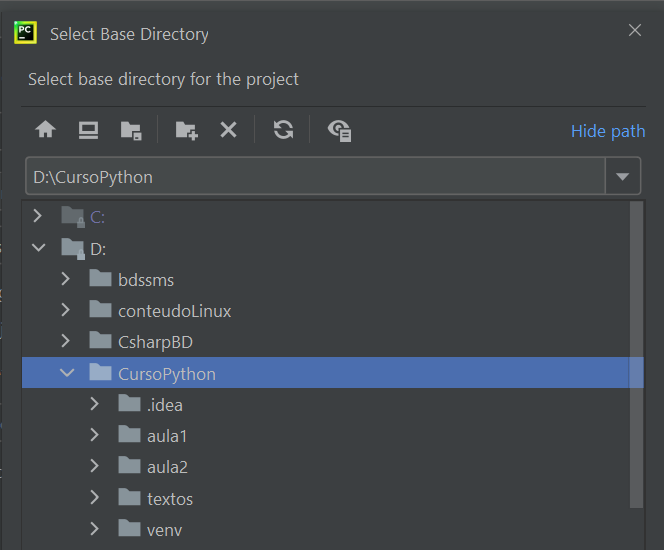
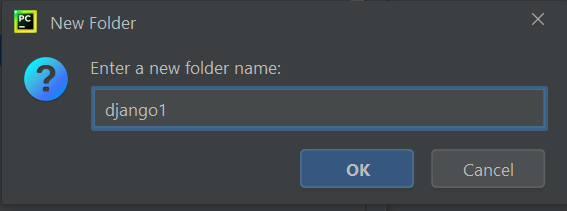
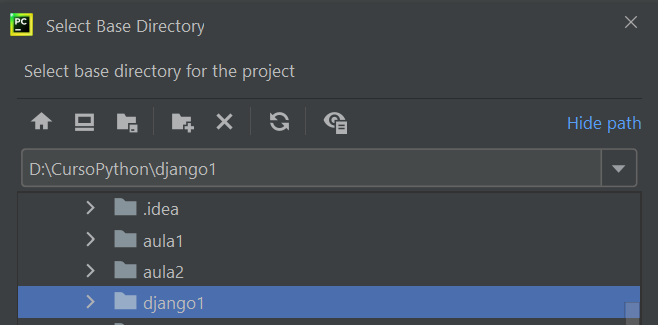
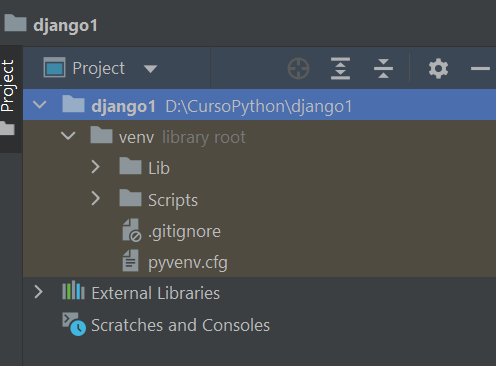
Django

Criar projeto





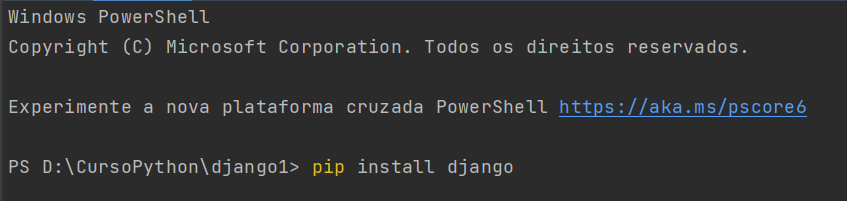




Observe que a pasta venv (Virtual enviroment) ...

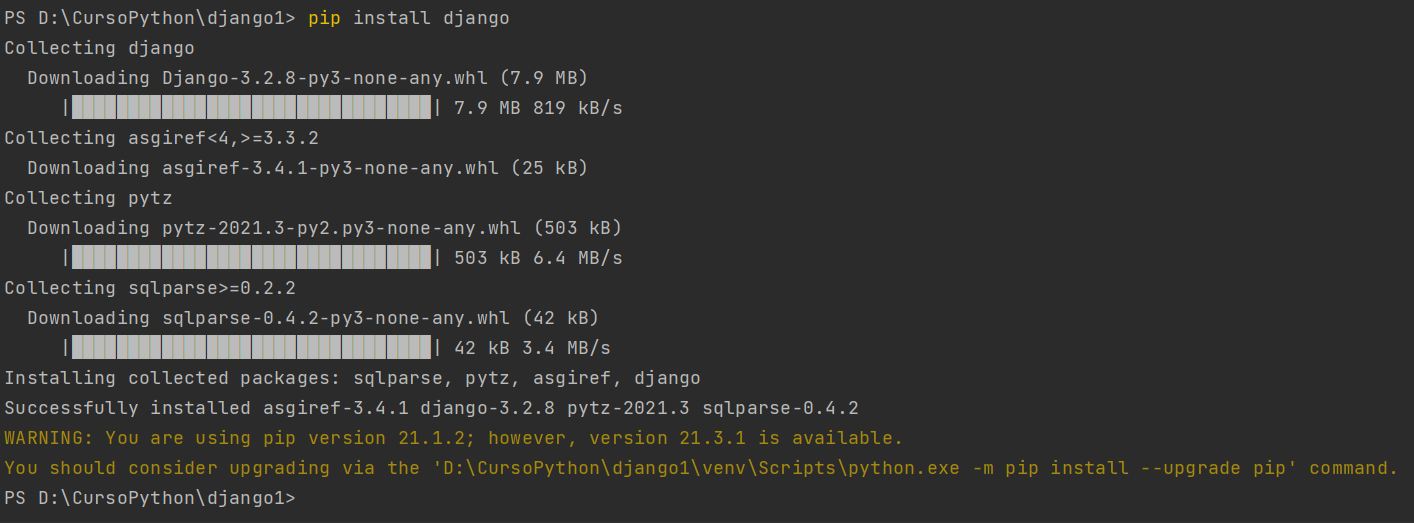
Após iniciar o projeto **devemos instalar o Django** neste projeto criado... Ele deve ser instalado como qualquer outro package dentro do projeto python

Abra o Terminal: ALT + F12



Caso você precise instalar uma versão especifica do Django utilize o comando

pip install Django==2.2.3 (por exemplo para instalar a versão 2.2.3)



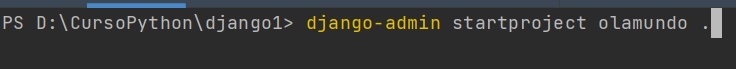
Após a instalação retornamos para o prompt de comando normal

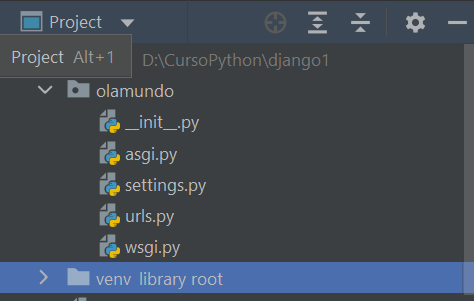
Digite: **Exit** (fechar terminal) e reabra-o novamente ALT + F12

Criar o Admin do Django

Django-admin startproject olamundo .

(*o . é para evitar que ele cria uma pasta chamada olamundo e dentro dela outra pasta com os arquivos django*)



 pasta criada após o comando executado

Nesta pasta é onde criaremos todos os apps e el devem ser registrado no arquivo *settings.py* (veremos depois como fazer isto)

\_\_init\_\_ - classe inicializadora

Setings – configurações do projeto em Djano

Urls – Definição de rotas

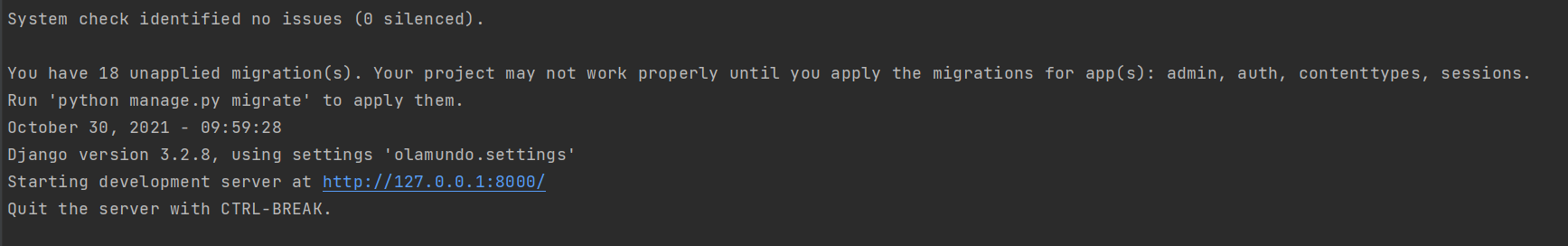
Wsgi – arquivo para upar o projeto na web (web service gateway interface)

Apps? São partes, sessões do nosso site (cada parte home, produtos, sobre etc.. )

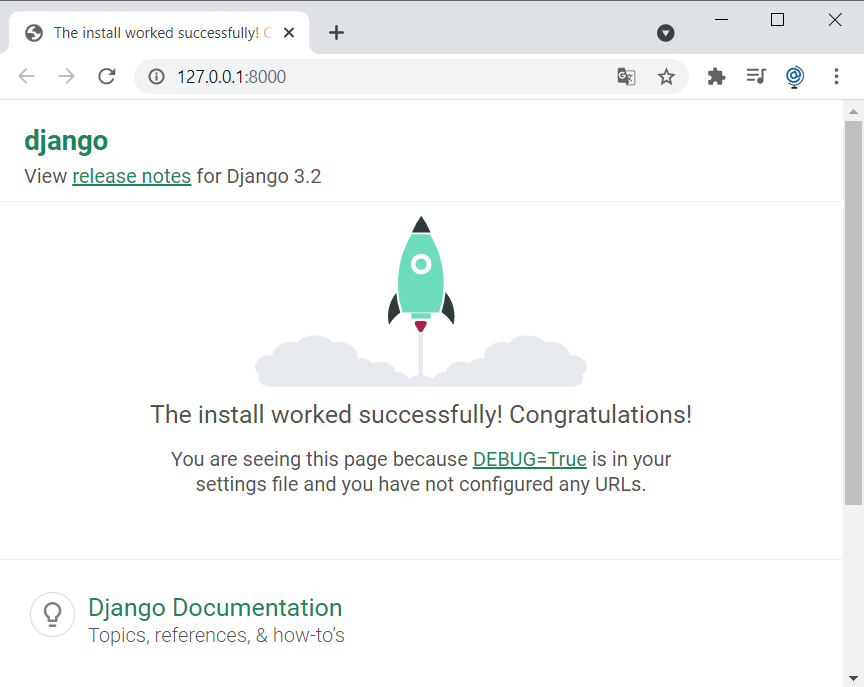
É isto que precisaser registrar em setting.py e urls são as rotas

Startar servidor





Um aviso sobre 18 migrations não aplicados... e ele indica a url do projeto



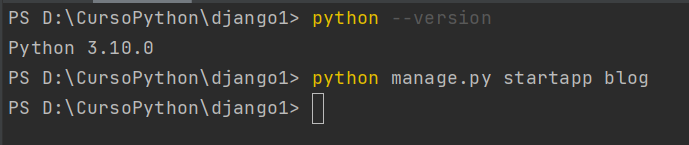
CTRL + C – encerra servidor

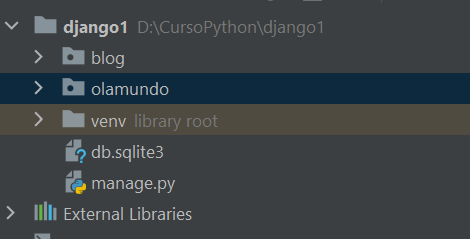
Pode mudar a porta caso ela esteja ocupaa...



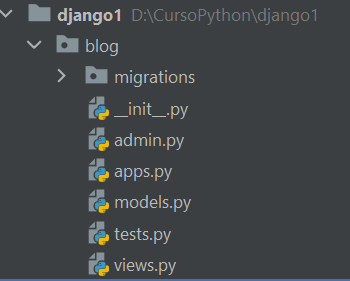
Colocar nossas próprias páginas – pare o servidor

Criar uma sessão chamada blog



 observe que o app, não é criado dentro da pasta de controle Django (olamundo)

A pasta blog tem a seguinte características



Importante

Migrations (tebelas do banco de dados)

Admin (registro desta sessão administrador desta área)

Apps (deveria ser config.py mas chama-se e assim devemos registrar na settings)

Models (modelos da tabela )

Views – página que será exibida

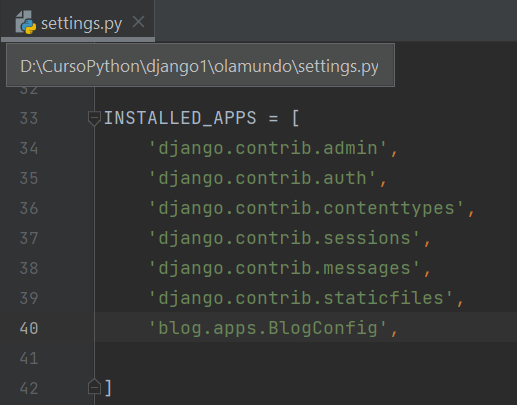
OBS. Precisamos registrar em 2 locais

O que registrar? O nome da classe

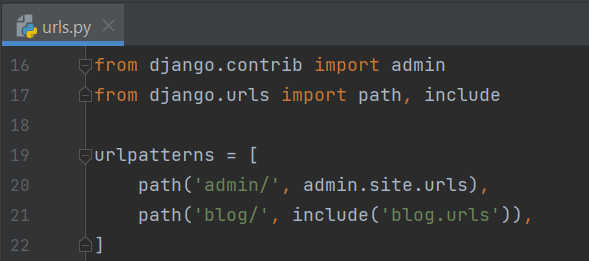


Nome do app - Classe BlogConfig

Em settings.py – INSTALLED\_APPS (linha 40 )



E também na urls.py

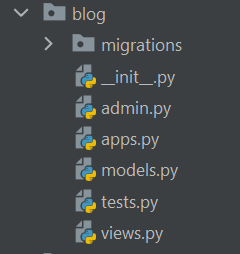
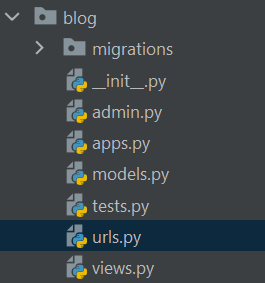


Acrescentar o include no import da linha 17

Na linha 21 acrescentar o path do app.

(observe que dentro do include desta linha devo referenciar o arquivo urls de dentro do app) PRECISAMOS CRIAR ESTE ARQUIVO que será o controlador dentro deste app ok

Pasta sem o arquivo pasta com o arquivo criado

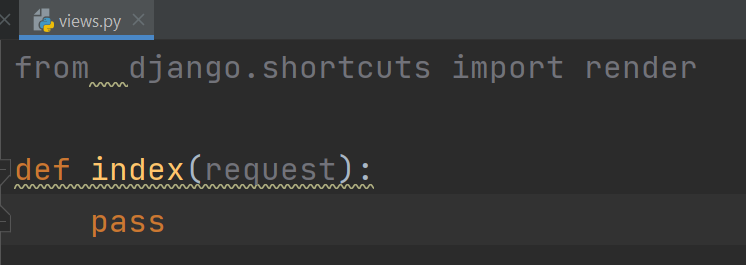


No arquivo urls.py em blog deve conter o seguinte conteúdo



A linha 5 faz chamada do arquivo ***views.py*** que está em blog... o **index** é o método que deve ser executado

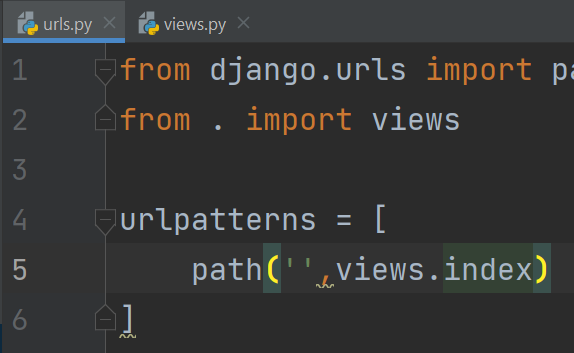
Em views para que o python não reclame coloque o seguinte método



Ao executar o link no navegador qual a sequencia seguida pelo python?

Quando o link blog for chamado ele ira procurar o arquivo: urls..

na url como não existe um endereço completo ele executará o que está na raiz do link...

 o que deverá ser executado neste momento? O arquivo views e o método index..

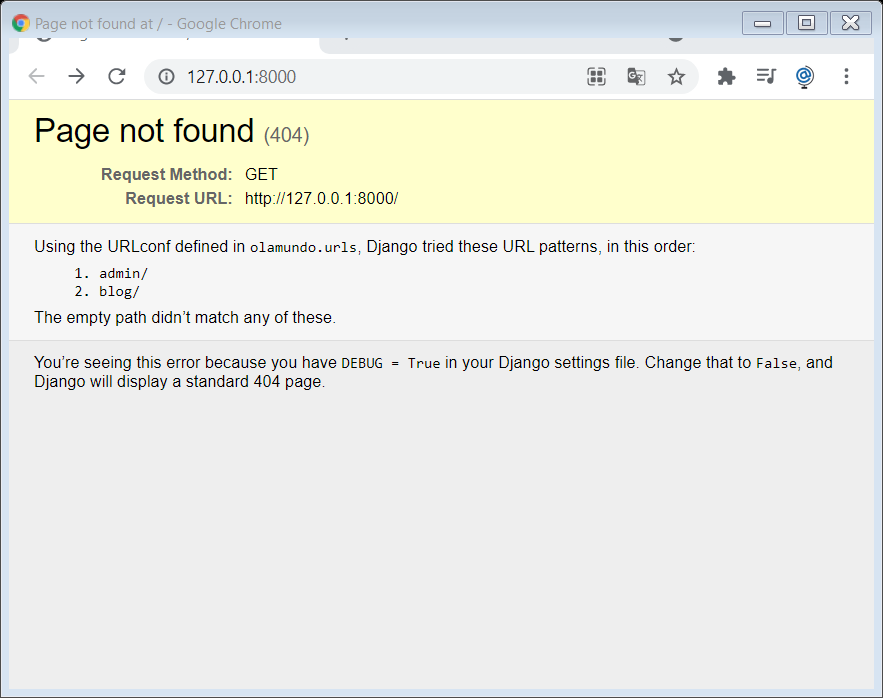
No views.py altere o conteúdo para este abaixo:



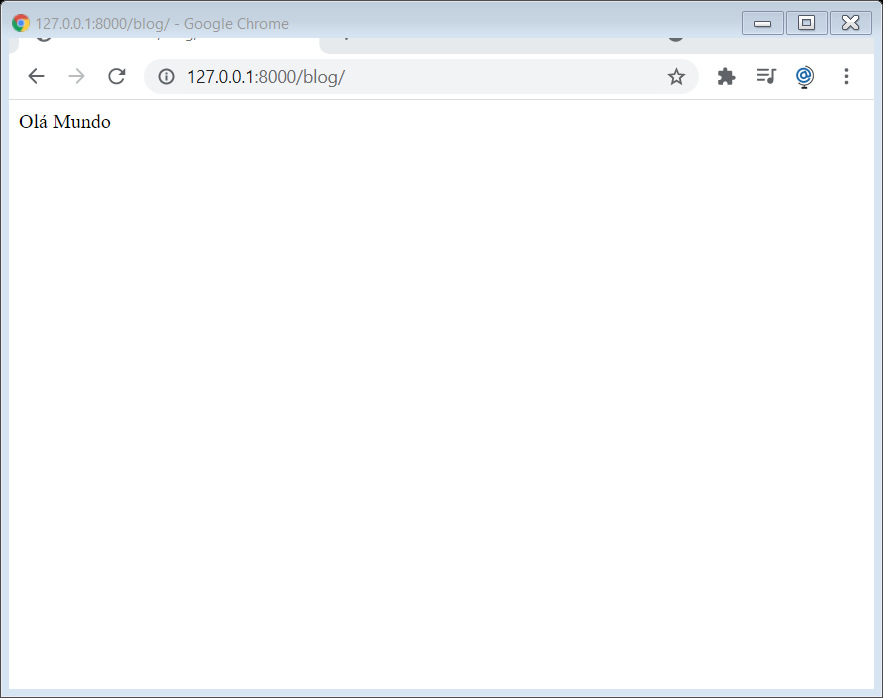
O request sempre receberá um response, padrão do HTTP... não esqueça de importa linha 2

Ao abrir o servidor

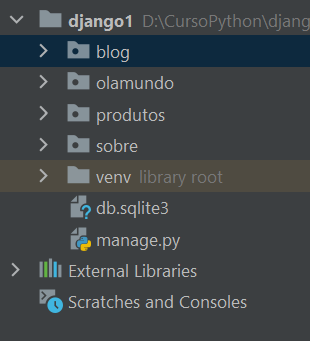
Uma página de “erro” será exibida...



não estamos com conteúdo aplicado na Home... digite a url abaixo



Exercícios



Criar o app produtos

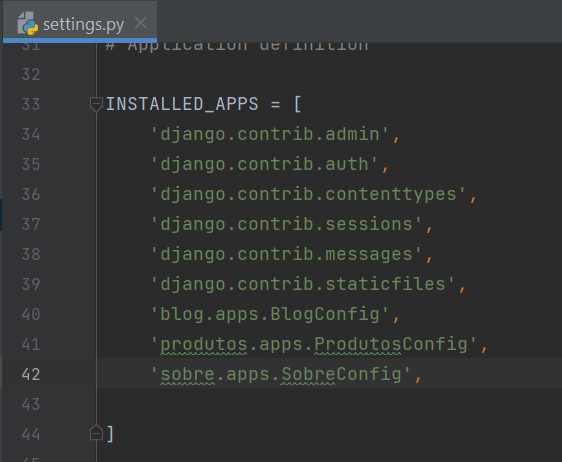


Criar o app sobre

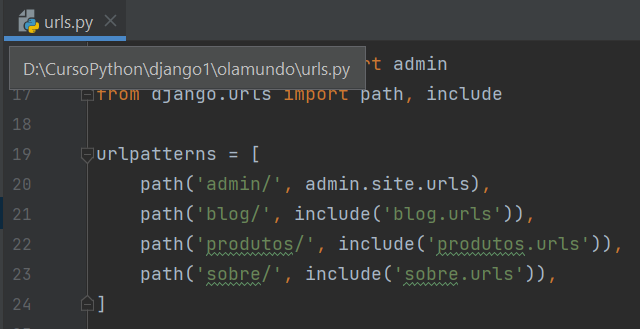


REGISTRAR os dois apps no **settings.py**

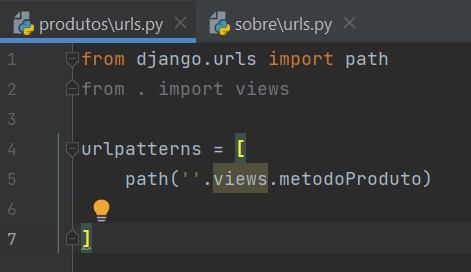
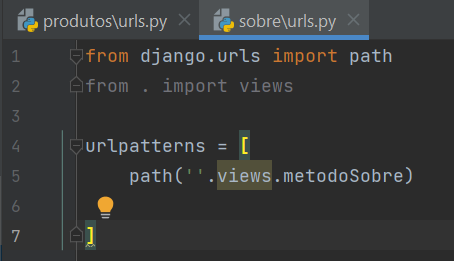
Linhas 41 e 42



Indica na urls.py (cuidado neste ponto este é apontamento do raiz... ok) as rotas



Criar os arquivos urls.py (internos em cada app) para cada método desejado

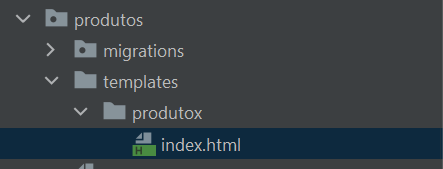


***Observou o erro na linha 5 dos dois arquivos?***

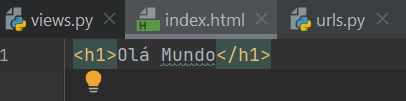
Uso de virgula(,) não ponto( . )

Vamos chamar uma página html em produtos

1. Crie uma pasta chamada **templates** dentro de produtos
2. Dentro dela uma nova chamada **produtox**, nesta pasta deverá existir o index.html

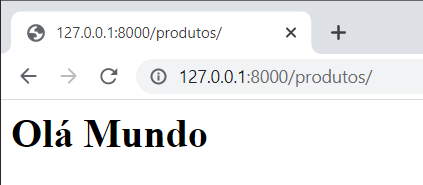


Escreva algo nele



Se você ligar o servidor irá acontecer alguns erros informados no terminal... quais seriam estes erros?

Corriga-os e abra o produtos

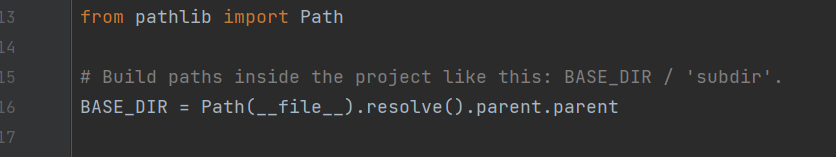


Abra o lorem ipsum html copiar um código e colar em index.html de produtox

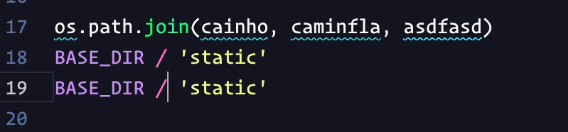
Importante

Na versão 2 alguma coisa o Django para referenciar o arquiv base utilizaa import os

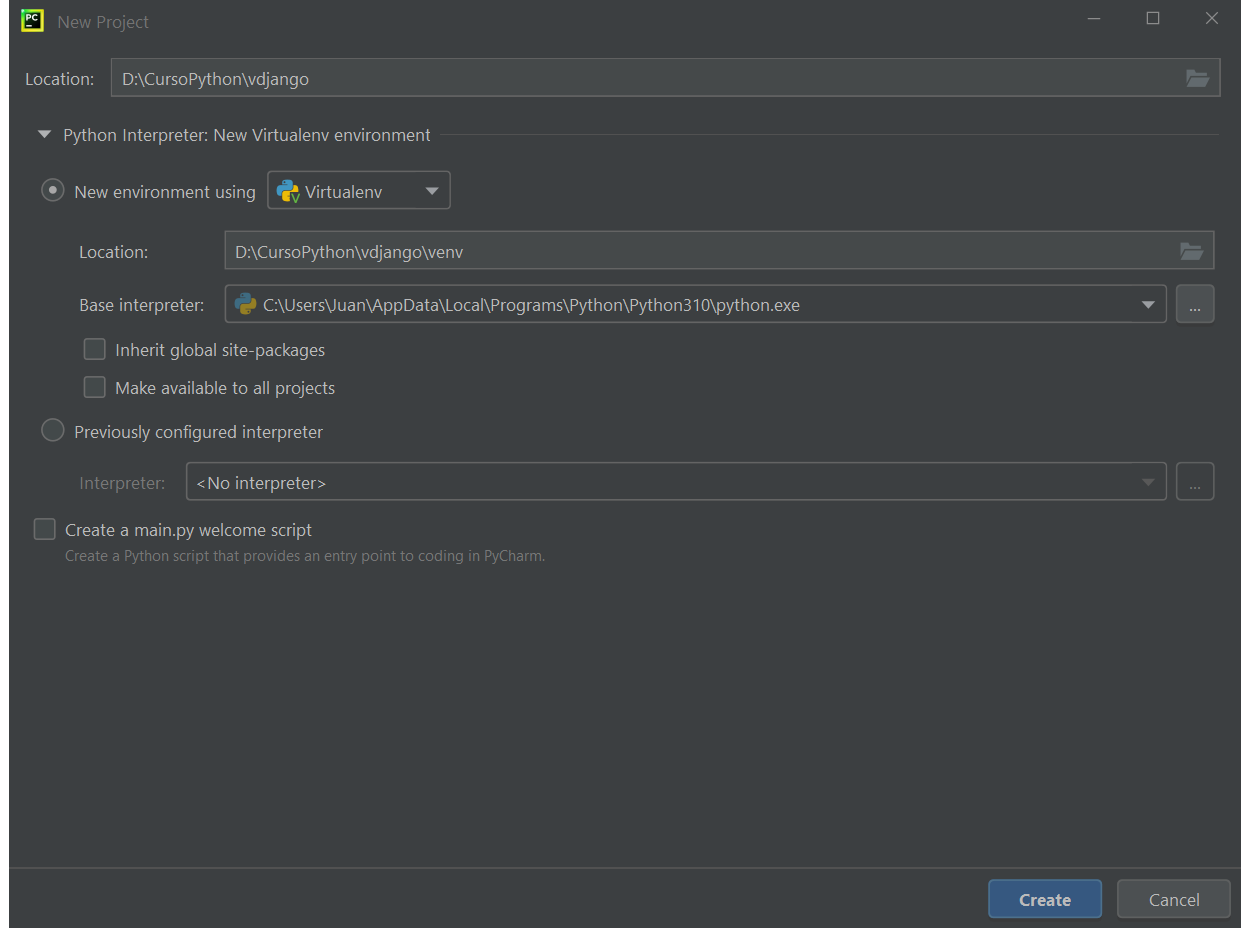
Na versão 3 em diante não usa ele faz uso do PATCH que está na linha 13 do arquivo seetings.py



Se você vir algum comando como o da linha 17 você pode alterar para as linhas 18 e 19 por exemplo ou ainda caso prefira adicionar um **import os** no início do arquivo



Template



Passos:

Instalar o Django

1. Pip install Django



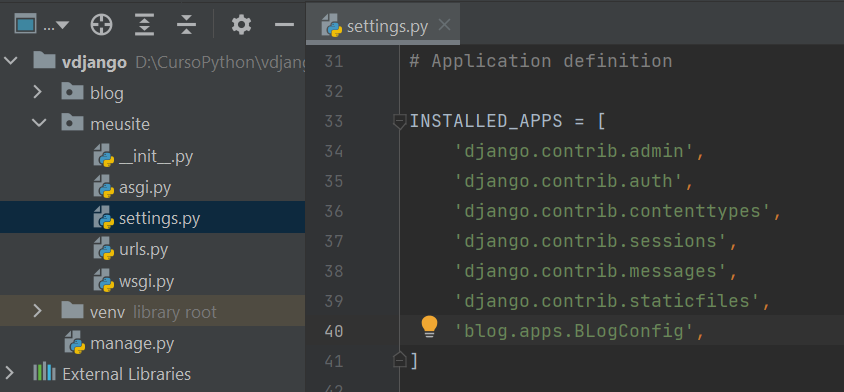
1. Start o projeto



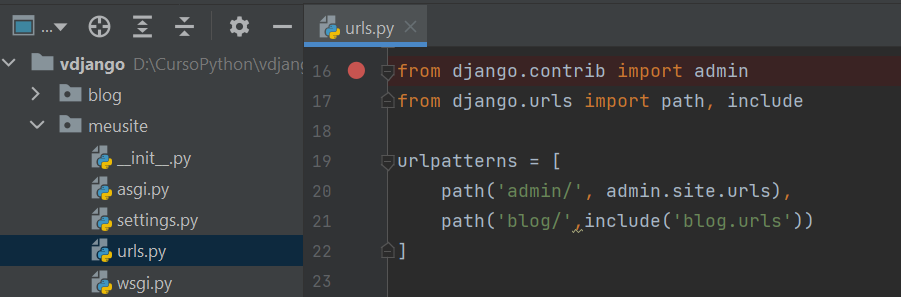
1. Criar os apps no projeto

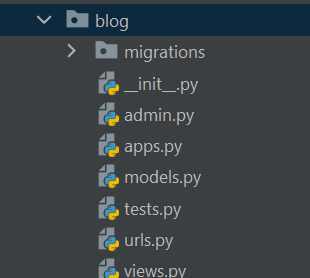


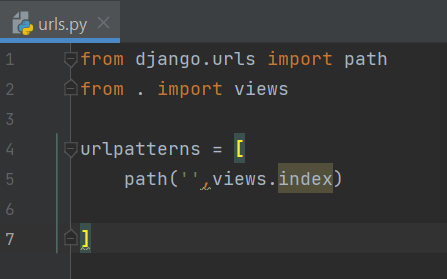
1. Registrar o app no setting.py



1. Registrar na urls.py



Obs.: Criar o arquivo urls.py na pasta blog

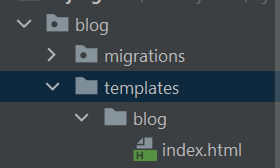


1. Abra views.py



Erro na linha 5 ok nome do arquivo

1. Criar a pasta template, depois uma subpasta blog, e dento dela um arquivo chamado index.html

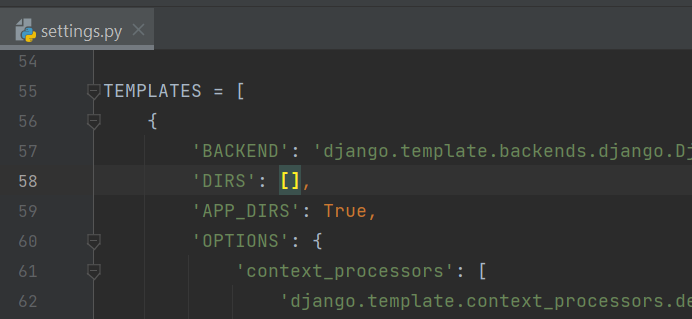


Rode o servidor

Se criarmos um novo app termos que ter um novo HTML

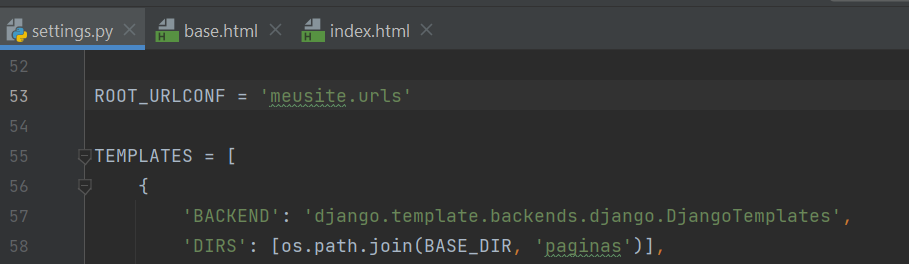
Vamos criar um template **base.html**

Abra o arquivo setting.py para definir um template – observe a linha 58

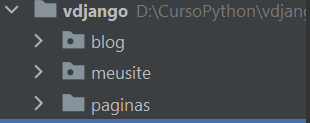


Altere a linha 58

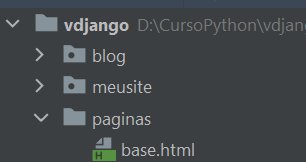
Importe neste arquivo o **import os**



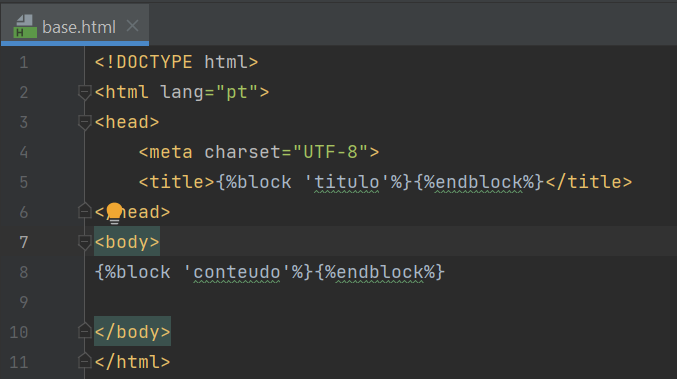
Crie uma pasta chamada paginas na raiz do projeto

 dentro dele um arquivo

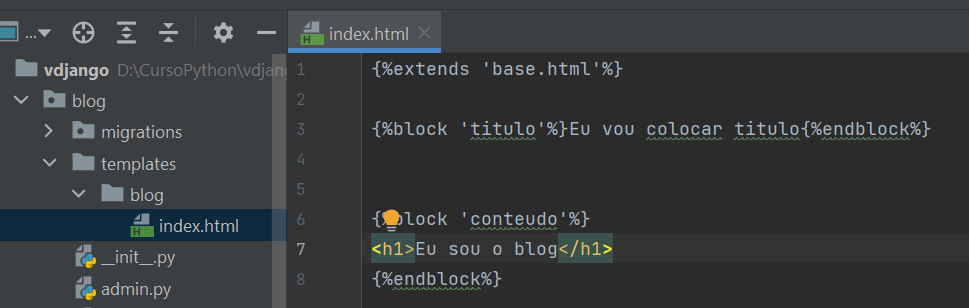
Chamado base.html



Com o seguinte conteúdo



O arquivo index da pasta template/blog deverá ser alterado para

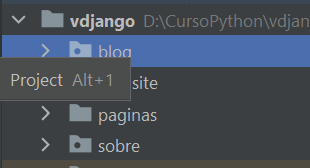


Acesse



Criar novo app





Registrar o app o settings

Colocar na urls

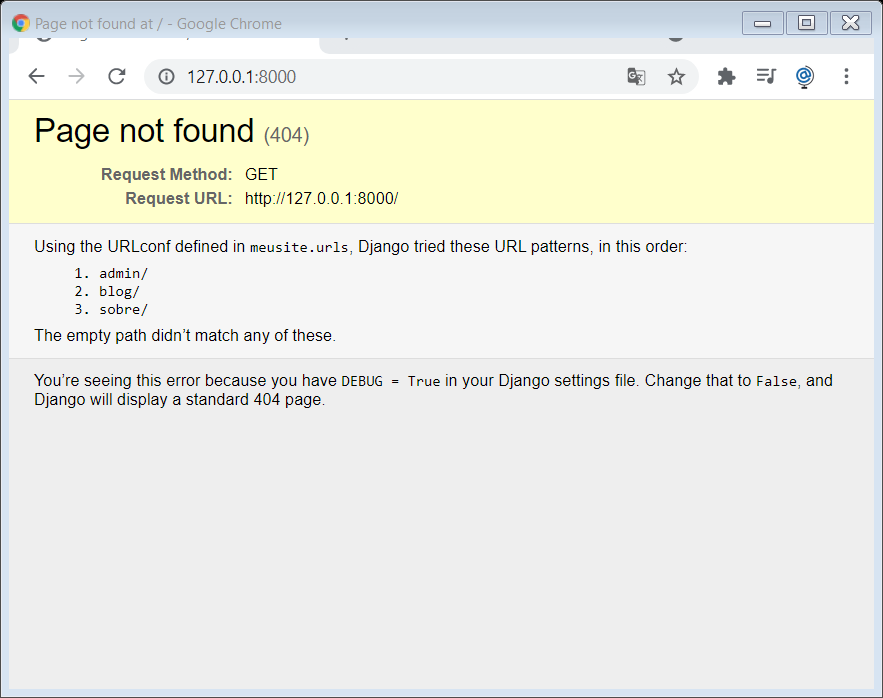
Acrescentar o arquivo urls.py na pasta sobre

Digitar o texto para a view chamar o arquivo index.html

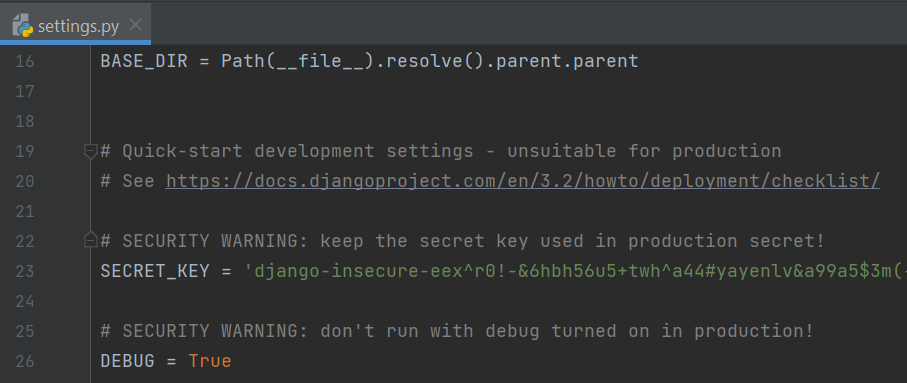
Criar a pasta templates abaixo de sobre

Dentro dele criar o index.html

PÁGINA HOME



Localização em settings



Vamos configurar uma url

Pode criar um app e apontar pra HOME