- django -> criar o backend de websites, atraves do python

- download:

sudo pip install django

para verificar as versoes do python e do django -> pip freeze

- comecar um projeto:

django-admin startproject nome\_do\_projeto(django\_try)

- manage.py -> um programa que auxilia a acessar a db, criar usuarios, runserver, entreoutros

- help -> python manage.py -h

- in django\_try:

\_\_init\_\_.py -> mostra que o projeto criado eh um pacote python

settings.py -> configuracoes gerais do site criado

urls.py -> table of contents of the website, onde voce coloca as urls que acessam os apps do django\_try; user requested and give a response

wsgi.py -> webserver gateway interface, basic overview

- runserver -> python manage.py runserver, crtl C para parar de rodar o server

- conectando/sincronizando o workspace com o db:

python manage.py migrate

obs -> sincroniza os installed apps (settings.py) com o db

- create super user -> python manage.py createsuperuser

- app -> slack n eh um app, eh um programa de computador; app, sim, eh cada parte de um website, tal como em um canal do youtube que possui videos, forum, isso sao apps -> cada app deve ter uma funcao

- criar um app -> python manage.py startapp app\_name(music)

- in django\_try /music:

migrations -> conecta os codigos do app com o db

\_\_init\_\_.py -> mostra que o projeto eh um pacote python

admin.py -> deleta users, posts; sudo

apps.py -> configuracoes do app

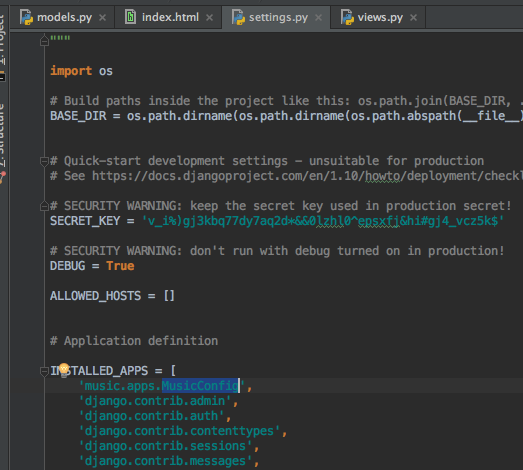
models.py -> a forma de como armazenar a data para esse app(templates)

tests.py

views.py -> pegam um request do usuario e retorno uma resposta(pode ser html response por exemplo, ou um download)

- instalando o app:

em django\_try /setting.py -> adicionar o app:



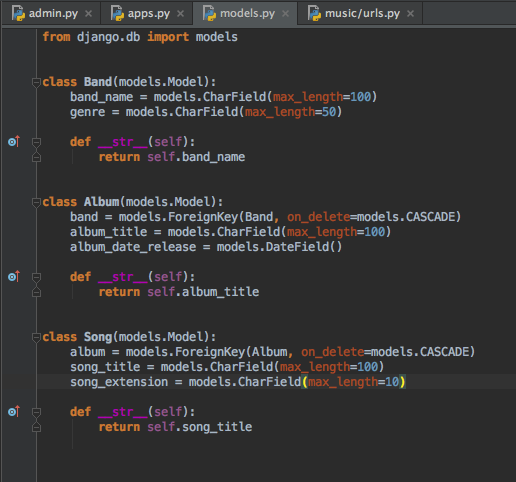
python manage.py makemigrations music

python manage.py migrate

obs -> toda mudança no app, executar as duas linhas acima no terminal

-montando um app:

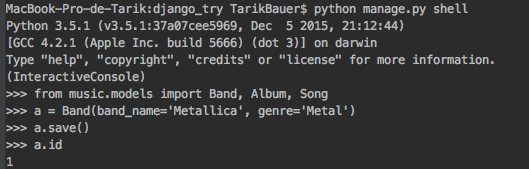
em music/models.py:

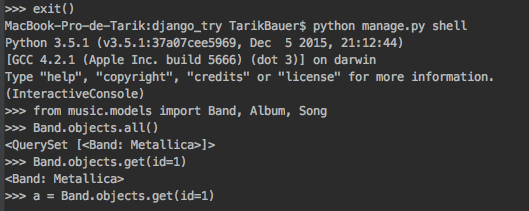


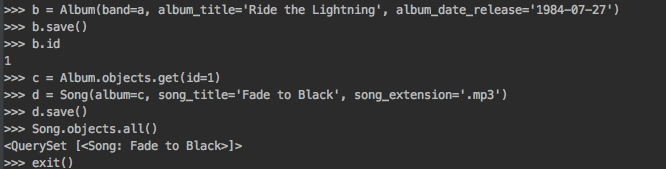
, esse é um exemplo de mudança

- obs: ver as funções de models

para instanciar objetos nessas classes:

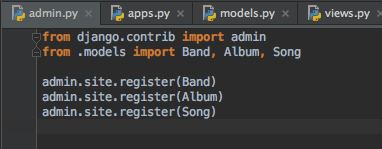






ou:

em music/admin.py:

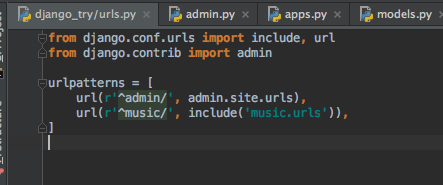


e editar diretamente na url -> .../admin/music

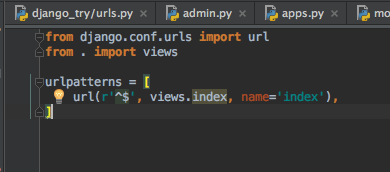
- criando urls para cada app, nesse caso teremos: 127.0.0.1:8000/music/url\_criada:

criar um python file (urls.py) no app criado (music)

em django\_try/urls.py:



em music/urls.py:

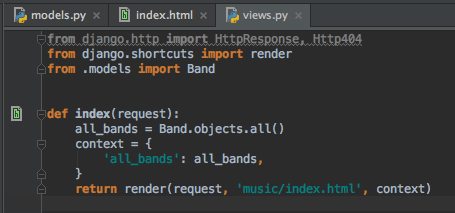


obs -> essa url criada significa o main page do app (music), o seu view se dá no views.index

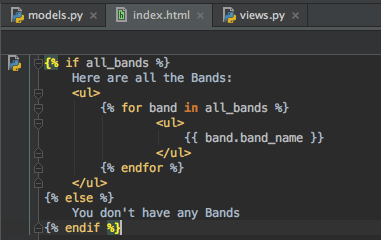
- criando um view:

criar em musica um diretório templates, dentro de templates criar um diretorio music, dentro de music criar index.html

em views:



em index.html:



esse é o básico!