Python programming notes

李鹏飞

2020-01-04

pengfei.li2017@outlook.com

目录

插图

表格

1 Package installation

1.1 pip install using mirror sites

using command: *pip install -i https://pypi.tuna.tsinghua.edu.cn/simple some-package*

or set the mirror permanently using:

pip install pip -U

pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple

1.2 alias 别名

有些情况下，在 linux 系统中，你会发现 pip 安装了某个 module，结果直接在 terminal 运行该 module 命令，却没有任何反映。这时运行 python3 -m IPython 却可以运行，这是因为 IPython 没有注册到系统中，需要运行，alias ipython = “python3 -m IPython”

2 Python 图像处理

2.1 20200104 Image2ExcelData

3 IPython

3.1 Useful commands

execute commands in IPython like in window cmd or linux Terminal.

1. %cd 改变当前的工作目录
2. %edit：打开编辑器，并关闭编辑器后执行键入的代码
3. %env：显示当前环境变量
4. %pip install [pkgs]：无需离开交互式 shell，就可以安装软件包
5. %time 和%timeit：测量执行 Python 代码的时间

4 Django web development

4.1 Django html 路径

```
path('', include("mainpage.urls")),
```

这里网站的 working directory 是 mainpage app 下的 folder. 设置 css 或者图片等 static file 注意事项:

1. 在 app folder 下创建 static 文件夹，如同 templates 一样。
2. 在 static 文件下创建与 app 同名的 appname 文件夹，将所有的 css，img 等放入文件夹中。
3. html template 中写入以下 code：

   ```
   [language = python]
   {%load static%},
   具体语句中，写入 src or href
   = “{% static ‘appname/style/stype.css’ %}”
   ```

   注意这里的 appname 必须要写，这是用以区分，不同 app 使用的不同的 static files。Django 非常不错的地方，就在与如此，Django 可以帮助用户寻找相应的 app 下的 static files。

4.2 Serving Django app over Apache

4.2.1 For pi with default python3.5

There are less problems with python 3.5, however when I attempted to execute the same procedures on my linux computer with Conda environment, there are many issues which will be explained in next section.

1. install required packages:
   (a) update Ubuntu’s package using: sudo apt-get update
   (b) install apache2 and wsgi using: sudo apt-get install python3-pip apache2 libapache2-mod-wsgi-py3
2. Create your Django Apps. Create these Apps using the Django server, because during serving, the server will check your python codes for all Apps.
3. Prepare server configurations. There are three parts. The configuration file is in:

   ```
   /etc/apache2/sites-available/000-default.conf
   ```

   (a) granting access to wsgi.py file
   (b) configure WSGIDaemonProcess directory
   (c) tell apache to serve static files and media files

for example what I have on my pi:

```
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

<Directory /home/pi/Desktop/website_venv/WebsiteProject/django_project>
    <Files wsgi.py>
    Require all granted
    </Files>
</Directory>

WSGIDaemonProcess mysite python-path=/home/pi/Desktop/website_venv/WebsiteProject python-ho
WSGIProcessGroup mysite
WSGIScriptAlias / /home/pi/Desktop/website_venv/WebsiteProject/django_project/wsgi.py

Alias /static /home/pi/Desktop/website_venv/WebsiteProject/static
<Directory /home/pi/Desktop/website_venv/WebsiteProject/static>
Require all granted
</Directory>

Alias /media /home/pi/Desktop/website_venv/WebsiteProject/media
<Directory /home/pi/Desktop/website_venv/WebsiteProject/media>
Require all granted
</Directory>
```

4. check any syntax errors, using: sudo apache2ctl configtest

5. Update settings.py by add:

```
[language=python]
            STATIC_URL = '/static/'
            STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
```

6. collect all static files to a root static folder using: python manage.py collectstatic

7. some references: https://www.digitalocean.com/community/tutorials/how-to-serve-django-applica wsgi-on-ubuntu-16-04 https://omkarpathak.in/2018/11/11/ubuntu-django-apache/

8. potential problems during development:

9. documents permission issue for uploading files: for example:

   ```
   PermissionError at /ProcessPlot/
   [Errno 13] Permission denied:
   '/home/pi/Desktop/website_venv/WebsiteProject/ProcessPlot/documents/YinYang.jpg'
   Request Method: POST
   Request URL: http://192.168.0.102/ProcessPlot/
   Django Version: 2.2.12
   Exception Type: PermissionError
   Exception Value:
   [Errno 13] Permission denied:
   '/home/pi/Desktop/website_venv/WebsiteProject/ProcessPlot/documents/YinYang.jpg'
   Exception Location:
   /home/pi/Desktop/website_venv/lib/python3.5
   /site-packages/django/core/files/storage.py in _save, line 266
   Python Executable: /home/pi/Desktop/website_venv/bin/python
   Python Version: 3.5.3
   Python Path:
   ['/home/pi/Desktop/website_venv/WebsiteProject',
   '/usr/lib/python35.zip',
   '/usr/lib/python3.5',
   '/usr/lib/python3.5/plat-arm-linux-gnueabihf',
   '/usr/lib/python3.5/lib-dynload',
   '/home/pi/Desktop/website_venv/lib/python3.5/site-packages']
   Server time: Sun, 3 May 2020 13:49:26 +0000
   ```

   to solve this problem, use the following command:

   ```
   pi@raspberrypi:~/Desktop/website_venv $
   chmod 664 WebsiteProject/sci_compute/static/sci_compute/img.png
   pi@raspberrypi:~/Desktop/website_venv $
   sudo chown :www-data WebsiteProject/sci_compute/static/sci_compute/img.png
   ```

   if you encounter other permission problems like database or a folder, use same operations, but for folder you have to use **chmod 775**.

10. run apache server: sudo /etc/init.d/apache2 restart, other commands on line don't work on my pi.

11.

4.2.2 For Elementary OS with Conda environment

When I tried to follow the same procedure mentioned above in my Elementary OS, and then I visit 192.168.0.112, which is the ip address of my own OS. The browser always show there is a misconfigure sort of errors, but the apache server runs OK. To solve the problem, first check /var/log/apache2/error.log using vscode **(a good thing with vscode is that it automatically updates its content while it servers the website)**. In error.log, check the information.

```
[Mon May 04 11:21:23.839582 2020] [wsgi:error]
[pid 4137:tid 140078815827712] [remote 192.168.0.112:50656]
ModuleNotFoundError: No module named 'django'

[Mon May 04 11:23:05.274582 2020] [mpm_event:notice] [pid 7359:tid 140297128098752] AH00489:
Apache/2.4.29 (Ubuntu) mod_wsgi/4.5.17 Python/3.6 configured -- resuming normal operations
```

After search with Google, all information points to a malfunctioning virtual environment, I get into the virtual environment where my project locates, and run python manage.py runserver, it seems there is no problem. When run apache2 server, the problem shows again. Keep searching Google, with the information in the error.log, the problem is associated with python version. When using the default installation, sudo apt-get install python3-pip apache2 libapache2-mod-wsgi-py3, the libapache2-mod-wsgi-py3 looks for the default python3 version in my system (by checking the version, you have to deactivate Conda environment, which gives a (base) in terminal). The default python3 is python3.6, which is the mod-wsgi-py3 compiles according to, as what is shown in the error.log.

The problem is now clear, it is the virtual environment I created in the Conda environment, where the python default version is python 3.7 (check the version using python –version). But the wsgi is compiled with the system python3.6. Therefore, these two versions are not compatible.

To solve the problem, deactivate the Conda environment, and use the system python3.6 to create a separate virtual environment, and copy (cp file target dir) to the new virtual environment. Update the 000-default.conf configuration file for the apache server. rerun the server, and it works. However, for a linux system with a python version of 3.7 how to install a lower version of python, to accommodate the libapache2-mod-wsgi-py3 module from the official Ubuntu repo, that is another topic.

For fire wall issues, the Elementary OS installed on my laptop uses *iptables* for firewall. When you want to access your website from other computers, you have to allow traffic to and from the server, in my case, the server is my labtop with Apache2 server. The command to do this is: iptables -A INPUT -m state –state NEW -m tcp -p tcp –dport 80 -j ACCEPT

4.3 Plot data to html

Two methods can be used for plotting data to a html file, there have their own pros and cons.

4.3.1 Plot to a local directory

4.3.2 Plot to memory or buffer

4.4 Create a new django app

The right sequences are listed below:

1. run python manage.py startapp newapp

2. cp previous_app_folder \urls.py newapp, and change corresponding scripts.

3. copy the url name after views.method to the main page index template.

4. In django_project (our main website project folder): copy class name for example, ProcessPlotOptimizeConfig from the app.py file in the newapp folder, to settings.py file, add the new app to the system settings.

5. In django _project (the main project): modify the urls.py file, add a new line as below:

   ```
   path("process_plot_optimize/", include("process_plot_optimize.urls")),
   ```

   this is very import, I forgot one time, the server gives reverse error. and you have to restart the server.

5 Next step

1. go through my new app, the process_plot_optimize app, edit the template models and other parts.

2. go through django session, to see how we deal with concurrence.

6 文献

[?]

参考文献