10485760  # 10 MB
```

3. **Acesse a variável em qualquer lugar no projeto:**

No código Python:

python
Copiar código
```
from django.conf import settings

def my_function():
    print(settings.API_KEY)
```

2.2 Usando um módulo Python dedicado

Se você deseja separar as variáveis globais para não sobrecarregar o settings.py, crie um módulo próprio.

Passos:

1. Crie um arquivo chamado globals.py na raiz do seu projeto ou dentro de um app específico.

python
Copiar código
```
# globals.py
API_KEY = "123456789abcdef"
SITE_NAME = "Meu Site Django"
```

2. Importe e use a variável onde necessário:

python
Copiar código
```
from myproject.globals import API_KEY

def my_function():
    print(API_KEY)
```

2.3 Usando Middleware para Variáveis Globais no Contexto de Requisição

Se suas variáveis precisam estar disponíveis em templates ou depender do contexto de requisição, você pode usar middleware ou context_processors.

Usando Context Processors:

1. Crie um arquivo chamado context_processors.py em um app.

python
Copiar código
```
# context_processors.py
def global_variables(request):
    return {
        'SITE_NAME': "Meu Site Django",
        'SUPPORT_EMAIL': "support@meusite.com",
    }
```

2. Adicione o context processor no settings.py:

```
python
Copiar código
TEMPLATES = [
  {
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [],
    'APP_DIRS': True,
    'OPTIONS': {
      'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
        'meuapp.context_processors.global_variables',  # Adicione aqui
      ],
    },
  },
]
```

3. Acesse no template HTML:

```
html
Copiar código
<p>Bem-vindo ao {{ SITE_NAME }}</p>
<p>Para suporte, envie um e-mail para {{ SUPPORT_EMAIL }}</p>
```

---

3. Boas Práticas

1. **Evite usar variáveis globais para dados sensíveis diretamente no código.**
   Use variáveis de ambiente ou bibliotecas como django-environ para proteger informações como senhas e chaves de API.
2. **Documente suas variáveis globais.**
   Mantenha uma descrição clara no código para facilitar o entendimento e manutenção.
3. **Limite o uso de variáveis globais.**
   Prefira usar funções ou classes para encapsular lógica, mantendo a aplicação modular e fácil de testar.
4. **Verifique a segurança.**
   Ao expor variáveis em templates, tenha certeza de que não está vazando dados confidenciais.

---

4. Exercício Prático

Vamos criar uma variável global para o nome do site e utilizá-la em diferentes partes da aplicação.

1. **Defina no settings.py:**

python
Copiar código
SITE_NAME = "Minha Loja Online"

2. Crie um view para exibir o nome do site:

python
Copiar código
from django.conf import settings
from django.http import HttpResponse

def site_name_view(request):
    return HttpResponse(f"Bem-vindo ao {settings.SITE_NAME}")

3. Adicione a variável ao contexto do template:

No context_processors.py:

python
Copiar código
def site_name(request):
    return {'SITE_NAME': settings.SITE_NAME}

No settings.py:

python
Copiar código
TEMPLATES[0]['OPTIONS']['context_processors'].append('meuapp.context_processors.site_name')

No template:

html
Copiar código
<h1>Bem-vindo ao {{ SITE_NAME }}</h1>

Conclusão

Criar variáveis globais no Django é uma tarefa simples e essencial para centralizar configurações e constantes do sistema. Escolha a abordagem que melhor se adapta às necessidades do seu projeto, seguindo boas práticas para garantir segurança e manutenção eficiente.