CRUD SIMPLES UTILIZANDO PYTHON E DJANGO

Antônio Fernandes De Santana Neto
Universidade Tiradentes

20/07/2024

Abstract

Este artigo apresenta uma aplicação simples do acrônimo CRUD (Create Read Update Delete) utilizando Python como linguagem de programação e um framework seu para desenvolvimento web, Django.
Palavras chave: CRUD, Python, Django, Web, Aplicação.

This article presents a simple application of the acronym CRUD (Create Read Update Delete) using Python as a programming language and its own framework for web development, Django.
Keywords: CRUD, Python, Django, Web, Application.

1 Introdução

No contexto de aplicações web, manipular informações com maestria é de suma importância para o funcionamento preciso de qualquer serviço online. A um baixo nível de abstração podemos dizer que o propósito de qualquer persistencia de dados podem ser resumidas em 4 opreações: Criar, Ler, Atualizar e Deletar, respectivamente traduzidas em Create, Read, Update, Delete.

"CRUD (Criar, Ler, Atualizar, Excluir) é um acrônimo para maneiras de operar com dados armazenados. É um mnemônico para as quatro funções básicas do armazenamento persistente. [...]" (Mozila Developer Network).

Este artigo apresenta uma maneira simples de criar um CRUD em poucos minutos utilizando a linguagem de programação Python e uma coleção de ferramentas prontas (Framework) para o desenvolvimento web, Django.

2 Utilidades

A aplicação desenvolvida pode ser usada para maior parte dos cenários, portanto que haja seus respectivas mudanças relacionadas a sua regra de negócio. Com um serviço versátil e facilmente auditável pelo painel admnistrativo nativo, torna as possíbilidades mais variadas e aplícaveis dentro do seu determinado contexto.

3 Metodologia

Será abordado um gerenciamento de usuários, contendo todas as operações do acrônimo CRUD de forma visual, intuitiva e de fácil compreensão. Será criado um "super usuário" que terá a autonômia máxima no sistema, pondendo inclusive gerenciar os outros usuários já presentes.

Uma possibilidade de extensão que estará disponível, todavia não abordado nesse artigo, é a criação de "grupos de usuários", onde é possivel auditar as permissões de todos os outros usuários.

4 Materiais e Métodos

4.1 Sistema Operacional

Neste artigo foi utilizado o sistema operacional Linux (Linux debian 5.10.0-18-amd64 1 SMP Debian 5.10.140-1 (2022-09-02) x86-64).
Mais informações em: https://packages.qa.debian.org/l/linux.html

4.2 Python 3.11.9

Foi utilizado a linguagem de programação Python em sua versão 3.11.9.
Para mais informações: https://www.python.org/downloads/release/python-3119/

5 Implementação

5.1 Criaçao do diretório

Para fins de organização, foi criado uma pasta chamada "crud" e posteriormente adentrado na mesma, como ilustra a imagem a seguir:

```
antonio@debian:~$ mkdir crud
antonio@debian:~$ cd crud
antonio@debian:~/crud$
```

5.2 Ambiente Virtual

Também para fins de organização, será criado um ambiente virtual onde instalaremos o Django e todas suas respectivas dependências.

```
antonio@debian:~/crud$ python3 -m venv venv
antonio@debian:~/crud$ ls
venv
```

Depois de criar o ambiente virtual, iremos ativar o mesmo:

```
antonio@debian:~/crud$ source venv/bin/activate
(venv) antonio@debian:~/crud$
```

5.3 Django

5.3.1 Instalação

Após configurar e acessar o ambiente virtual, instala-se o Django

```
(venv) antonio@debian:~/crud$ pip install django
Collecting django
  Downloading Django-4.2.14-py3-none-any.whl (8.0 MB)
     |████████████████████████████████| 8.0 MB 81 kB/s
Collecting asgiref<4,>=3.6.0
  Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.5.1-py3-none-any.whl (44 kB)
     |████████████████████████████████| 44 kB 297 kB/s
Collecting typing-extensions>=4
  Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-4.2.14 sqlparse-0.5.1 typing-extensions-4.12.2
(venv) antonio@debian:~/crud$
```

5.3.2 Criação da aplicação

Agora com o Django instalado, basta criar o projeto:

```
(venv) antonio@debian:~/crud$ django-admin startproject crud
(venv) antonio@debian:~/crud$
```

5.3.3 Migrations

Com a aplicação já criada, precisamos modelar o banco de dados que armazenará as informações. O Django já traz um recurso para facilitar todo o processo de modelagem, chamado de "Migrations", que basicamente são arquivos estáticos que conseguem salvar os processos de alterações nas tabelas cronologicamente. "Migrations são a maneira do Django de propagar as alterações feitas em seus Models (adicionando um campo, excluindo um modelo, etc.) em seu esquema de banco de dados." (Documentação oficial do Django). Estes "Models" são arquivos criados manualmente pelo desenvolvedor para modular alguma tabela do banco de dados, todavia existem algumas que já vem por padrão criadas, dentre elas, o Model de "User" (usuário). Então podemos aplicar os arquivos de migrations que já vem com o framework e teremos o banco de dados modularizados para auditar usuários.

```
(venv) antonio@debian:~/crud$ python3 crud/manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(venv) antonio@debian:~/crud$
```

5.3.4 Super Usuário

Quando todas as aplicações dos arquivos de Migrations forem aplicados, poderemos criar o "super usuário" que auditará toda e qualquer informação do sistema. Neste caso foi criado com o nome de usuário "root" e senha "root".

```
(venv) antonio@debian:~/crud$ python3 crud/manage.py createsuperuser
Username (leave blank to use 'antonio'): root
Email address: root@root.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(venv) antonio@debian:~/crud$
```

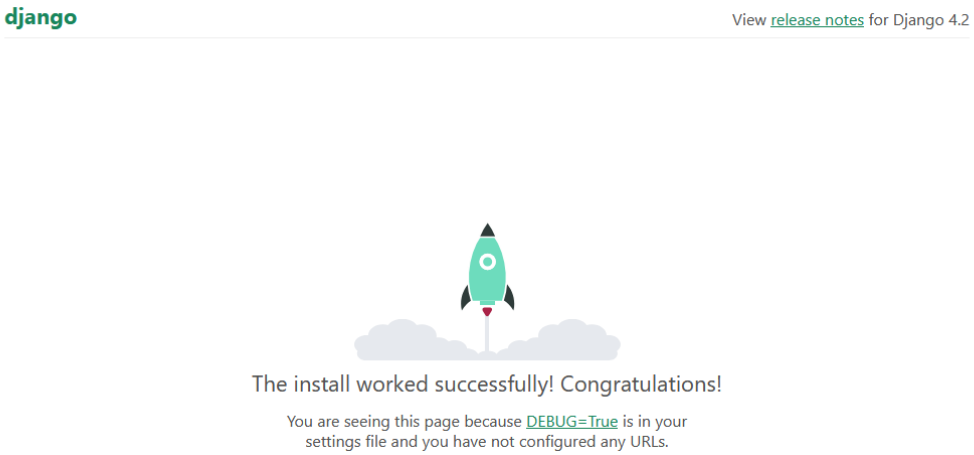
5.3.5 Executar a Aplicação

Depois de seguir todos os passos anteriores, a aplicação está pronta para ser utilizada. Após executada, estará disponível em seu endereço local na porta 8000 por padrão.

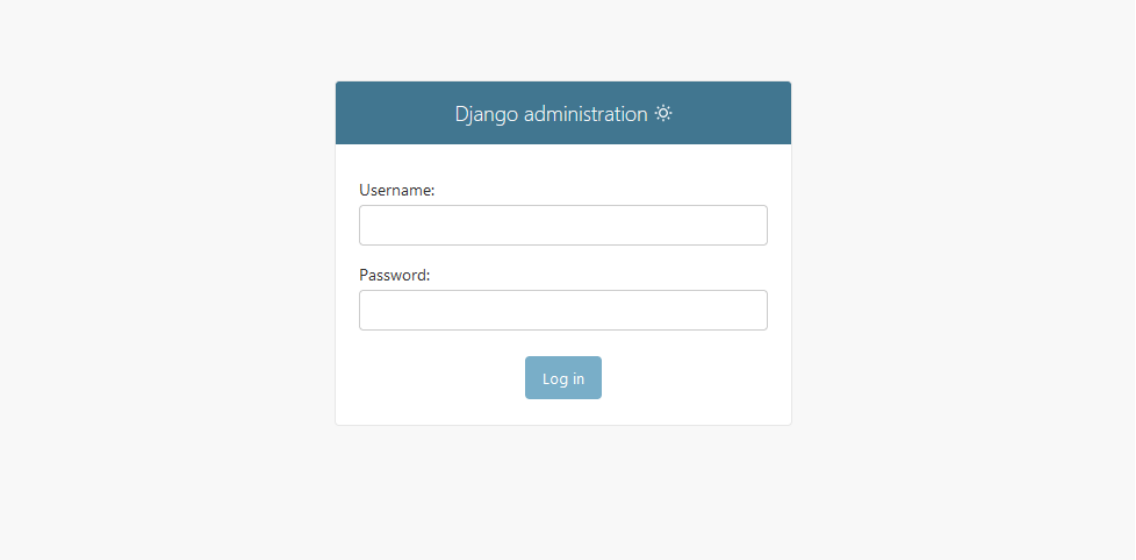
```
(venv) antonio@debian:~/crud$ python3 crud/manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 21, 2024 - 03:24:44
Django version 4.2.14, using settings 'crud.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

6 Utilizando a Aplicação

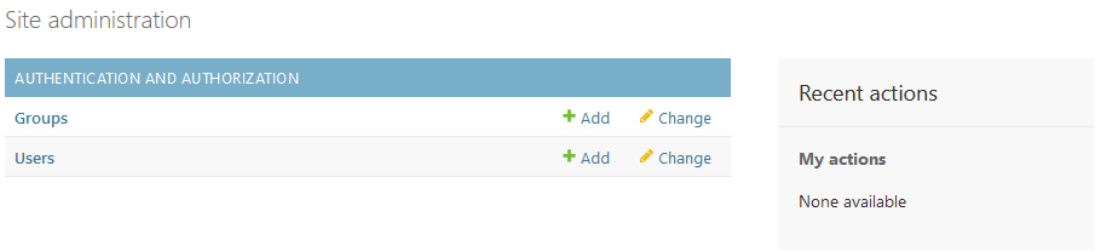
Após completar todo o processo de implementação, a aplicação deve estar disponível em "http://localhost:8000". Acessando o endereço indicado este é o resultado:



Acessando a rota "/admin" temos o seguinte resultado:



Utilizando o acesso anteriormente criado para o super usuario, é possivel efetuar o login e acessar o ambiente admnistrativo.



7 Resultados

Os resultados incluem um aplicativo web funcional que permite realizar operações de criação, leitura, atualização e exclusão de usuários.

8 Conclusão

Em conclusão, este artigo demonstrou a viabilidade e a simplicidade de implementar persistência de dados utilizando Python e Django. A abordagem passo a passo proporcionou uma compreensão clara das etapas necessárias para construir uma aplicação web básica.

A implementação de CRUD é fundamental em praticamente qualquer aplicação web, e dominar essa habilidade com Django permite a criação de soluções eficientes e escaláveis.