

CONECTANDO AO FACEBOOK COM DJANGO

Raíssa Azevedo

[Raíssa Azevedo](#)

```
pip3 install django social-auth-app-django django-bootstrap4
```

Após instalar o Django e fazer as configurações básicas...

No settings.py indicar as aplicações básicas instaladas...

```
'core',  
'bootstrap4',  
'social_django',
```

As autenticações de facebook são feitas dessa forma.

```
AUTHENTICATION_BACKENDS = [  
    'social_core.backends.facebook.FacebookOAuth2',  
    # Apenas se quiser manter a autenticação padrão do django  
    'django.contrib.auth.backends.ModelBackend',  
]
```

Todas as autenticações de social medias seguem a mesma premissa.

Os tipos de autenticações suportadas por python e django podem ser encontrados em:

[CLIQUE AQUI](#)

Ou pesquise no google **Python Social Auth** em seguida de backends.

Em templates(settings.py), adicionar mais 2 context processors:

```
'social_django.context_processors.backends',  
'social_django.context_processors.login_redirect',
```

Depois, definir as rotas pra quando houver Login e o Logout da página.

```
LOGIN_URL = 'login'  
LOGIN_REDIRECT_URL = 'index'  
LOGOUT_URL = 'logout'  
LOGOUT_REDIRECT_URL = 'login'
```

Em seguida, elaborar as rotas nas URLS. Na url do projeto, a configuração é padrão.

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path("", include('core.urls')),
    path('admin/', admin.site.urls),
]
```

Entretanto na rota do aplicativo, a configuração muda um pouco.

```
from django.urls import path, include
from django.contrib.auth import views as auth_views

from core.views import IndexView, LoginView

urlpatterns = [
    path('login/', LoginView.as_view(), name='login'),
    path('logout/', auth_views.LogoutView.as_view(), name='logout'),
    path('social-auth/', include('social_django.urls', namespace='social')),
    path("", IndexView.as_view(), name='index'),
]
```

Isso ocorre devido as configurações de backend para a conexão de login com redes sociais não serem padrões do django, entretanto, o framework possui algumas configurações facilitadoras.

Na views.py:

```
from django.views.generic import TemplateView
from django.contrib.auth.mixins import LoginRequiredMixin

class IndexView(LoginRequiredMixin, TemplateView):
    template_name = 'index.html'

class LoginView(TemplateView):
    template_name = 'login.html'
```

Os Templates:

A base.html:

```
{% load static %}
{% load bootstrap4 %}
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device.width, initial-scale=1.0"/>
  <meta http-equiv="X-UA-Compatible" content="ie=edge"/>
  {% bootstrap_css %}
  <link rel="stylesheet" href="{% static 'css/style.css' %}" />
  <title>Django Social</title>
</head>
<body>
  <div class="container-fluid">
    <div>
      <h1 class="text-white text-center">{% block title %}{% endblock %}</h1>
      <div class="card p-5">
        {% block content %}{% endblock %}
      </div>
    </div>
  </div>
{% bootstrap_javascript jquery='full' %}
</body>
</html>
```

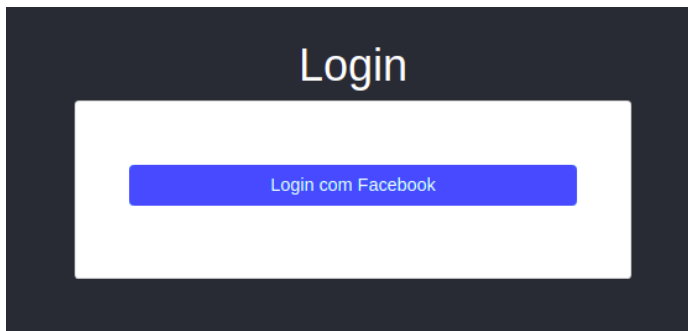
O index:

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Inicio{% endblock %}
{% block content %}
  <div class="row">
    <div class="col-sm-12 mb-3">
      <h4 class="text-center">Bem vindo(a) {{ user.username }}</h4>
    </div>
  </div>
{% endblock %}
```

O login.html:

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Login{% endblock %}
{% block content %}
  <div class="row">
    <div class="col-md-8 mx-auto social-container my-2 order-md-1">
      <button class="btn btn-primary mb-2">
        <a href="#">Login com Facebook</a>
      </button>
    </div>
  </div>
{% endblock %}
```

Isso irá gerar uma página de login padrão:



Em seguida, acessar as configurações de developers do Facebook, para ter acesso as chaves.

E criar o aplicativo para acesso pelo instagram.

Nas configurações básicas do aplicativo do facebook, denominar os dominios (localhost, ou onde o site tiver hospedado).

Depois voltar o settings da IDE e fazer as seguintes configurações:

```
# Configurações para qualquer rede social
SOCIAL_AUTH_RAISE_EXCEPTIONS = False
```

Se a pessoa já tiver feito acesso pelo facebook, deslogado, e em outro dia logar novamente, a conta já está salva, e isso impede a criação de uma exceção.

```
# Configurações para Facebook
SOCIAL_AUTH_FACEBOOK_KEY = "
SOCIAL_AUTH_FACEBOOK_SECRET = "
SOCIAL_AUTH_FACEBOOK_SCOPE = ['email', 'user', 'link']
SOCIAL_AUTH_FACEBOOK_PROFILE_EXTRA_PARAMS = {
    'fields': 'id, name, email, picture.type(large), link'
}
SOCIAL_AUTH_FACEBOOK_EXTRA_DATA = [
    ('name', 'name'),
    ('email', 'email'),
    ('picture', 'picture'),
    ('link', 'profile_url'),
]
```

Qualquer seja a configuração para a criação de login através da rede social. Irá gerar essas duas chaves (Secret - Key).

Em seguida voltar ao facebook developers.

Copiar o ID do aplicativo colar em **Key** e copiar a chave secreta do aplicativo e colar em **secret**.

Em seguida, voltar ao login html e corrigir o endereço href.

```
<a href="{% url 'social: begin' 'facebook' %}">Login com Facebook</a>
```

A configuração é a mesma para qualquer rede social.

Fazer uma ultima modificação no index.

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Inicio{% endblock %}
{% block content %}
<div class="row">
  <div class="col-sm-12 mb-3">
    <h4 class="text-center">Bem vindo(a) {{ user.username }}</h4>
  </div>
  {% for a in backends.associated %}
    {% if a.provider == 'facebook' %}
      <div class="col-md-4 text-center">
        
      </div>
      <div class="col-md-8 social-container my-2">
        <p>Logado via: {{ a.provider|title }}</p>
        <p>Nome: {{ a.extra_data.name }}</p>
        <p>Profile:<a href="{{ a.extra_data.profile_url }}">Link</a> </p>
      </div>
    {% endif %}
  {% endfor %}
  <div class="col-sm-12 mt-2 text-center">
    <button class="btn btn-warning">
      <a href="{% url 'logout' %}">Logout</a>
    </button>
  </div>
</div>
{% endblock %}
```