## Django+vue.js初探索

主要讲解内容：

1. Python环境及配置
2. PyCharm安装
3. 创建虚拟环境及管理
4. 新建django工程
5. Django工程配置Xadmin
6. Vue环境搭建与Django整合

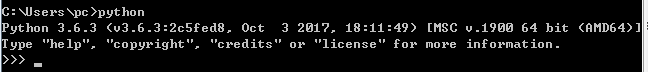
### 1、安装python环境：

1.1、python官网：<https://www.python.org/downloads/>

1.2、选择版本，这里选择3.6版本

1.3、配置python安装路径到系统环境变量

1.4、查看安装版本命令：python



### 2、PyCharm安装

2.1、PyCharm官网：

<https://www.jetbrains.com/pycharm/download/#section=windows>

2.2、选择专业版，下载安装

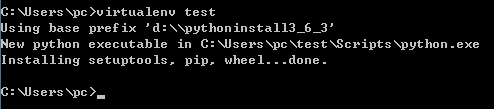
### 3、创建虚拟环境及管理

3.1、安装命令

pip install virtualenv

3.2、创建一个虚拟环境

virtualenv test



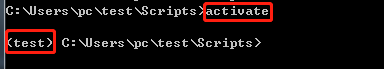
在目录C:\Users\pc\路径下会看到一个test文件夹

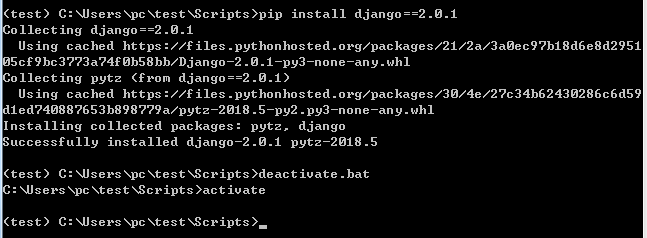
考虑到系统可能有多个版本的python环境，创建环境时也可以指定版本来创建需要的环境，命令如下：

virtualenv -p D:\pythonInstall3\_6\_3\python.exe test\_1

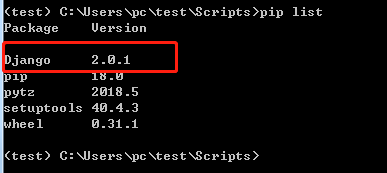
(virtualenv -p 安装python的路径 名字)

3.3、进入虚拟环境test下的Script目录，激活虚拟环境，会看到如下图，路径前面多了（test）标识，表示进入到虚拟环境中，现在在可以安装项目依赖库，现安装django依赖库作为测试：





查看安装结果：



3.4、退出虚拟环境命令



（test）标识消失

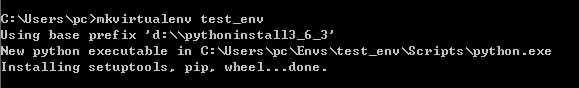
缺点：如果有多个这样的虚拟环境，要安装依赖库需要到相应的目录下安装，到另一个环境时又要切换到对应目录，命令比较繁琐。

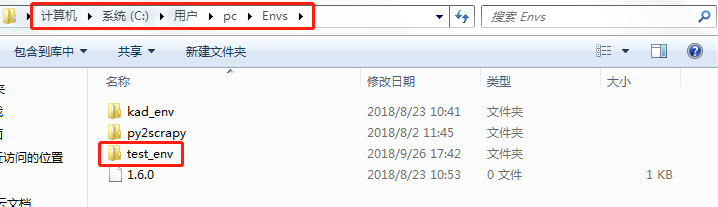
3.4、安装虚拟环境管理

pip install virtualenvwrapper-win #windows下安装要加-win

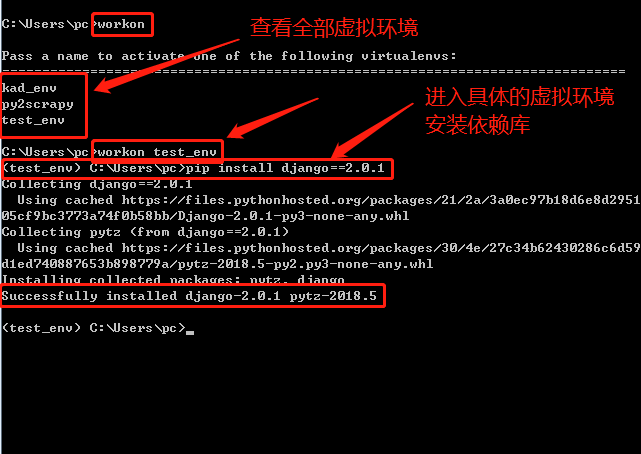
3.5、使用管理来创建虚拟环境

Mkvirtualenv test\_env，系统默认会在C:\Users\pc目录下新建envs文件夹（这个可以自行配置），里面存放同意管理新建的虚拟环境，如下图：



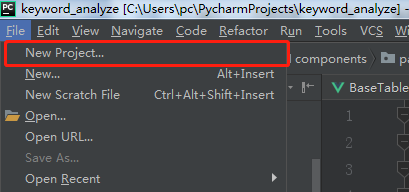


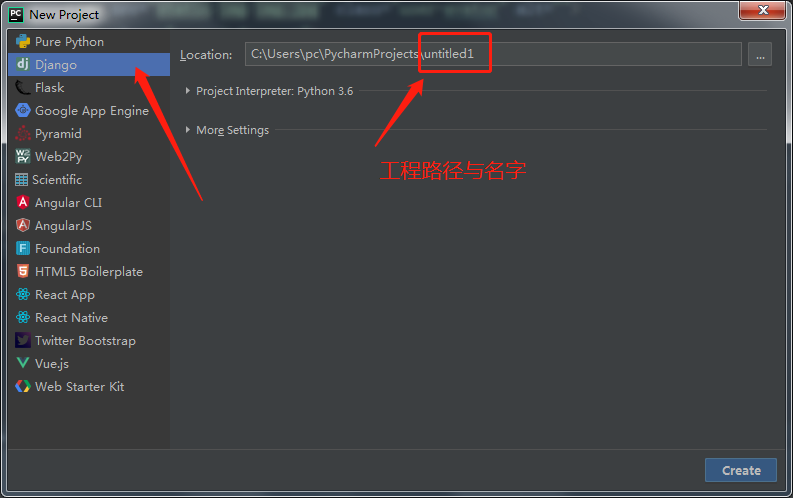
好处：各个虚拟环境不冲突，多个环境切换方便，容易管理，像workon test\_env即可进入激活状态，安装依赖包。如图：



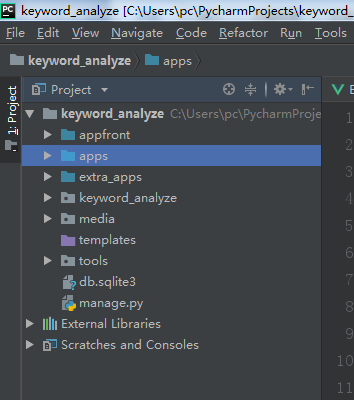
### 新建Django工程

4.1、新建一个工程





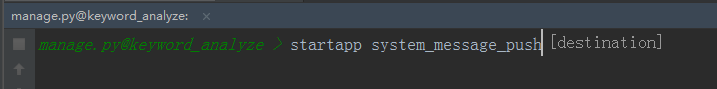
4.2、新建文件夹，把各类文件分类



appfront文件夹存放前端工程，apps文件夹存放自己定义的类，extra\_apps文件夹存放其他的类，本项目存放xadmin源码，media文件夹存放图片、音频等，tools文件夹存放工具文件。

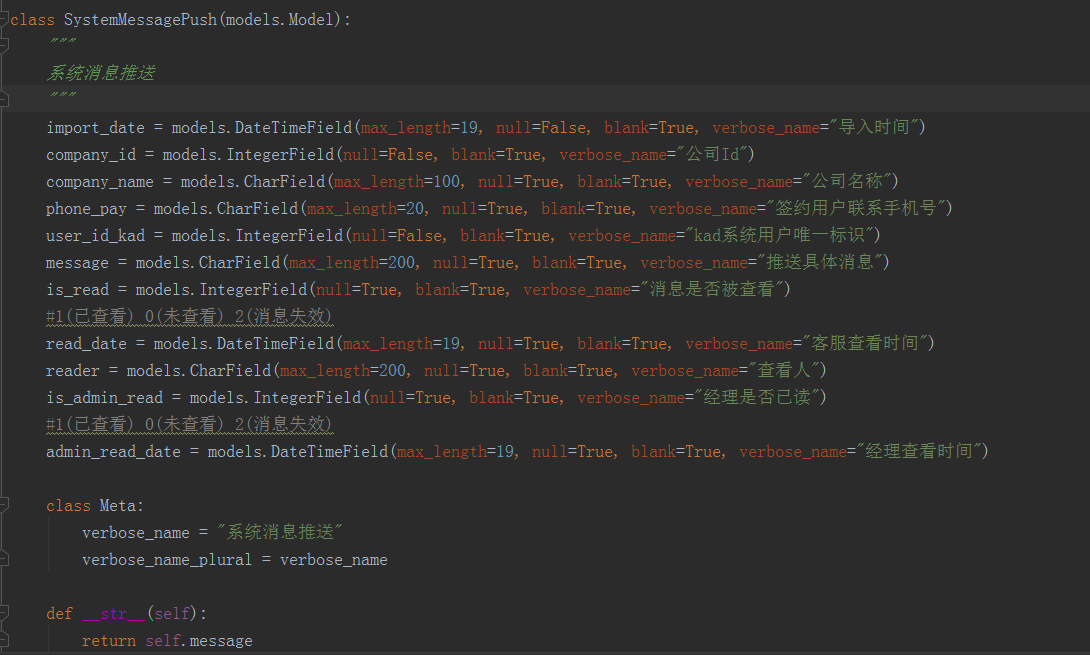
4.3、测试新建一个app

4.3.1、首先在tools菜单栏点击run manage.py Task...选项：



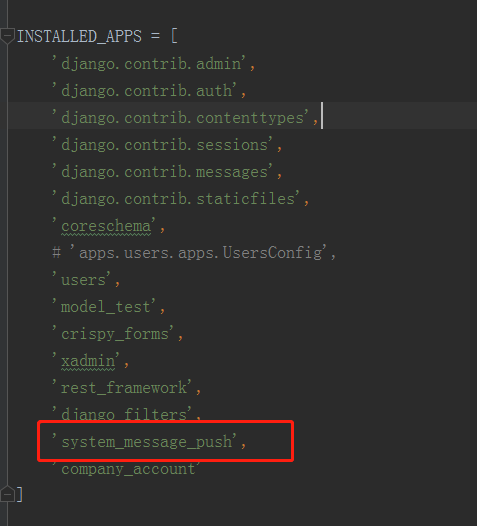
等待新建system\_message\_push文件夹完成，将改文件夹拖入到apps文件中

4.3.2、在system\_message\_push文件夹下model.py文件写一个model:



4.3.3、并把app加入到installed\_apps列表里：

在keyword\_analyze文件夹的setting文件夹



4.4、在keyword\_analyze下的settings.py配置文件中，把默认的sqllite3数据库换成我们的mysql数据库

# Database  
# https://docs.djangoproject.com/en/2.1/ref/settings/#databases  
  
DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.mysql',  
 'NAME': "ysanalyze", #数据库名  
 'USER': 'root', #用户名  
 'PASSWORD': "123456", #密码  
 'PORT': 3306, #端口号  
 'HOST': "127.0.0.1" #IP  
 }  
}

注意！这里需要安装数据库连接依赖（mysqlclient），不能用pip安装，需要下载库自行安装；下载连接：<https://www.lfd.uci.edu/~gohlke/pythonlibs/>

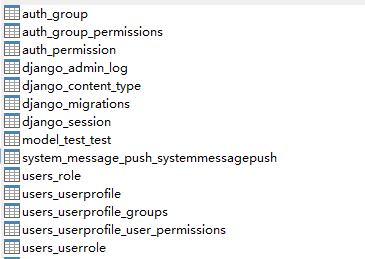
4.5、生成数据库表

Tools->run manage.py Task...，顺序执行两个命令：

makemigrations

migrate

执行完命令，可以看到数据库生成了自行定义的表和自带的一些表。



### Django工程配置Xadmin

为了方便管理，有些model需要到注册app到xadmin后台；

5.1、下载xadmin源码，加入到extra\_apps文件夹中

Xadmin源码地址：<https://github.com/sshwsfc/xadmin>

将下载内容下xadmin文件夹拷贝到extra\_apps目录下

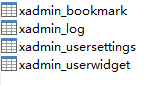
5.2、安装xadmin依赖库（有下面6个库）

* [django](http://djangoproject.com/) >=1.9
* [django-crispy-forms](http://django-crispy-forms.rtfd.org/) >=1.6.0 (For xadmin crispy forms)
* [django-reversion](https://github.com/etianen/django-reversion) ([OPTION] For object history and reversion feature, please select right version by your django, see [changelog](https://github.com/etianen/django-reversion/blob/master/CHANGELOG.rst) )
* [django-formtools](https://github.com/django/django-formtools) ([OPTION] For wizward form)
* [xlwt](http://www.python-excel.org/) ([OPTION] For export xls files)
* [xlsxwriter](https://github.com/jmcnamara/XlsxWriter) ([OPTION] For export xlsx files)

5.3、将xadmin、crispy\_forms加入到INSTALLED\_APPS中

INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
 'coreschema',  
 'users',  
 'model\_test',  
 'crispy\_forms', #如果项目提示则加入这个  
 'xadmin', #加入xadmin  
 'rest\_framework',  
 'django\_filters',  
 'system\_message\_push',  
]

重新执行makemigrations、migrate命令，生成xadmin相关表



5.4、在app(system\_message\_push)下新建adminx.py文件，内容为：

import xadmin  
from .models import SystemMessagePush  
from xadmin import views  
class SystemMessagePushAdmin(object):

list\_display = ["company\_name", "phone\_pay", "message", "reader"] #控制显示的列  
 search\_fields = ["company\_name", "phone\_pay", "message", "reader"] #控制搜索的列  
 list\_filter = ["company\_name", "phone\_pay", "message", "reader"] #控制过滤的列  
 def queryset(self): #查询语句  
 qs = super(SystemMessagePushAdmin, self).queryset()  
 qs = qs.filter(is\_read=0, is\_admin\_read=0, user\_id\_kad=271)  
 return qs

xadmin.site.register(SystemMessagePush, SystemMessagePushAdmin)

#注册到xadmin管理

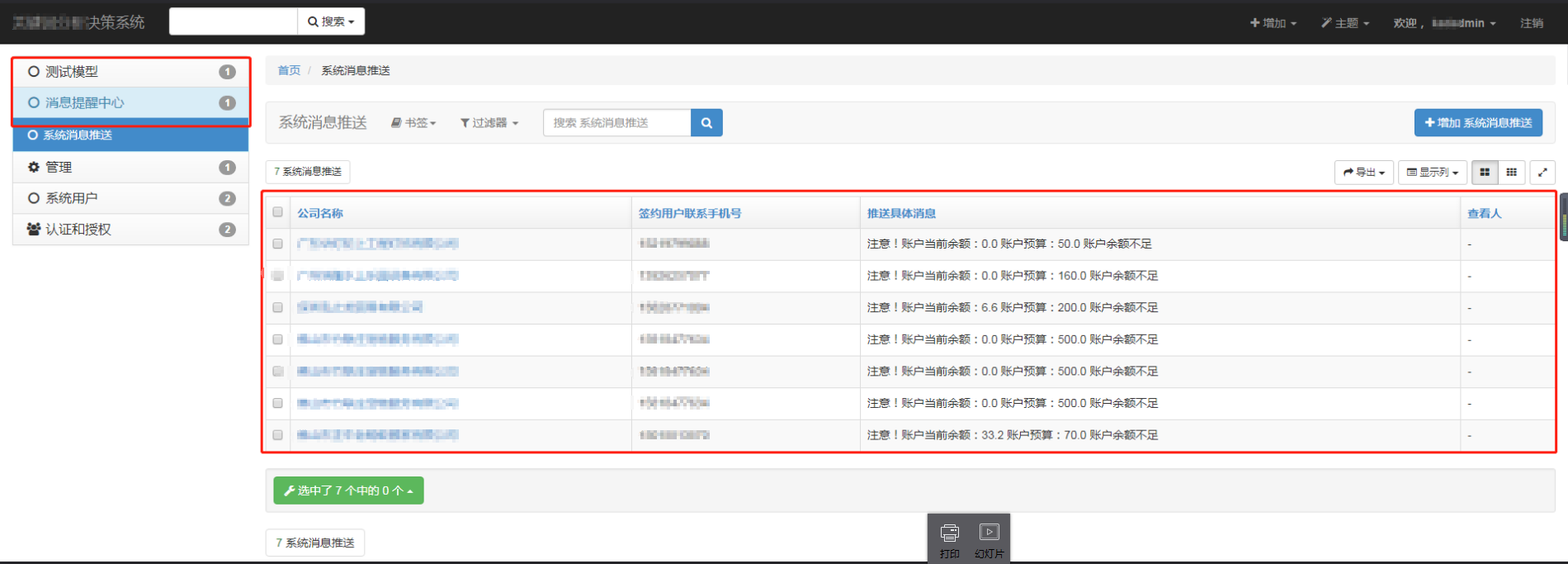
5.5、为xadmin后台创建超级管理员账户，tools->run manage.py Task ...

执行 createsuperuser 命令

按提示创建账号

5.6、访问xadmin后台：<http://127.0.0.1:8000/xadmin/>

可以看到后台效果图：

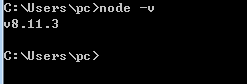


### Vue环境搭建与Django整合

6.1、安装vue环境

6.1.1、下载node.js安装包，链接：<http://nodejs.cn/download/>

查看版本：



6.1.2、安装淘宝镜像，下载较快（个人习惯）

npm install -g cnpm --registry=https://registry.npm.taobao.org

#### 安装 vue-cli（Vue 提供了一个[官方的 CLI](https://github.com/vuejs/vue-cli" \t "https://cn.vuejs.org/v2/guide/_blank)，为单页面应用 (SPA) 快速搭建繁杂的脚手架。它为现代前端工作流提供了 batteries-included 的构建设置。只需要几分钟的时间就可以运行起来并带有热重载、保存时 lint 校验，以及生产环境可用的构建版本。）

$ npm install --global vue-cli

6.3、创建基于vue的前端工程，这里介绍两种方式：

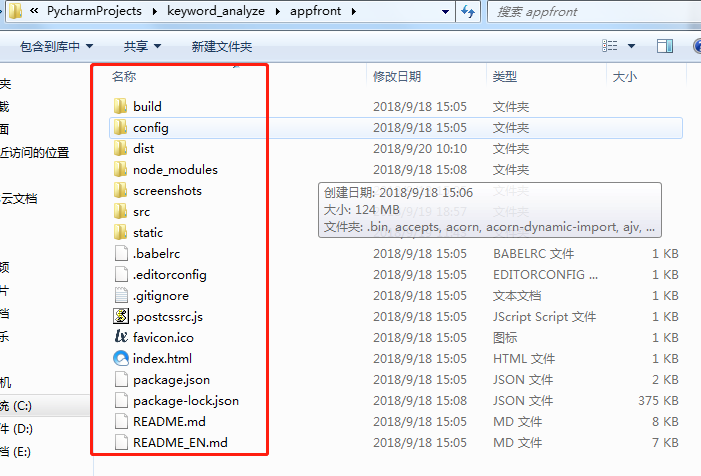
第一种：

vue-init webpack appfront //这里建立一个appfront的前端工程，安装过程会有提示，可以参考：<https://blog.csdn.net/zhuming3834/article/details/72778147>

进入appfront目录，运行命令：

npm install //安装vue所须要的node依赖

安装完成可以看到整个目录：



前端工程创建完成，如果需要集成其他依赖，如：[element-ui](http://element-cn.eleme.io/" \l "/zh-CN/component/quickstart" \t "https://www.jianshu.com/p/_blank)组件（使用其组件美化界面）、[axios](https://www.kancloud.cn/yunye/axios/234845" \t "https://www.jianshu.com/p/_blank) 使用其完成前端到后端的请求等，执行以下命令：

npm install element-ui --save

npm install axios --save

第二种：

也是本次讲解项目采用的方式，使用别人写好模板的前端工程，github链接：<https://github.com/lin-xin/vue-manage-system>

将下载的项目全部源码文件拷贝到建好的appfront前端工程文件夹中（如果项目没有element-ui、axios依赖，如第一种方式一样执行命令）

1. cmd模式进入appfront根目录执行npm install命令，安装vue所须要的node依赖

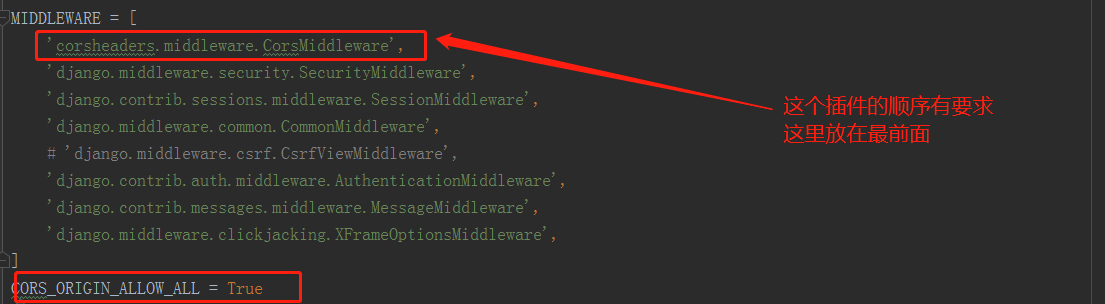
b、npm run dev启动服务，浏览器访问 <https://localhost:8080/>

6.4、前端工程请求后台解决跨域问题

当前端页面请求后台时，会提示请求不了，这就需要解决两个不同服务器之间的跨域问题

1. 进到对应的虚拟环境
2. 安装依赖库，执行命令pip install django-cors-headers

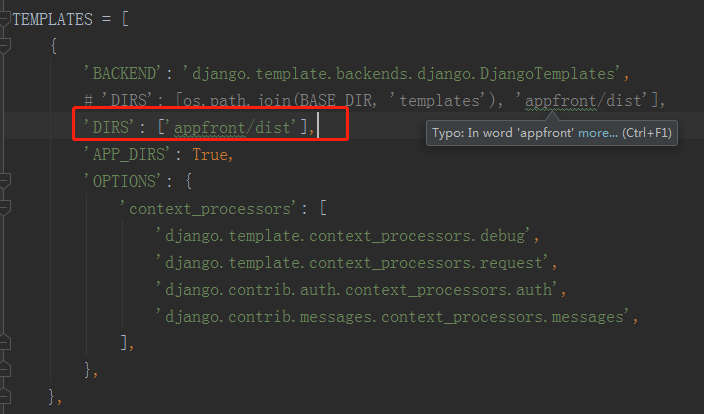
c、在setting.py文件配置，如下：



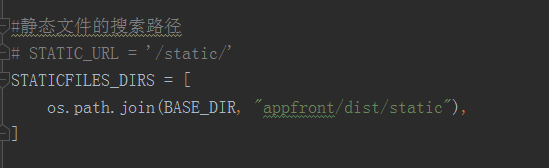
6.5、前端工程与后台工程运行在各自的服务器上，现在整合到后台服务器上

a、在前端工程目录下，输入npm run build，如果项目没有错误的话，就能够看到所有的组件、css、图片等都被webpack自动打包到dist目录

b、需要配置一下模板使Django知道从哪里找到



c、需要配置一下静态文件的搜索路径

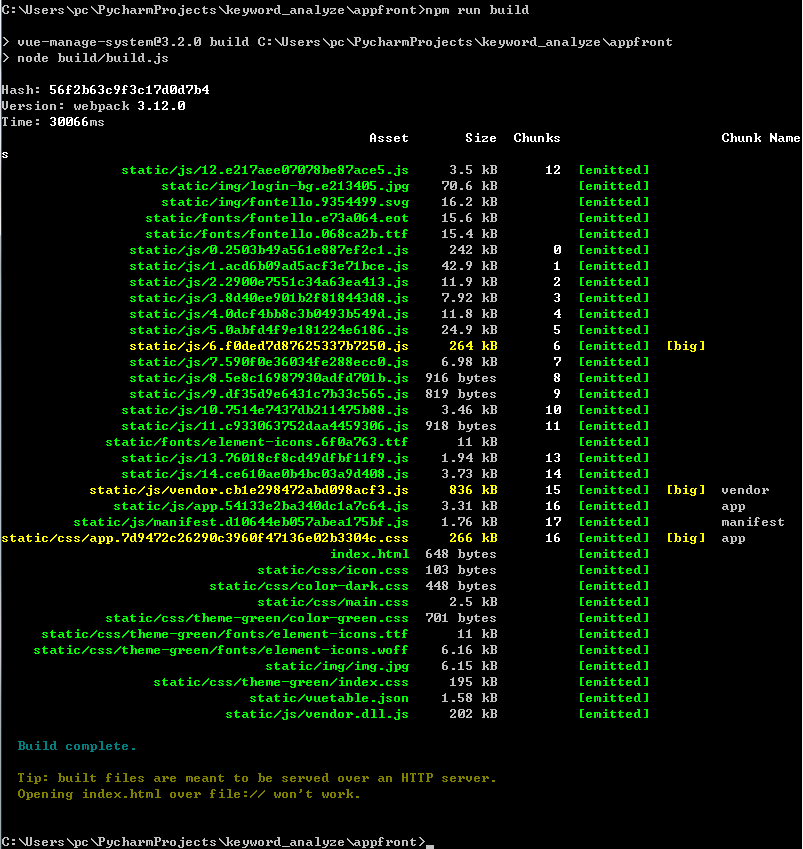


d、访问 『/』时直接返回 index.html

### 

### 6.5、打包前端工程

在前端工程目录下，输入npm run build，如果项目没有错误的话，就能够看到所有的组件、css、图片等都被webpack自动打包到dist目录，如图：



### 这是只要启动后台工程服务即可，不需要启动前端工程

浏览器访问路径：[https://127.0.0.1:8000/](http://127.0.0.1:8000/)

**7、drf(django rest framework)**

项目和文档都会打包放到码云

项目依赖库

