



GERANDO ARQUIVO PDF COM DJANGO

Raíssa Azevedo

[Raíssa Azevedo](#)

Como gerar PDF na aplicação Django?

Existem várias formas de gerar documentos PDF com Django ou Python;

As bibliotecas necessárias para esse projeto são:

```
pip3 install django
django-admin startproject relatorio .
django-admin startapp core
```

Fazer as configurações necessárias no settings.py

Usar a biblioteca **Reportlab** recomendada nas documentações do Django:

```
pip3 install reportlab
python3 manage.py shell
import reportlab
```

Executar os seguintes passos para verificar se não houve nenhum erro na instalação da biblioteca.

Criar um arquivo de rotas na aplicação core:

```
from django.urls import path
from .views import IndexView

urlpatterns = [
    path("", IndexView.as_view(), name='index'),
]
```

Avisar a rotas do projeto:

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

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

Criando os comandos para a geração do PDF:

Na views.py:

```
import io
from django.http import FileResponse
from django.views.generic import View

from reportlab.pdfgen import canvas

class IndexView(View):

    def get(self, request, *args, **kwargs):
        # Cria um arquivo para receber os dados e gerar o PDF
        buffer = io.BytesIO()

        # Cria o arquivo PDF
        pdf = canvas.Canvas(buffer)

        # Insere "coisas" no PDF
        pdf.drawString(100, 100, "Gerando PDF com Django")

        # Ao finalizar a inserção de coisas
        pdf.showPage()
        pdf.save()

        # Por fim, retornar o buffer para o início do arquivo
        buffer.seek(0)

        return FileResponse(buffer, as_attachment=True, filename='Relatorio1.pdf')

        # Abre o PDF direto no navegador
        return FileResponse(buffer, filename='Relatorio1.pdf')
```

Isso irá gerar um página em PDF.

O reportlab possui duas versões, a opensource com algumas limitações, e a versão paga.

Há outra forma de gerar PDF com Django:

Através da biblioteca **WeasyPrint** que é opensource e desenvolvido para usar com Python.

A partir de uma página HTML essa biblioteca printa a página e gera o arquivo PDF.

```
pip3 install WeasyPrint
```

No template de relatório.html:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Report - Django</title>
  <style>
    body {
      background: #0080ff;
    }
  </style>
</head>
<body>
  <h1>Report - Django</h1>
  {% for p in texto %}
    <p>{{ p }}</p>
  {% endfor %}
</body>
</html>
```

Na views.py:

```
# De uso para o WeasyPrint
from django.core.files.storage import FileSystemStorage
from django.template.loader import render_to_string
from django.http import HttpResponse
from weasyprint import HTML

class Index2View(View):
    def get(self, request, *args, **kwargs):
        texto = ['Geek University', 'Evolua seu lado Geek', 'Programação Web com Python e Django']

        html_string = render_to_string('relatorio.html', {'texto': texto})

        html = HTML(string=html_string)
        html.write_pdf(target='/tmp/relatorio2.pdf')

        fs = FileSystemStorage('/tmp')

        with fs.open('relatorio2.pdf') as pdf:
            response = HttpResponse(pdf, content_type='application/pdf')
            # Faz o download do arquivo PDF
            #response['Content-Disposition'] = 'attachment: filename="relatorio2.pdf"'
            # Abre o PDF como página HTML:
            response['Content-Disposition'] = 'inline: filename="relatorio2.pdf"'
        return response
```

Na urls da aplicação:

```
from django.urls import path
from .views import IndexView, Index2View

urlpatterns = [
    path("", IndexView.as_view(), name='index'),
    path('2/', Index2View.as_view(), name='index2'),
]
```

Essas são as duas melhores formas de gerar arquivo PDF com Django.