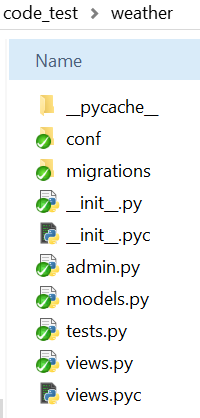
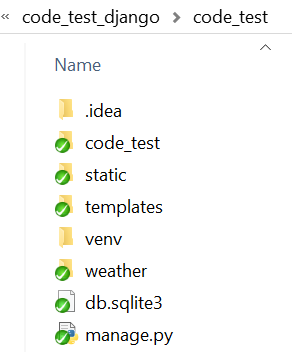
1. **项目搭建**

**1 项目结构搭建**

1. 执行django-admin.py startproject code\_test，创建code\_test项目；
2. 执行python manage.py startapp weather，创建weather应用；
3. 进入项目工程目录code\_test下，创建文件夹static，用来存放js、css、图片等静态资源；
4. 在code\_test目录下，创建templates文件夹，用来存放前端的html静态模板；
5. 进入weather应用目录下，创建conf文件夹，用来存放该应用的配置文件，比如城市的配置文件。

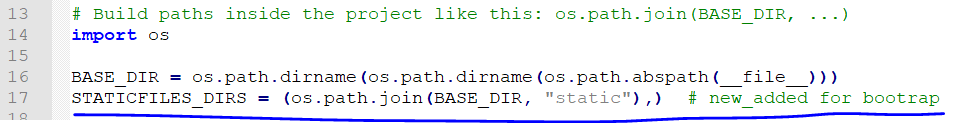
以上五步完成后，项目结构基本搭建完成，如图：



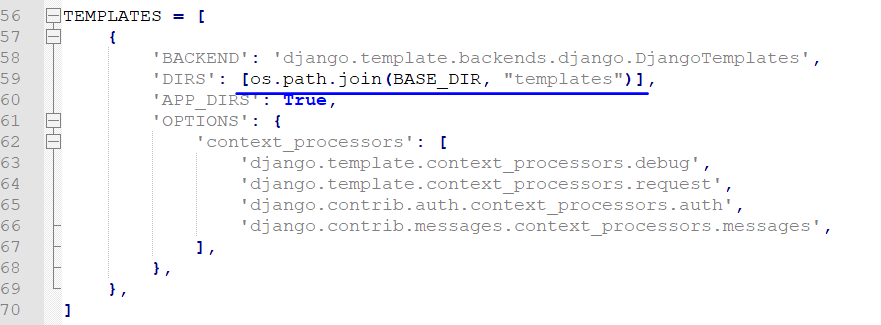
**2 项目设置及启动**

1. 进入项目工程目录code\_test\code\_test下，打开settings.py，进行一些项目的配置。

* 设置静态资源读取路径，如图：



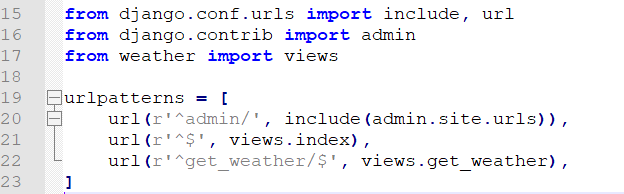
* 设置项目模板的读取路径，如图：



* 设置html模板中引入静态资源的基本路径：STATIC\_URL = '/static/'

此外，如果我们需要添加新的app应用，需要在INSTALLED\_APPS中引入。如需要设置数据库，则需要修改DATABASES的“ENGINE”、“HOST”、“PORT”、“NAME”、“USER”和“PASSWORD”。

1. 打开同目录下的urls.py文件，我们可以配置前端模板和view视图关联的路由，如图：



1. 进入到weather应用目录中，打开views.py文件，编辑对应的视图函数index和get\_weather。
2. 进入到templates目录中，添加模板index.html。
3. 进入到static目录中，添加index.html模板使用的js、css资源文件。
4. 进入到weather/conf目录下，添加城市配置文件city.txt。
5. 在weather/tests.py中编写测试用例。
6. 如果需要从数据库取数据，在settings.py中配置好DATABASES信息之后，可在models.py中建立model类。每个model类对应一个数据库表，每个属性对应一个数据库表的属性。表名可在class Meta的db\_table中配置。

然后，执行数据迁移命令：

python manage.py makemigrations

python manage.py migrate

1. 最后，执行python manage.py runserver即可启动。

**3 项目运行思路**

db

models.py

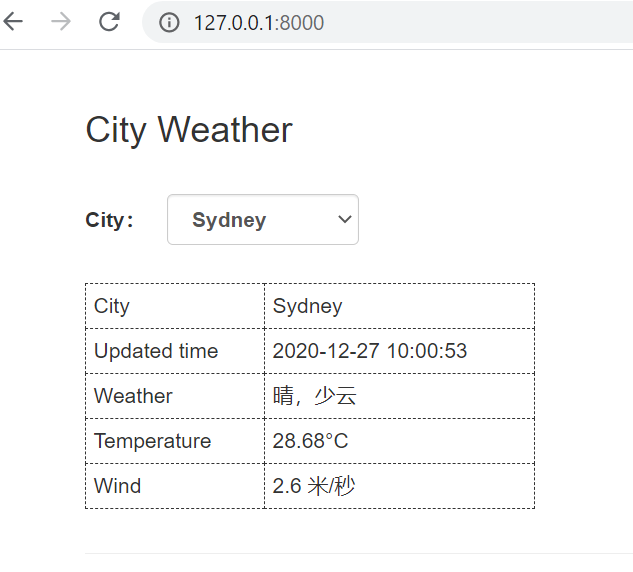
views.py

urls.py

template

1. **屏幕快照**

谷歌浏览器访问地址：<http://127.0.0.1:8000/>，可看到各城市天气情况，如图：



城市下拉框数据可在city.txt文件中进行配置。切换下拉框的值之后，可动态展示不同城市当前天气情况。

1. **云部署**

**1 申请华为云服务器**



**2 项目部署**

**1、搭建python环境**

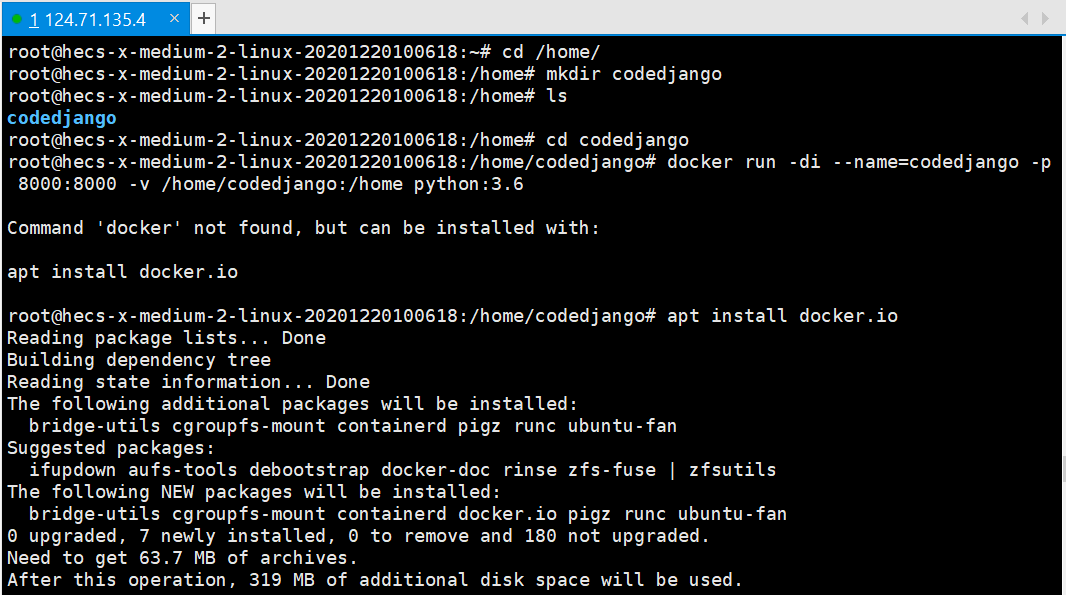
1. 在开发环境执行pip3 freeze > a.txt，生成安装文件a.txt。
2. XShell登录云服务器:

cd /home/

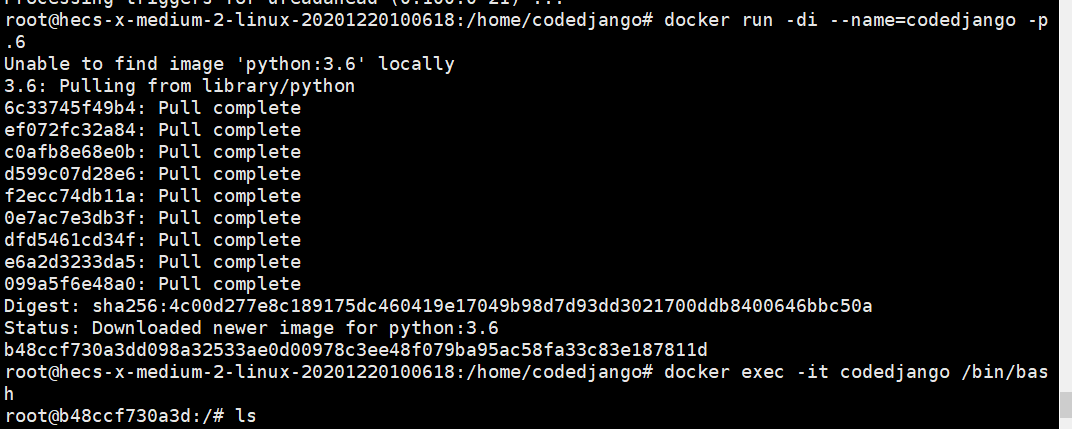
mkdir codedjango

cd codedjango

apt install docker.io

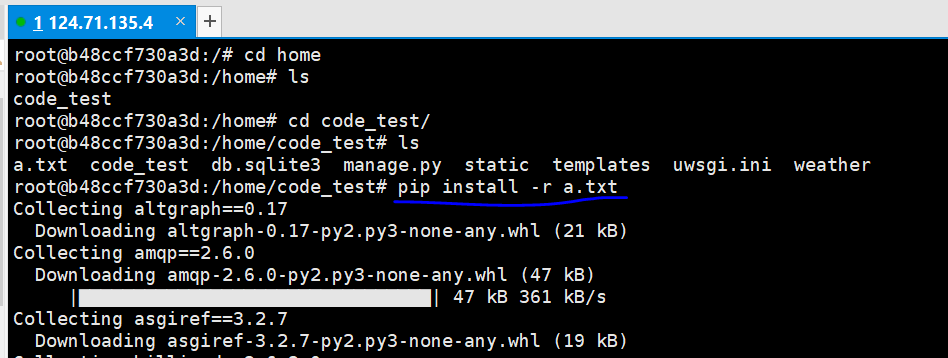


docker run -di –name=codedjango -p 8000:8000 -v /home/codedjango:/home python3.6



将code\_test项目代码拷贝到codedjango文件夹下，然后进入容器。

docker exec -it codedjango /bin/bash



cd /home/

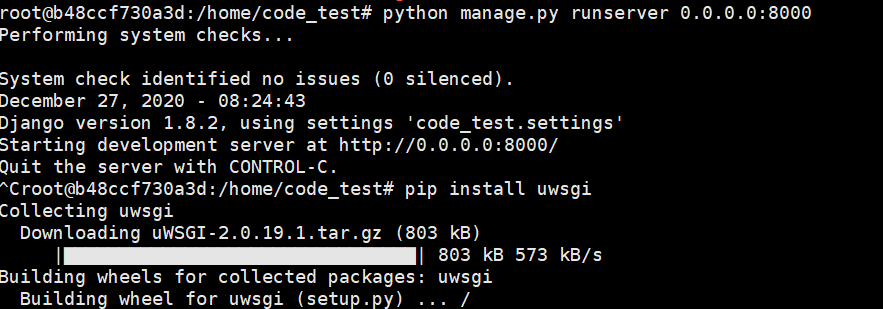
cd /code\_test/

pip install -r a.txt

apt-get update

apt-get install vim-gtk

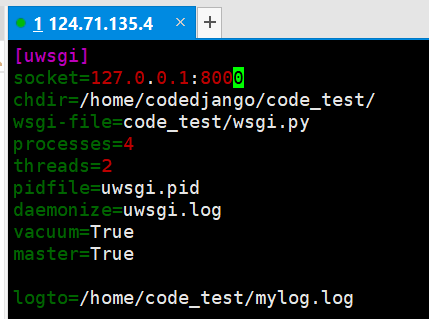
python manage.py runserver 0.0.0.0:8000，启动成功。



**2、安装uwsgi**

pip install uwsgi

在code\_test项目目录下创建uwsgi.ini文件

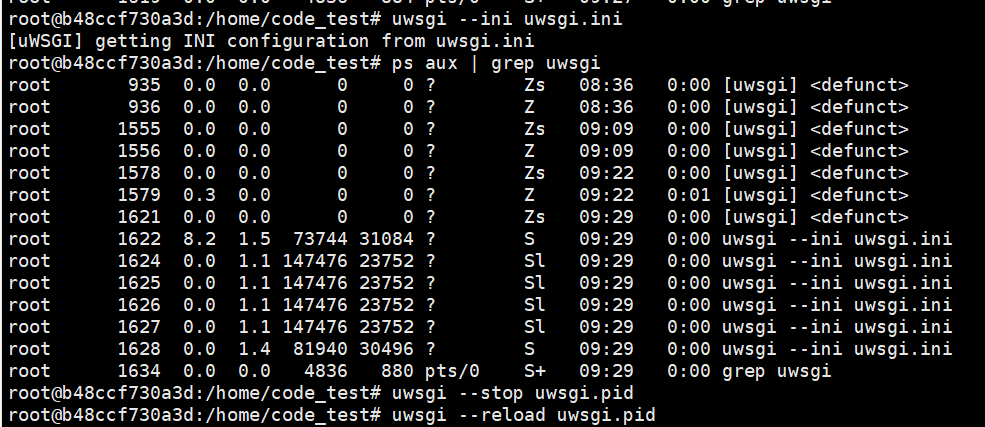


uwsgi --ini uwsgi.ini #启动

ps aux | grep uwsgi　　　#按照程序名查询

uwsgi --stop uwsgi.pid #通过uwsg停止uwsgi

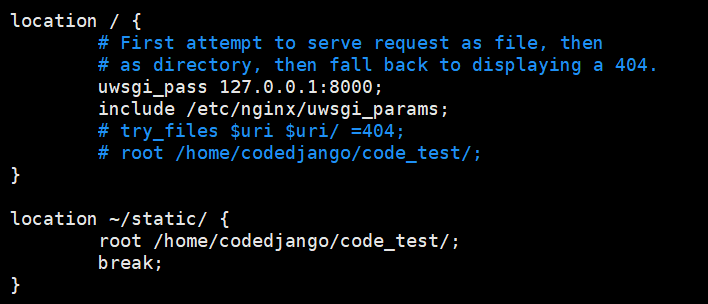
uwsgi --reload uwsgi.pid #重启



**3、nginx配置**

sudo apt install nginx # 安装

sudo vim /etc/ngnix/sites-available/default # 修改nginx配置



sudo /etc/init.d/ngnix start # 启动

sudo /etc/init.d/ngnix restart # 重启

