

FORMAÇÃO COMPLEMENTAR

Mentor: Felipe Moreira de Assunção

Conteúdo para formação complementar de Desenvolvimento Web

INFORMAÇÕES DO MÓDULO

Descrição

Conheça os conceitos básicos sobre desenvolvimento Web com Python e Django

Objetivos do Ensino

- Como desenvolver para web com Python
- Criar um ambiente de desenvolvimento
- Adaptar o backend do site
- Trabalhar com módulos e aplicações

Demonstração das Ferramentas

- Utilização do Django para Desenvolvimento Web
- Utilização do Atom para edição de código

PRIMEIROS PASSOS COM O DJANGO

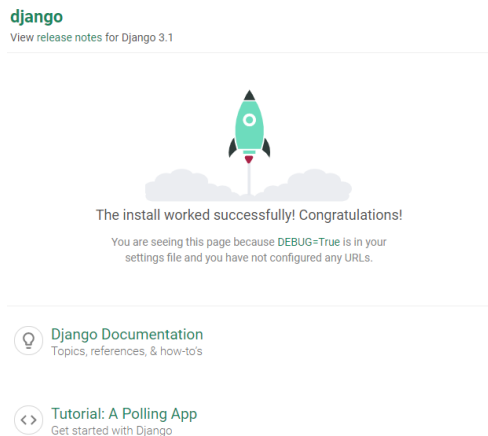
Obs: Caso necessário verifique a declaração das variáveis de ambiente / sistema para o DJANGO

Preparando o ambiente:

1. Criar a pasta que você irá armazenar o seu projeto
2. Abra o CMD e crie um ambiente virtual com o nome de sua preferência ou django-project
3. Continue digitando as seguintes instruções no CMD e avaliando cada uma das instruções executadas
 - Vá até o diretório do seu ambiente virtual recém criado e ative-o :
>>> .\Scripts\activate
 - Instalando o Django a partir do CMD:
>>> pip install django
 - Fazendo upgrade do PIP:
>>> python -m pip install --upgrade pip
 - Outra maneira de fazer upgrade do PIP:
>>> python.exe -m pip install --upgrade pip
 - Verificando a versão do Django instalada:
>>> django-admin --version
 - Iniciando um projeto no Django chamado helloworld:
>>> django-admin startproject helloworld
 - Lista arquivos do diretório:
>>> dir
 - Entra na pasta helloworld:
>>> cd helloworld

- Lista arquivos do diretório:
`>>> dir`
- Inicia o servidor de desenvolvimento no IP interno na porta 8000:
`>>> python manage.py runserver`
- Para checar o funcionamento do Django, abrir o navegador e digitar `http://localhost:8000`

Pronto! O Django está configurado e irá aparecer uma tela como essa:



INVESTIGANDO OS ARQUIVOS DO SEU PROJETO DJANGO

Após a criação do seu projeto Django, teremos a seguinte estrutura de pastas dentro do diretório:

```
mysite/
  manage.py
  mysite/
    __init__.py
    settings.py
    urls.py
    asgi.py
    wsgi.py
```

Esses arquivos são:

- O diretório **mysite/** exterior é um contêiner para o seu projeto. Seu nome não importa para o Django; você pode renomeá-lo para qualquer nome que você quiser.
- **manage.py**: um utilitário de linha de comando que permite a você interagir com esse projeto Django de várias maneiras.
- O diretório **mysite/** interior é o pacote Python para o seu projeto. Seu nome é o nome do pacote Python que você vai precisar usar para importar coisas do seu interior (por exemplo, **mysite.urls**).
- **mysite/__init__.py**: um arquivo vazio que diz ao Python que este diretório deve ser considerado um pacote Python.
- **mysite/settings.py**: configurações para o projeto Django.

- **mysite/urls.py**: as declarações de URLs para este projeto Django; um “índice” de seu site movido a Django.
- **mysite/asgi.py**: um ponto de integração para servidores web compatíveis com ASGI usado para servir seu projeto.
- **mysite/wsgi.py**: um ponto de integração para servidores web compatíveis com WSGI usado para servir seu projeto.

Se você criou o projeto e deseja iniciar novamente o servidor, basta você abrir o CMD e ir até a pasta do seu projeto, onde existe o arquivo manage.py, por exemplo, no meu caso, digitei:

```
>>> cd C:\Users\felip\Desktop\Projetos\django\project\helloworld
```

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

Depois, basta abrir o navegador e checar o funcionamento digitando `http://localhost:8000`

INTEGRANDO O BANCO DE DADOS SQLITE3

Digite no CMD, no mesmo diretório do seu projeto o comando

```
>>> python manage.py migrate
```

Se tudo der certo, irá aparecer um log, conforme imagem abaixo:

```

Prompt de Comando
Microsoft Windows [versão 10.0.19041.804]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\felip>cd C:\Users\felip\Desktop\Projeto\website\helloworld

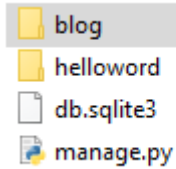
C:\Users\felip\Desktop\Projeto\website\helloworld>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying 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

C:\Users\felip\Desktop\Projeto\website\helloworld>
```

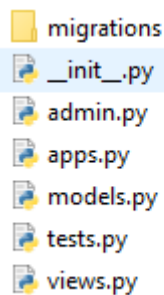
CRIANDO UMA APP DENTRO DO DIRETÓRIO DO NOSSO SITE

```
>>> python manage.py startapp blog
```

Irá aparecer uma nova pasta com o nome blog, conforme imagem abaixo:

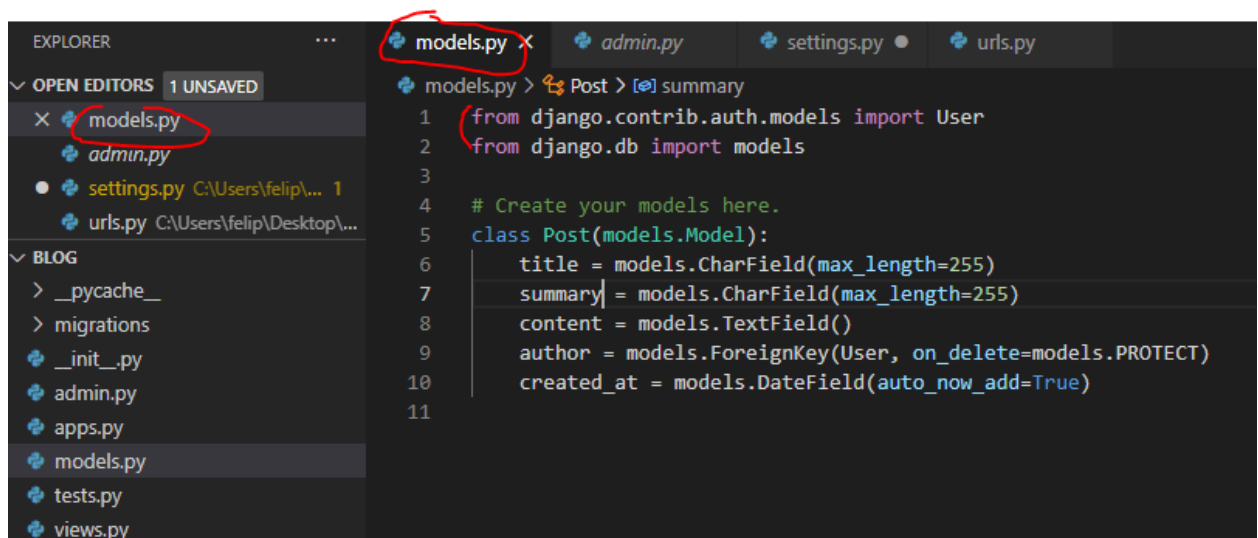


Contendo os seguintes arquivos, conforme imagem abaixo:



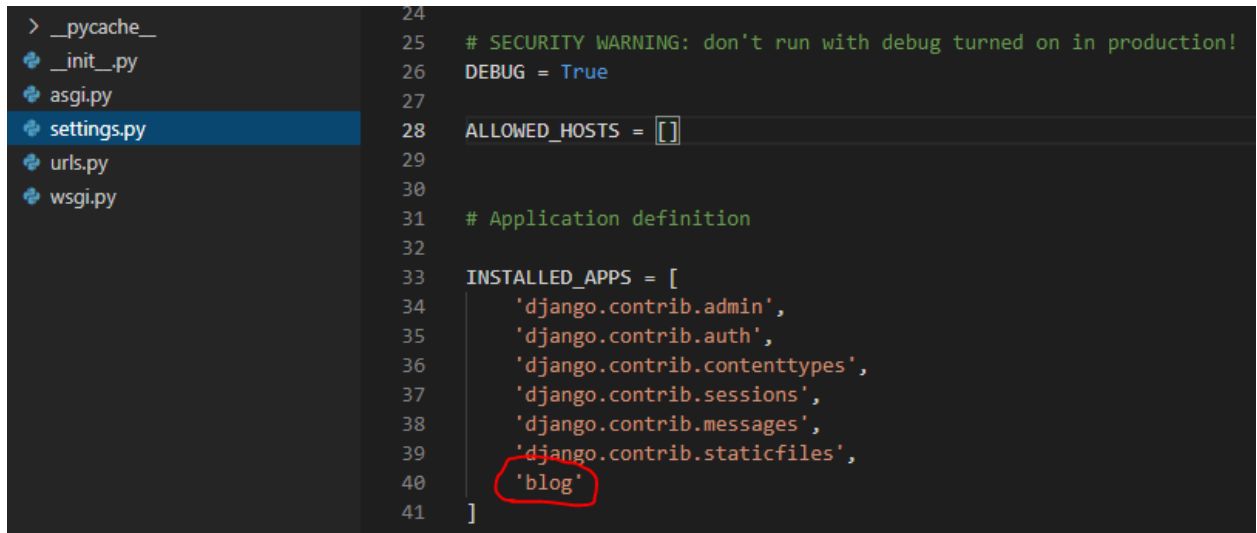
EDITANDO O ARQUIVO models.py

- Dentro do diretório 'blog' recém criado, iremos editar o arquivo models.py, conforme imagem abaixo.
- A ideia é que dentro do nosso app blog, tenhamos um módulo com o nome Post.
- O módulo Post contará com alguns campos específicos que irão lhe permitir personalizar o seu módulo. Além disso, cada um dos campos pode contar ainda com possibilidades de restrição, como a quantidade de número de caracteres, dentre outros.



FAZENDO O LINK DO PROJETO BLOG

Retorne ao diretório do seu site e edite o arquivo settings, inserindo 'blog' para que o modelo alterado anteriormente e toda a pasta seja reconhecido pela nossa aplicação, conforme imagem abaixo:



```
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'blog'
41 ]
```

Não esqueça de salvar todas as alterações!

Digite no CMD, no mesmo diretório do seu projeto o comando:

```
>>> python manage.py makemigrations blog
```

O que são Migrations (migrações)?

As migrações são a maneira do Django de propagar as mudanças feitas em seus modelos (adicionar um campo, deletar um modelo, etc.) em seu esquema de banco de dados. Eles foram projetados para serem quase sempre automáticos, mas você precisará saber quando fazer migrações, quando executá-las e os problemas comuns que pode enfrentar.

Você irá criar um super usuário para acessar o backend do Django, onde você poderá ver a tela de administração do sistema.

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

Crie o nome do usuário

Crie o password com letras e números

EDITE O ARQUIVO admin.py

Você irá editar o arquivo admin para que você possa visualizar no painel admin (backend), a aplicação blog e o módulo post recém criado

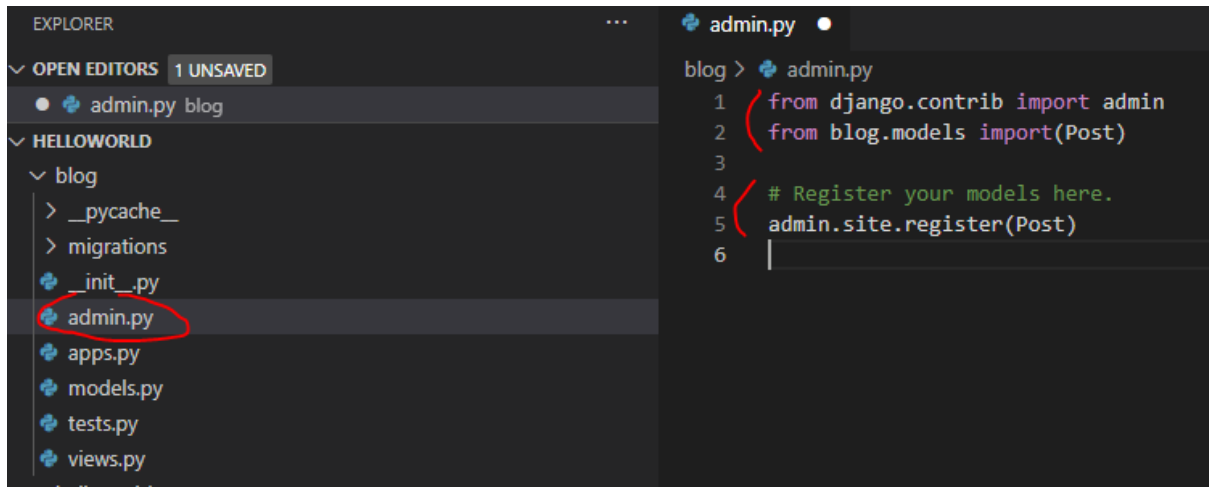
Faça a edição do arquivo admin.py, na pasta blog:

```
>>> from blog.models import(Post)
```

```
>>> admin.site.register(Post)
```

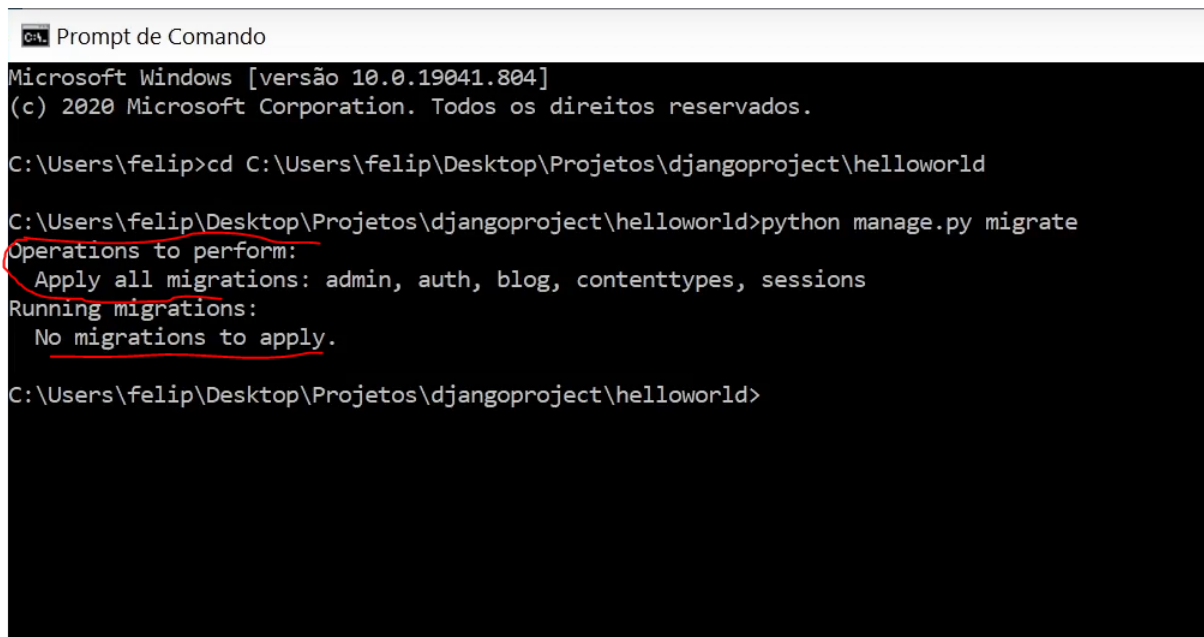
A partir desse comando, a sessão de Post irá aparecer no backend do site.

Verifique o conteúdo do código, conforme imagem abaixo:



Retorne ao terminal e digite:

```
>>> python manage.py migrate
```



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

```
cmd Prompt de Comando - python manage.py runserver
Microsoft Windows [versão 10.0.19041.804]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\felip>CD C:\Users\felip\Desktop\Projetos\django\project\helloworld

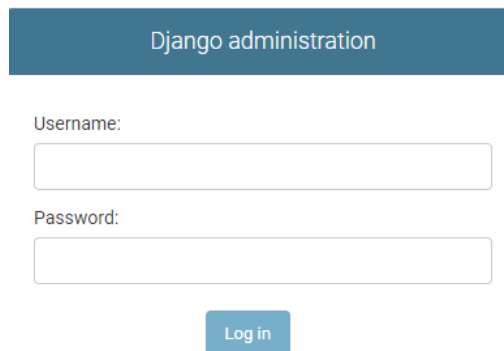
C:\Users\felip\Desktop\Projetos\django\project\helloworld>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 26, 2021 - 10:46:53
Django version 3.1.6, using settings 'helloworld.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

ACESSANDO A AREA ADMINISTRATIVA

Abra o navegador e digite o seguinte endereço: <http://localhost:8000/admin/login/>

Irá aparecer a tela de login, conforme imagem abaixo:



The image shows the Django administration login page. It features a blue header with the text 'Django administration'. Below the header, there are two input fields: 'Username:' and 'Password:'. At the bottom, there is a blue 'Log in' button.

Faça o login conforme criamos na etapa de criação do usuário para acessar a área administrativa e clique em Add post:

Django administration

WELCOME, FELIPE. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » Blog » Posts » Add post

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

BLOG

Posts [+ Add](#)

Add post

Title:

Resumo:

Content:

Author: [+](#) [x](#)

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Pronto! Você criou um módulo Post com os campos que você especificou e agora já pode editá-lo no backend do seu site.

LINKS INTERESSANTES

- Esta apostila vai te ajudar bastante no seu aprendizado com Django e te ajudar a desenvolver o seu Projeto com maior facilidade.

http://www.ezautomation.com.br/media/attachments/produtos_produto/490/apostila_django.pdf

- [Django Brasil](#), site da comunidade brasileira de Django, possui cookbook, tutoriais e documentação em português.
- [The Django Book](#) é um livro gratuito e super detalhado sobre Django feito por desenvolvedores do framework

APLICAÇÕES E ROTINAS PARA DJANGO

- [Django Basic Apps](#) é um repositório no github com várias aplicações simples como sistema para redes sociais e favoritos
- [Django Snippets](#) possui várias rotinas enviadas pelos usuários
- [Pinax Project](#) integra várias apps reusáveis como blog, verificação de email, clone do twitter, sistema de tags, importação de contatos do Gmail e outros

BIBLIOGRAFIA

- Desenvolvimento Web com Django 3 Cookbook: Soluções práticas para problemas comuns no desenvolvimento web com Python
- Aprenda Django 3 com Exemplos: Crie aplicações web profissionais em Python, começando do zero
- Python e Django: Desenvolvimento web Moderno e ágil
- Documentação Django – Disponível em: <https://docs.djangoproject.com/pt-br/3.1/>