

Elias Santos

Aula: Autenticação

Nota: A configuração necessária para iniciar a autenticação foi feita para nós quando criamos o aplicativo. As tabelas de banco de dados para usuários e permissões de modelo foram criadas quando chamamos pela primeira vez `python manage.py migrate`.

A configuração está definida nas seções `INSTALLED_APPS` e `MIDDLEWARE` no `settings.py`

PYTHON

```
INSTALLED_APPS = [
    ...
    'django.contrib.auth', #Core authentication framework and its default models.
    'django.contrib.contenttypes', #Django content type system (allows permissions to be
associated with models).
    ....

MIDDLEWARE = [
    ...
    'django.contrib.sessions.middleware.SessionMiddleware', #Manages sessions across requests
    ...
    'django.contrib.auth.middleware.AuthenticationMiddleware', #Associates users with requests
using sessions.
    ....
```

Criando um super usuário

Para fazer login no site de administração, precisamos de uma conta de usuário com o status da equipe ativado. Para visualizar e criar registros, também precisamos que esse usuário tenha permissões para gerenciar todos os nossos objetos. Você pode criar uma conta "superusuário" que tenha acesso total ao site e todas as permissões necessárias usando `manage.py`.

Chame o seguinte comando, no mesmo diretório que `manage.py`, para criar o superusuário. Você será solicitado a digitar um nome de usuário, endereço de e-mail e senha forte.

```
>> python manage.py createsuperuser
```

Quando esse comando for concluído, um novo superusuário será adicionado ao banco de dados. Agora reinicie o servidor de desenvolvimento para que possamos testar o login:

```
>> python manage.py runserver
```

Fazendo o login e usando o site

Para fazer login no site, abra o URL `/admin` (e.g. <http://127.0.0.1:8000/admin>) e insira suas novas credenciais de usuário e senha de superusuário (você será redirecionado para a página de login e, em seguida, de volta para o URL `/admin` depois de inserir seus detalhes).

Esta parte do site exibe todos os nossos modelos, agrupados por aplicativo instalado. Você pode clicar no nome de um modelo para ir a uma tela que lista todos os seus registros associados e clicar nos registros para editá-los. Você também pode clicar diretamente no link Adicionar ao lado de cada modelo para começar a criar um registro desse tipo.

Em nosso exemplo, experimente adicionar uma nova Pessoa.

Criando grupos

1. Clique no botão Adicionar (ao lado de Grupo) para criar um novo grupo; digite o Nome "Membros do Projeto" para o grupo. Não precisamos compor permissão, então após inserir o grupo clique em SALVAR.

Administração Django

Home > Autenticação e Autorização > Grupos > Adicionar grupo

Comece a digitar para filtrar...

AUTENTICAÇÃO E AUTORIZAÇÃO

- Grupos + Adicionar
- Usuários + Adicionar

CONTATO

- Pessoas + Adicionar

Adicionar grupo

Nome:

Permissões:

Permissões disponíveis ?

- administrador | entrada de registro | Pode adicionar entrada de registro
- administrador | entrada de registro | Pode alterar a entrada de registro
- administrador | entrada de registro | Pode excluir entrada de registro
- administrador | entrada de registro | Pode visualizar a entrada de registro
- autenticação | grupo | Pode adicionar grupo
- autenticação | grupo | Pode mudar de grupo
- autenticação | grupo | Pode excluir grupo
- autenticação | grupo | Pode visualizar o grupo

Criando usuário

1. Clique no botão Adicionar ao lado de *Usuários* para abrir a caixa de diálogo *Adicionar usuário*.
2. Digite um nome de usuário e uma senha/confirmação de senha adequados para o usuário de teste.

Adicionar usuário

Primeiro, insira um nome de usuário e uma senha. Então, você poderá editar mais opções do usuário.

Nome de usuário:	<input type="text" value="elias"/>
<small>Obrigatório. 150 caracteres ou menos. Somente letras, dígitos e @/./+/-/_.</small>	
Senha:	<input type="password" value="....."/>
<small>Sua senha não pode ser muito semelhante às suas outras informações pessoais. Sua senha deve conter pelo menos 8 caracteres. Sua senha não pode ser uma senha comumente usada. Sua senha não pode ser totalmente numérica.</small>	
Confirmação Da Senha:	<input type="password" value="....."/>
<small>Digite a mesma senha de antes, para verificação.</small>	

3. Pressione SALVAR para criar o usuário. O site de administração criará o novo usuário e levará você imediatamente para uma tela Alterar usuário, na qual é possível alterar seu nome de usuário e adicionar informações aos campos opcionais do modelo de usuário. Esses campos incluem o nome, o sobrenome, o endereço de email e o status e as permissões do usuário (somente o sinalizador Ativo deve ser definido). Mais abaixo, você pode especificar os grupos e permissões do usuário e ver datas importantes relacionadas ao usuário (por exemplo, a data de ingresso e a última data de login).
4. Importante definir um endereço de email para o usuário.

✔ O usuário "elias" foi adicionado com sucesso. Você pode editá-lo novamente abaixo.

Mudar usuário

Élias

Nome de usuário:	<input type="text" value="elias"/>
<small>Obrigatório. 150 caracteres ou menos. Somente letras, dígitos e @/./+/-/_.</small>	
Senha:	<small>algoritmo:pbkdf2_sha256 iterações:600.000 salt:RyZ0cc***** hash:3j4+y8*****</small>
<small>As senhas brutas não são armazenadas, portanto não há como ver a senha desse usuário, mas você pode alterar a senha usando este formulário.</small>	

Informação pessoal

Primeiro nome:	<input type="text"/>
Sobrenome:	<input type="text"/>
Endereço de email:	<input type="text"/>

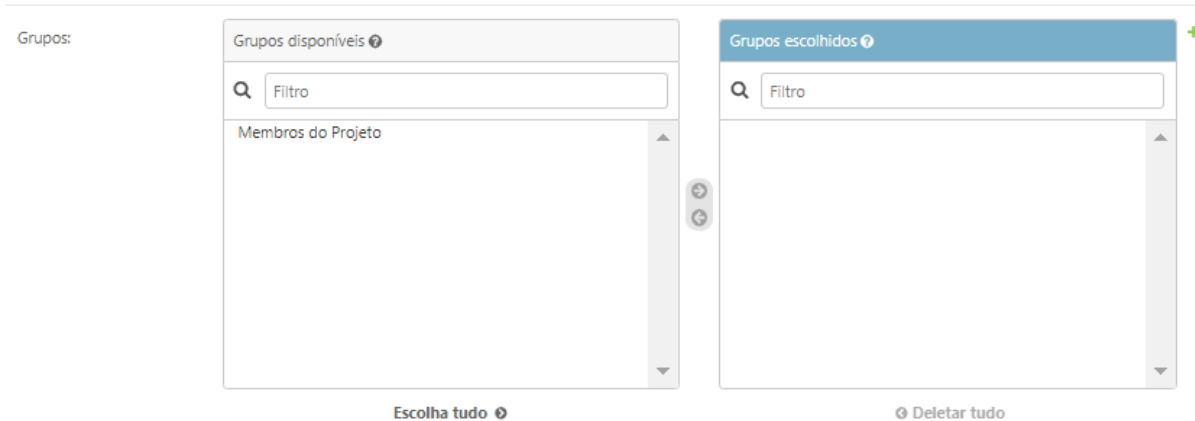
Permissões

<input checked="" type="checkbox"/> Ativo
<small>Designa se este usuário deve ser tratado como ativo. Desmarque esta opção em vez de excluir contas.</small>
<input type="checkbox"/> Status da equipe
<small>Designa se o usuário pode fazer login neste site de administração.</small>
<input type="checkbox"/> Status de superusuário
<small>Designa que este usuário tem todas as permissões sem atribuí-las explicitamente.</small>

Elias Santos

Aula: Autenticação

5. Na seção *Grupos*, selecione grupo de Membros do Projeto na lista de *Grupos disponíveis* e pressione a seta para a direita entre as caixas para movê-lo para a caixa *Grupos escolhidos*.



Agora basta SALVAR

URLs do Projeto

Adicione o seguinte à parte inferior do arquivo `urls.py` do projeto (`projeto/projeto/urls.py`):

```
#Adicione URLs de autenticação de site Django (para login, logout, gerenciamento de
senha)
urlpatterns += [
    path('accounts/', include('django.contrib.auth.urls')),
]
```

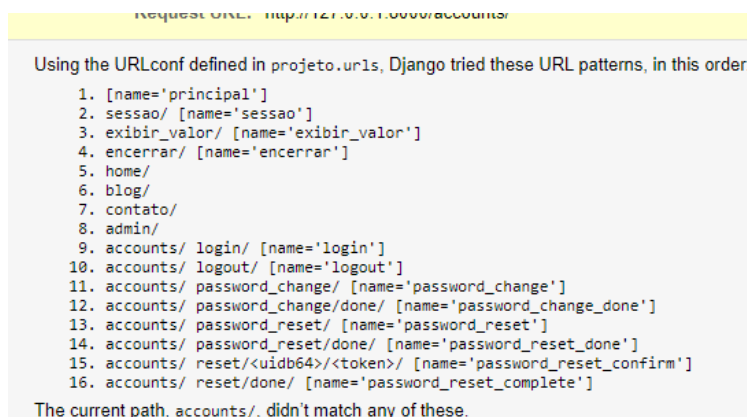
Certifique-se de ter em seu `urlpatterns` a linha:

```
path('admin/', admin.site.urls),
```

Lembrando que o modelo admin tem a seguinte linha de importação:

```
from django.contrib import admin
```

Navegue até URL `http://127.0.0.1:8000/accounts/` e o Django mostrará um erro que não foi possível encontrar esse URL e listará todos os URLs que ele tentou. A partir disso, você pode ver os URLs que funcionarão, por exemplo:



Note que já existem caminhos definidos que você não criou, mas estão no template.

Elias Santos

Aula: Autenticação

Agora tente navegar para o URL de login (<http://127.0.0.1:8000/accounts/login/>). Isso falhará novamente, mas com um erro informando que estamos perdendo o modelo necessário (`registration/login.html`) no caminho de pesquisa do modelo. Você verá as seguintes linhas listadas na seção amarela na parte superior:

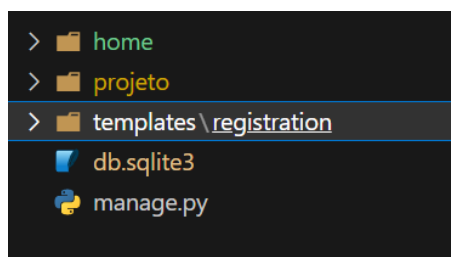
Exception Type: `TemplateDoesNotExist`

Exception Value: `registration/login.html`

A próxima etapa é criar um diretório de registro no caminho de pesquisa e adicionar o arquivo `login.html`.

Diretório de Templates

Os URLs (e implicitamente, visualizações) que acabamos de adicionar esperam encontrar seus modelos associados em um diretório `/registration/` em algum lugar no caminho de pesquisa de modelos.



Neste site, colocaremos nossas páginas HTML de tratamento da autenticação no diretório `templates/registration/`. Esse diretório deve estar no diretório raiz do projeto, ou seja, o mesmo diretório que a pasta `home` e `projeto`. Por favor, crie essas pastas agora.

Template de login.html

```
{% extends 'global/base.html' %}

{% block info %}

    {% if form.errors %}
        <p>Seu nome de usuário e senha não correspondem. Por favor, tente novamente.</p>
    {% endif %}

    {% if next %}
        {% if user.is_authenticated %}
            <p>Sua conta não tem acesso a esta página. Para prosseguir, faça login com uma conta que tenha acesso.</p>
        {% else %}
            <p>Faça login para ver esta página.</p>
        {% endif %}
    {% endif %}

{% endif %}
```

Elias Santos

Aula: Autenticação

```

<form method="post" action="{% url 'login' %}" class="form-control">
    {% csrf_token %}
    <table>
        <tr>
            <td>{{ form.username.label_tag }}</td>
            <td>{{ form.username }}</td>
        </tr>
        <tr>
            <td>{{ form.password.label_tag }}</td>
            <td>{{ form.password }}</td>
        </tr>
    </table>
    <input type="submit" value="login" />
    <input type="hidden" name="next" value="{{ next }}" />

    {% Supõe que você configure a visualização password_reset em seu URLconf %}
    <p><a href="{% url 'password_reset' %}">Esqueceu a senha?</a></p>

</form>

{% endblock info %}

```

No settings.py, vamos incluir o diretório templates que foi criado anteriormente:

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTe
        'DIRS': [
            BASE_DIR / 'base',
            os.path.join(BASE_DIR, 'templates'),
        ],
        'APP_DIRS': True,
    },
]

```

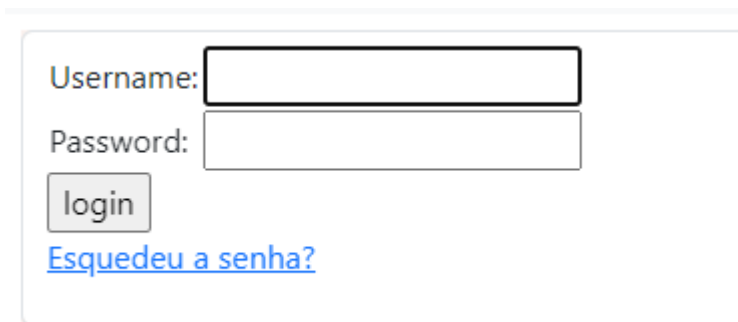
A biblioteca **os** em Python é uma biblioteca padrão que fornece um conjunto de funções para interagir com o sistema operacional subjacente, independentemente do sistema operacional em que o código Python está sendo executado. Ela é extremamente útil para tarefas que envolvem manipulação de arquivos, gerenciamento de diretórios, manipulação de caminhos de arquivo e execução de comandos do sistema.

Elias Santos

Aula: Autenticação

A finalidade da instrução **os.path.join** é criar um caminho completo (path) para um arquivo ou diretório, levando em consideração as regras do sistema operacional em que o código está sendo executado.

Agora, navegue de volta para a página de login (<http://127.0.0.1:8000/accounts/login/>). Depois de salvar seu modelo, você verá algo assim:



The image shows a web form for logging in. It has two input fields: 'Username:' and 'Password:'. Below the 'Password:' field is a button labeled 'login'. At the bottom of the form is a blue link that says 'Esqueceu a senha?'.

Se você fizer login usando credenciais válidas, será redirecionado para outra página (por padrão, isso será <http://127.0.0.1:8000/accounts/profile/>). O problema é que, por padrão, o Django espera que, ao fazer o login, você deseje ser levado para uma página de perfil, o que pode ou não ser o caso. Como você ainda não definiu esta página, receberá outro erro!

Abra as configurações do projeto (`/projeto/projeto/settings.py`) e adicione o texto abaixo na parte inferior. Agora, quando você faz login, deve ser redirecionado para a página inicial do site por padrão.

```
# Redirecionar para URL inicial após login (redirecionamento padrão para /accounts/profile/)
LOGIN_REDIRECT_URL = '/'
```

Template de logout

Se você navegar para o URL de logout (<http://127.0.0.1:8000/accounts/logout/>) você verá um comportamento estranho - seu usuário será desconectado com certeza, mas você não teve um caminho definido. Precisamos modificar isso.

→ Crie e abra `/projeto/templates/registration/logged_out.html`. Copie o texto abaixo:

```
{% extends 'global/base.html' %}

{% block info %}
    <p>Desconectado!</p>
    <a href="{% url 'login'%}">Clique aqui para fazer login novamente.</a>
{% endblock %}
```

Elias Santos

Aula: Autenticação

Resultado:

Desconectado!

[Clique aqui para fazer login novamente.](#)

Templates para reset de password

O sistema de redefinição de senha padrão usa o email para enviar ao usuário um link de redefinição. Você precisa criar formulários para obter o endereço de email do usuário, enviar o email, permitir que ele insira uma nova senha e anotar quando todo o processo está completo.

Os seguintes modelos podem ser usados como ponto de partida.

Formulário para reset de password

Este é o formulário usado para obter o endereço de email do usuário (para enviar o email de redefinição de senha). Crie /projeto/templates/registration/password_reset_form.html e forneça o seguinte conteúdo:

```
{% extends 'global/base.html' %}

{% block info %}
    <form action="" method="post">
        {% csrf_token %}
        {% if form.email.errors %}
            {{ form.email.errors }}
        {% endif %}
        <p>{{ form.email }}</p>
        <input type="submit" class="btn btn-default btn-lg" value="Reset password">
    </form>
{% endblock %}
```


Elias Santos

Aula: Autenticação

Password reset done

Este formulário é exibido após a coleta do seu endereço de email. Crie /projeto/templates/registration/password_reset_done.html, e forneça o seguinte conteúdo:

```
{% extends 'global/base.html' %}

{% block info %}
    <p>Enviamos a você por e-mail instruções para definir sua senha. Se eles não
    chegarem em alguns minutos, verifique sua pasta de spam.</p>
{% endblock %}
```

Password reset email

Este modelo fornece o texto do email em HTML que contém o link de redefinição que enviaremos aos usuários. Crie /projeto/templates/registration/password_reset_email.html e forneça o seguinte conteúdo:

```
Alguém pediu redefinição de senha para e-mail {{ email }}. Siga o link abaixo:
{{ protocol }}://{{ domain }}{% url 'password_reset_confirm' uidb64=uid token=token %}
```

Password reset confirm

É nesta página que você digita sua nova senha depois de clicar no link no e-mail de redefinição de senha. Crie /projeto/templates/registration/password_reset_confirm.html e forneça o seguinte conteúdo:

```
{% extends 'global/base.html' %}

{% block info %}
    {% if validlink %}
        <p>Por favor insira (e confirme) sua nova senha.</p>
        <form action="" method="post">
            {% csrf_token %}
            <table>
                <tr>
                    <td>
                        {{ form.new_password1.errors }}
                        <label for="id_new_password1">Nova Senha:</label>
                    </td>
                    <td>{{ form.new_password1 }}</td>
                </tr>
            </table>
        </form>
    {% else %}
        <p>Link inválido.</p>
    {% endif %}
{% endblock %}
```

Elias Santos

Aula: Autenticação

```

        <tr>
            <td>
                {{ form.new_password2.errors }}
                <label for="id_new_password2">Confirme sua senha:</label>
            </td>
            <td>{{ form.new_password2 }}</td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Mudar minha senha" /></td>
        </tr>
    </table>
</form>
{% else %}
    <h1>Falha na redefinição de senha</h1>
    <p>
        O link de redefinição de senha era inválido, possivelmente porque já foi
        usado. Solicite uma nova redefinição de senha.
    </p>
{% endif %}
{% endblock %}

```

Password reset complete

Este é o último modelo de redefinição de senha, exibido para notificá-lo quando a redefinição de senha for bem-sucedida. Crie /projeto/templates/registration/password_reset_complete.html e forneça o seguinte conteúdo:

```

{% extends 'global/base.html' %}

{% block info %}
    <h1>A senha foi alterada!</h1>
    <p><a href="{% url 'login' %}">fazer login novamente?</a></p>
{% endblock %}

```

templates\registration		
dj	logged_out.html	U
dj	login.html	U
dj	password_reset_complete.html	U
dj	password_reset_confirm.html	U
dj	password_reset_done.html	U
dj	password_reset_email.html	U
dj	password_reset_form.html	U

Esta é a lista de arquivos que foram criados em registration para as definições de login/logout

Elias Santos

Aula: Autenticação

Testando as novas páginas de autenticação

Agora que você adicionou a configuração da URL e criou todos esses modelos, as páginas de autenticação agora devem funcionar!

Você pode testar as novas páginas de autenticação tentando fazer login e sair da sua conta de superusuário usando estes URLs:

- <http://127.0.0.1:8000/accounts/login/>
- <http://127.0.0.1:8000/accounts/logout/>

Você poderá testar a funcionalidade de redefinição de senha no link na página de login. Esteja ciente de que o Django enviará apenas emails de redefinição para endereços (usuários) que já estão armazenados em seu banco de dados!

Nota: O sistema de redefinição de senha exige que seu site suporte e-mail, que está além do escopo deste artigo, portanto esta parte ainda não funcionará. Para permitir o teste, coloque a seguinte linha no final do seu arquivo settings.py. Isso registra todos os emails enviados ao console (para que você possa copiar o link de redefinição de senha do console).

```
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

Para mais informações, veja [Sending email](#) (Django docs).

Teste agora o seu projeto, você verá a saída de solicitação de nova senha no seu prompt, é só copiar o endereço e colar no seu navegador.

Testando usuários autenticados

Esta seção analisa o que podemos fazer para controlar seletivamente o conteúdo que o usuário vê, com base em se está logado ou não.

O Django já tem tudo preparado no template, e com isso você pode obter informações sobre o usuário conectado no momento em modelos com a variável de template `{{ user }}` (isso é adicionado ao contexto do template por padrão quando você configura o projeto como fizemos em nosso projeto).

Elias Santos

Aula: Autenticação

Normalmente você primeiro testará contra a variável de template `{{ user.is_authenticated }}` para determinar se o usuário está qualificado para ver conteúdo específico. Para demonstrar isso, em seguida, atualizaremos nosso menu com um link "Login" se o usuário estiver desconectado e um link "Logout" se estiverem conectados.

Abra o template base (/projeto/base/global/partials/menu.html) e altere o seu menu, adicionando após contato, o tag de verificação de usuário, conforme abaixo:

```
{% if user.is_authenticated %}
    <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="{% url
'logout'%}?next={{request.path}}">Usuário: {{ user.get_username }} | Logout</a>
    </li>
{% else %}
    <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="{% url
'login'%}?next={{request.path}}">Login</a>
    </li>
{% endif %}
```

⇒ Execute seu projeto e aqui você já tem a validação do login

Testando nas views

Se você estiver usando views baseadas em funções, a maneira mais fácil de restringir o acesso a suas funções é aplicando o decorator `login_required` à sua função view, como mostrado abaixo. Se o usuário estiver logado, seu código de exibição será executado normalmente. Se o usuário não estiver conectado, isso será redirecionado para o URL de login definido nas configurações do projeto.(`settings.LOGIN_URL`), passando o caminho absoluto atual como o next no parâmetro da URL. Se o usuário conseguir fazer login, ele retornará a esta página, mas desta vez autenticado.

PYTHON

```
from django.contrib.auth.decorators import login_required

@login_required
def my_view(request):
    ...
```

Para detalhes adicionais, consulte o [Django docs here](#).

from django.contrib.auth.decorators import login_required

@login_required

Elias Santos

Aula: Autenticação

Desafio: Crie um formulário para permitir a criação de um novo usuário

Os campos do template e a forma de passar os dados para a criação de um novo usuário, estão na imagem a seguir:

PYTHON

```
from django.contrib.auth.models import User

# Create user and save to the database
user = User.objects.create_user('myusername', 'myemail@crazymail.com', 'mypassword')

# Update fields and then save again
user.first_name = 'John'
user.last_name = 'Citizen'
user.save()
```

O construtor passa para o objeto User, o nome do usuário, o email e a senha. E os outros dois campos, são referidos como first_name e last_name.

Dica:

- Você precisa de um local para inserir o formulário (sugestão: app home)
- Criar a view que irá inserir o novo registro e a url
- Crie primeiro sem se preocupar com as críticas aos campos e posteriormente com o funcionamento, faça as devidas críticas nos campos.