

# python-extra

## 虚拟环境

- 类似于虚拟机、容器，python可以创建一个虚拟环境
- 一个虚拟环境就是一个目录
- 可以将python的软件包安装在虚拟环境目录
- 将来该环境不需要了，只要删除目录即可

```
# 创建名为dj2env的虚拟环境
[root@localhost ~]# python3 -m venv ~/dj2env
[root@localhost ~]# ls -d dj2env
dj2env

# 激活虚拟环境
[root@localhost ~]# source ~/dj2env/bin/activate
(dj2env) [root@localhost ~]#
(dj2env) [root@localhost ~]# python --version
Python 3.6.7

# 在虚拟环境中安装 django2.2.12
(dj2env) [root@localhost ~]# pip install django==
(dj2env) [root@localhost ~]# pip install django==2.2.12
(dj2env) [root@localhost ~]# python -m django --version
2.2.12
```

## 使用中的问题

- 直接使用sqlite作为数据库，提示sqlite版本过低，无法运行
- 解决方案是升级sqlite3

## CentOS升级sqlite3

- 获取源代码

```
[root@localhost ~]# cd ~
[root@localhost ~]# wget https://www.sqlite.org/2020/sqlite-autoconf-3310100.tar.gz
[root@localhost ~]# tar xvf sqlite-autoconf-3310100.tar.gz
```

- 构建并安装

```
[root@localhost ~]# cd sqlite-autoconf-3310100/
[root@localhost sqlite-autoconf-3310100]# ./configure --prefix=/usr/local
[root@localhost sqlite-autoconf-3270200]# make && make install
[root@localhost ~]# find /usr/ -name sqlite3
/usr/bin/sqlite3
/usr/lib64/python2.7/sqlite3
/usr/local/bin/sqlite3
/usr/local/lib/python3.6/sqlite3
```

- 检查版本 最新安装的sqlite3版本

```
[root@localhost ~]# /usr/local/bin/sqlite3 --version
3.31.1 2020-01-27 19:55:54
3bfa9cc97da10598521b342961df8f5f68c7388fa117345eeb516eaa837bb4d6
```

- sqlite3的版本是旧版本，需要更新。

```
[root@localhost ~]# mv /usr/bin/sqlite3 /usr/bin/sqlite3_old
[root@localhost ~]# ln -s /usr/local/bin/sqlite3 /usr/bin/sqlite3
```

- 查看当前全局sqlite3的版本

```
[root@localhost ~]# sqlite3 --version
3.31.1 2020-01-27 19:55:54
3bfa9cc97da10598521b342961df8f5f68c7388fa117345eeb516eaa837bb4d6
```

- 将路径传递给共享库，将下面的export语句写入~/.bashrc并source它

```
export LD_LIBRARY_PATH="/usr/local/lib"
```

- 检查Python的SQLite3版本

```
[root@localhost ~]# python3
Python 3.6.7 (default, Jun 11 2020, 08:58:45)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-39)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sqlite3
>>> sqlite3.sqlite_version
'3.31.1'
>>>
```

## django2.2.12使用mysql数据库

```
# 创建名为mysite2的项目后
(dj2env) [root@localhost mysite2]# pip install pymysql

# 创建数据库
[root@localhost ~]# mysql -uroot -ptedu.cn
MariaDB [(none)]> CREATE DATABASE dj2mysite2 DEFAULT CHARSET utf8;

# mysite2/settings.py
... ..
ALLOWED_HOSTS = ['*']
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'dj2mysite2',
        'USER': 'root',
        'PASSWORD': 'tedu.cn',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}
LANGUAGE_CODE = 'zh-hans'

# mysite2/__init__.py
import pymysql

pymysql.install_as_MySQLdb()

# 运行开发服务器
(dj2env) [root@localhost mysite2]# python manage.py runserver

# 报错如下：
django.core.exceptions.ImproperlyConfigured: mysqlclient 1.3.13 or newer is required;
you have 0.9.3.

# 解决方案如下：
[root@localhost ~]# cd ~/dj2env/lib64/python3.6/site-packages/django/db/backends/mysql/
[root@localhost mysql]# vim base.py
# 注释掉第35和36行
... ..
# if version < (1, 3, 13):
#     raise ImproperlyConfigured('mysqlclient 1.3.13 or newer is required; you have %s.'
% Database.__version__)
... ..
[root@localhost mysql]# vim operations.py
# 将146行的decode改为encode
... ..
    if query is not None:
        query = query.encode(errors='replace') # 146
    return query
... ..
```

## 使用django2编程

- 与django 1.11.x一样
- 在django 2.2.x中，将1.11.x中的url函数换成re\_path，即可

```
# 例：
(dj2env) [root@localhost mysite2]# python manage.py startapp polls

# mysite2/settings.py
INSTALLED_APPS = [
    ...
    'polls',
]

# mysite2/urls.py
from django.contrib import admin
from django.urls import path, re_path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    re_path('^polls/', include('polls.urls')),
]
```