# [在PyCharm导入Anaconda](https://blog.csdn.net/qq_18424081/article/details/85856713" \l "t3" \t "_self)

参考博客：<https://blog.csdn.net/qq_18424081/article/details/85856713>

# 查看系统的环境变量

我的电脑———系统属性———高级系统设置——环境变量

解决python not found 问题

打开管理应用执行别名，关闭安装和下载程序

# Mysqlclient库安装

找到pycharm ——settings ——project——python interpreter—— +

——搜索——install

# 查找python支持的版本

Cmd中直接输入一下两句命令

import pip.\_internal

print(pip.\_internal.pep425tags.get\_supported())

# 解决 query = query.decode(errors=‘replace‘) AttributeError: ‘str‘ object has no attribute ‘decode‘

找到：query = query.decode(errors='replace')

替换为：query = errors='replace'

# Django在terminal命令行创建超级用户时入到password时输入为什么没有反应？

原文链接：https://blog.csdn.net/xuantaiyang888/article/details/81144039

其实，输入的时候并不是没有反应，只是你输入的时候命令行没有将你的输入显示出来，关键是输入行对Password：\*\*\*\*\*\*\*\*也不是采用这种方式隐藏，而是完全的空白。所以即使按键盘感觉没有反应你也要先输入password，再按回车这时会出现password（again），如果出现Error: Your passwords didn't match.说明两次输入的密码不同，如出现Superuser created successfully.则说明已经success.

# Pycharm里面的切换django版本

Settings——project——python interpreter——+搜索——下载对应版本

快捷键 ：按ctrl+/即可快速注释代码块

# ModuleNotFoundError: No module named ‘views‘

博文：https://blog.csdn.net/weixin\_43956958/article/details/114902495

**解决办法：import view 改成 from . import view**

# NameError: name 'os' is not defined

https://learndjango.com/tutorials/nameerror-name-os-not-defined#:~:text=NameError%3A%20name%20%27os%27%20is%20not%20defined.%20Starting%20with,import%20os%20%23%20new%20from%20pathlib%20import%20Path.

The quick fix is to import os at the top of your settings.py file:

*# settings.py*

import os *# new*

from pathlib import Path

# [WinError 10013]以一种访问权限不允许的方式做了一个访问套接字的尝试

<https://blog.csdn.net/peng_for_csdn/article/details/90515042>

1：进入windows中的命令行窗口（win+R之后输入cmd就可以进去）

2.输入 netstat -ano|findstr 8000(一般而言默认启动django项目是8000端口）

3：使用taskkill /pid 进程号 /F

4.这时候我们的项目就可以正常使用pycharm启动了。

或者：**问题解决1：默认启动的服务端口是8000，启动时修改该端口即可**

# 先用npm安装vue-cli脚手架工具

<https://blog.csdn.net/weixin_43844418/article/details/114436815>

# A server erroroccurred. Please contact the administrator

本地化的时候，字母大小写没注意，正确的是LANGUAGE\_CODE = 'zh-Hans'和TIME\_ZONE = 'Asia/Shanghai'，往上再分析是setting文件里的内容被改之后无法被解释器识别，所以以后碰见同类问题可以按照这个思路分析（然后如果再从新创建一个项目试试的话，如果还是报原来的错误，可能是之前那个项目没关）

原文链接：<https://blog.csdn.net/yitiaodashu/article/details/79183689>

# From models import Book ModuleNotFoundError: No module named 'models'解决

*#from models import Book*from .models import \*

# 解决These dependencies were not found:\* vue-resource in ./src/main.js

原因  文件中引入了element-ui/lib/theme-default/index.css （这是旧版本的样式文件引入）

新版本文件引入

import 'element-ui/lib/theme-chalk/index.css'

就可以启动了

# 解决This dependency was not found:\* vue-resource in ./src/main.js

按照提示安装要求的包

我们可以看出在./src/main.js文件里面导入的vue-resource模块不存在，在node\_modules文件夹里也找不到该模块，所以运行项目前，我们要先导入vue-resource模块，根据error提示可知，可使用命令：npm install --save vue-resource, 然后再输入npm run dev 运行项目。

原文链接：<https://blog.csdn.net/weixin_43978056/article/details/105922865>

# Vue 项目提示:`(Emitted value instead of an instance of Error) the “scope“ attribute for scoped slots

scope 属性在2.5以后的版本中已经废弃， 被 slot-scope 替代

办法解决:

检查下你的列表组件里，slot 里的scope 改成slot-scope

Eg：template slot-scope="scope"

# nvm的下载，安装与使用以及快捷键

博文：<https://blog.csdn.net/qq_44401643/article/details/90400626>

https://blog.csdn.net/weixin\_44582077/article/details/110237056

查看node的安装路径 where node

Invalid list option.

Please use on of the following

- nvm list

- nvm list installed

- nvm list available

**windows中如何将已安装的nodejs高版本降级为低版本**

参考连接：<https://blog.csdn.net/weixin_44582077/article/details/110237056>

安装了对应版本：**nvm ls**（用于显示安装了哪些版本）

输入命令行**nvm install node**版本号(例如：nvm install 12.17.0)即可安装对应版本以及自动安装对应的npm版本。

## [Windows上安装NVM之后找不到NPM的解决办法](https://www.cnblogs.com/jaxu/p/13904018.html)

如果安装的对应版本下的nvm文件夹为空，要到（如<https://nodejs.org/download/release/v11.15.0/>是Node.js v11.15.0版本的下载地址）下载对应的压缩包解压后添加到相应的nvm文件夹

参考博文：<https://www.cnblogs.com/jaxu/p/13904018.html>

# Pycharm中配置nodejs

Setting——plugins——搜索nodejs下载——language&framework——

nodejs and NPM——选择本地安装路径

https://blog.csdn.net/lylg\_ban/article/details/113843964

# vue init webpack my-project 选项详解建立vue过程与解释

## 对应解释

<https://blog.csdn.net/github_38336924/article/details/100020408>

Project name (my-project): #项目名称，如果不需要就直接回车。注：此处项目名不能使用大写。

Project description (A Vue.js project): #项目描述，如果不需要就直接回车。

Author (xxx)：#项目作者，默认计算机用户名

vue build (Use arrow keys)

> Runtime + Compiler:recommended for most users #译：运行+编译：被推荐给大多数用户

> Runtime-only:about 6KB lighter min+gzip,but templates (or any Vue-specific HTML) are ONLY

allowed in .vue files-render functions are required elsewhere #译：只运行大约6KB比较轻量的压缩文件，但只允许模板（或任何VUE特定HTML，VUE文件需要在其他地方呈现函数。翻译不精准，意思大概是选择该构建方式对文件大小有要求

install vue-router? #安装vue的路由插件，需要就选y，否则就n [建议Y]

Use ESLint to lint your code? #是否使用ESLint检测你的代码？[ESLint 是一个语法规则和代码风格的检查工具，可以用来保证写出语法正确、风格统一的代码。建议选择 ‘N’ 因为选择 ‘Y’ 在做调试项目时,控制台会有很多 黄色警告 提示格式不规范,但其实并不影响项目]

Setup unit tests? #是否安装单元测试(暂不详细介绍)[建议N]

Setup e2e tests with Nightwatch(Y/n)? #是否安装E2E测试框架NightWatch（E2E，也就是End To End，就是所谓的“用户真实场景” [建议N]

Should we run 'npm install' for you after the project has been created? #项目创建后是否要为你运行“npm install”?这里选择包管理工具 [建议yes,use npm]

yes,use npm #使用npm

yes,use yarn #使用yarn

no,I will handle that myself #自己操作

## 例子

Target directory exists. Continue? Yes

? Project name**（ appfront）**

? Project description A Vue.js project**（回车）**

? Author**（回车）**

? Vue build standalone**（回车）**

? Install vue-router? **（Yes）**

? Use ESLint to lint your code? **（No）**

? Set up unit tests **（No）**

? Setup e2e tests with Nightwatch?**（ No）**

? Should we run `npm install` for you after the project has been created? (recommended) **（npm）**

## 目录文件介绍：

bulid

里面是一些操作文件，使用npm run \* 时其实执行的就是这里的文件。

config

配置文件，执行文件需要的配置信息。

src

资源文件，所有的组件以及所用的图片都是在这个文件夹下放着。简单看一下这个文件夹下都放了哪些东西。

assets

资源文件夹，放图片之类的资源，

components

组件文件夹，写的所有组件都放在这个文件夹下，

router

路由文件夹，这个决定了页面的跳转规则，

App.vue

应用组件，所有自己写的组件，都是在这个组件之上运行了。

main.js

webpack入口文件

原文链接：https://blog.csdn.net/lizy928/article/details/81464244

# 修改pycharm的运行内存避免循环运行python导致内存不足

编辑PyCharm安装目录下PyCharm 4.5.3\bin下的pycharm.exe.vmoptions文件，

-Xms 和 -Xmx是java 命令的一个选项，用来设置你的应用程序启动时的可用内存大小和运行时的可用的内存大小。   
修改   
-Xms为 256m   
-Xmx为 2048m

# { parser: "babylon" } is deprecated; we now treat it as { parser: "babel" }.

## 治标的解决方法：

找到你的工程文件夹里的 YourProName\node\_modules\[vue](https://so.csdn.net/so/search?from=pc_blog_highlight&q=vue)-loader\lib\template-compiler\index.js

//将以下代码

if (!isProduction) {

code = prettier.format(code, { semi: false, parser: 'babylon' })

}

//修改为：

if (!isProduction) {

code = prettier.format(code, { semi: false, parser: 'babel' })

}

原文链接：<https://blog.csdn.net/u011280778/article/details/88107472>

## 治本的解决方法：

{ parser: "babylon" } is deprecated; we now treat it as { parser: "babel" } 是prettier版本导致的，

直接在项目中： cnpm install prettier@~1.12.0 -D 或者cnpm install prettier@~1.12.0 --save-dev 然后重新npm run dev

原文链接：<https://blog.csdn.net/u011280778/article/details/88107472>

# DeprecationWarning: Access to process.binding('http\_parser') is deprecated.解决方案

**不要下载最新版本的node**

Version compatibility could be the issue. Downgrading node js version from 16.8.0 to 14.17.5 resolved my issue.

14.17.5下载路径:

<https://nodejs.org/download/release/v14.17.5/>

[Windows上安装NVM之后找不到NPM的解决办法](https://www.cnblogs.com/jaxu/p/13904018.html)，安装后node\_module显示为空处理方法：

到官网下载对应版本的安装包

将安装包的内容复制到对应的nodejs的安装文件夹里面

<https://www.cnblogs.com/jaxu/p/13904018.html>

# 浏览器出现TemplateDoesNotExist at /

**index.html，终端出现django.template.exceptions.TemplateDoesNotExist: index.html问题**

postman接口测试和参考文档出现相同结果，先忽视

raise TemplateDoesNotExist(template\_name, chain=chain)

raise TemplateDoesNotExist(**', '**.join(template\_name\_list), chain=chain)

# webpack-dev-server --inline --progress --config build/webpack.dev.conf.js 95% emitting DONE Compiled successfully in 无法解决

# 打包时出现警告运行npm run build

可以忽略不影响打包，可以通过降低node版本来解决（步骤在前面）

# Node\_module重建的方法：

Cd 到appfront 然后输入npm install [webpack](https://so.csdn.net/so/search?from=pc_blog_highlight&q=webpack)-dev-server --save-dev

# 中英文设置

LANGUAGE\_CODE='zh-hans'  
TIME\_ZONE = 'Asia/Shanghai'

解决WARNINGS:

api\_test.Book: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.

HINT: Configure the DEFAULT\_AUTO\_FIELD setting or the ApiTestConfig.default\_auto\_field attribute to point to a subclass of AutoField, e.g.

'django.db.models.BigAutoField'.

打开项目的settings.py并在文件底部添加新行

DEFAULT\_AUTO\_FIELD = 'django.db.models.AutoField'

# Django操作流程

<https://cloud.tencent.com/developer/article/1576599>

1）在cmd中输入 cd到指定文件夹 然后输入**python -m venv venv**（创建的目标目录不一样）

2）cd到对应的 script文件夹输入activate.bat即可激活环境

3）cd到ven安装django（**pip install Django**）

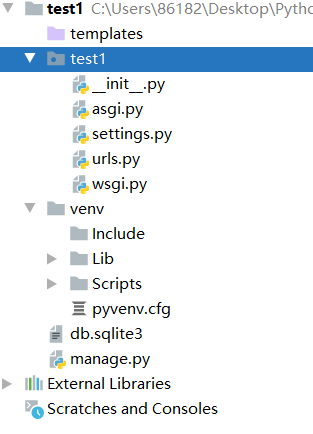
4）检查版本信息 输入python

然后输入import django

print(django.get\_version())

通过输入**quit()**退出python查看版本

4）文件夹建立好后的文件目录

各个同级文件夹

4）cd到manage.py对应的文件夹进行数据的迁徙（**python manage.py migrate**）数据建立到Django默认数据库为db.sqlite3

5）启动Django Server ，验证默认配置是否正常。（**python manage.py runserver 0.0.0.0:8000**）

6）接着如果出现网址为**http://0.0.0.0:8000/** 的页面可能存在问题，或者已永久移动到新的网址。

ERR\_ADDRESS\_INVALID

解决办法需要在浏览器上访问的网址是<http://localhost:8000/>并不是terminal上显示的网址

二、django的默认数据库转化为mysql

1）cmd 打开并进入数据库MySQL创建数据库的名字

mysql> CREATE DATABASE django\_vue\_db CHARACTER SET utf8;

2）到pycharm terminal中安装mysqlclient（**pip install mysqlclient**）

3）配置settings.py文件，配置Mysql数据库引擎。

DATABASES = {  
 **'default'**: {  
 **'ENGINE'**: **'django.db.backends.mysql'**,  
 **'NAME'**: **'django\_vue\_db'**,  
 **'USER'**: **'root'**,  
 **'PASSWORD'**: **'yyf314989'**,  
 **'HOST'**: **'127.0.0.1'**,  
 }  
}

4）执行同步操作，将数据迁移到Mysql。（**python manage.py migrate**）

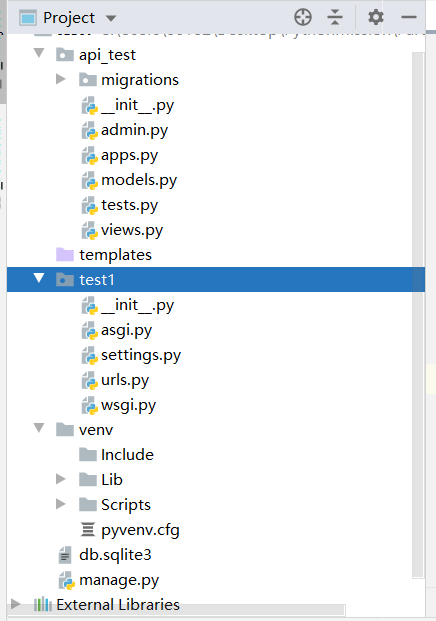
可以查看MySQL中是否创建好了对应的表格

5）运行Django Server，重新访问http://localhost:8000。在terminal中运行（**python manage.py runserver 0.0.0.0:8000）**

出现相同与6）的界面表示数据库切换成功

三、**创建Django实战项目App**

**1）terminal 中cd到项目的根目录创建（manage.py所在目录）（python manage.py startapp api\_test）**

 对应的目录列表

2）把创建的api\_test加入到test1/settings文件中的installed\_apps列表里：

3) 在api\_test目录下的models.py里我们简单写一个model如下:

(代码在对应文件下)

只有两个字段，书名book\_name和添加时间add\_time。如果没有指定主键的话Django会自动新增一个自增id作为主键。

4)在api\_test目录下的views里我们新增两个接口，一个是show\_books返回所有的书籍列表（通过JsonResponse返回能被前端识别的json格式数据），二是add\_book接受一个get请求，往数据库里添加一条book数据。

（代码在对应文件）

5）在api\_test目录下，新增一个urls.py文件，把我们新增的两个接口添加到路由里：

**文件可以通过在别的文件夹下复制然后更改内容**

6）我们还要把api\_test下的urls添加到项目test1下的urls中，才能完成路由：

7）在项目的根目录，输入命令：(**python manage.py makemigrations api\_test**

**)**出现下面错误

from \_\_future\_\_ import unicode\_literals

^

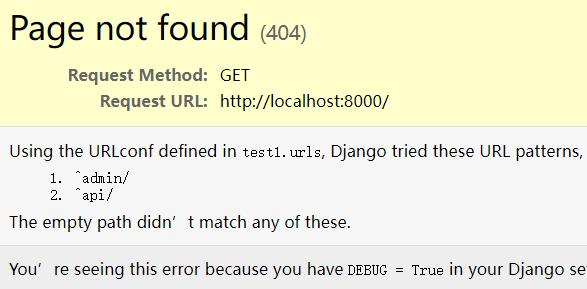
SyntaxError: from \_\_future\_\_ imports must occur at the beginning of the file

按照提示将models对应文件的语句放到该文件开头位置即可。

接着输入命令（**python manage.py migrate**）

8)查询数据库发现book表已经自动创建了：在该数据库下接着（**desc api\_test\_book）**;看出表格的结构

9）在项目的根目录，输入命令：(**python manage.py runserver 0.0.0.0:8000）**

**出现****界面把test1/文件夹下的urls的admin和api删除全部做空**

urlpatterns = [  
 url(**r'^'**, admin.site.urls),  
 url(**r'^'**, include(api\_test.urls)),  
]

在浏览器中出现django登录则在manage.py文件目录下输入

# python manage.py createsuperuser #输入

Username (leave blank to use 'root'): admin #输入用户名

Email address: admin@runoob.com #输入邮箱

Password: #输入新建的用户密码

Password (again): #再次输入

Superuser created successfully. #出现这个，表明创建成功

[root@solar HelloWorld]#

**10）**通过httpie测试接口，首先cmd pip install httpie 接

接着在cmd中输入对应端口号：

http <http://127.0.0.1:8000/api/add_book\?book_name\=mikezhou_talk>

显示不一样（与参考文档）

Postman显示一串乱码是因为没有把***# -\*- coding: utf-8 -\*-这段话加到views文件中***

11）出现Error at /Invalid base64-encoded string: number of data characters (217) cannot be 1 more than a multiple of 4

**四． 新建Vue.js前端项目**

1）在更目录下（manage.py）输入 **npm install -g vue-cli**

2）输入**vue-init webpack frontend选项按照前面的**

Vue建立选项

**3）**安装 vue 依赖模块

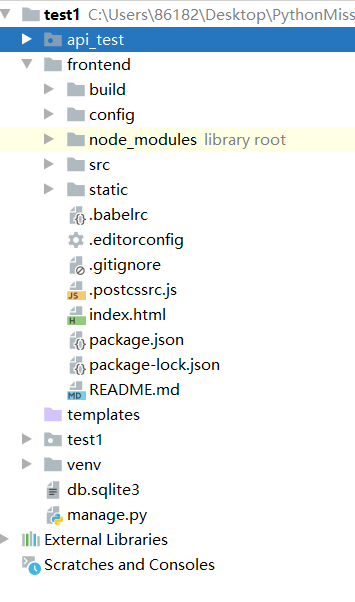
cd frontend

npm install

npm install vue-resource

npm install element-ui

4）现在我们可以看到整个文件目录结构是这样的：



5）在src/component文件夹下新建一个名为Home.vue的组件，通过调用之前在Django上写好的api，实现添加书籍和展示书籍信息的功能。

New ——vue组件

代码在bloc

6）在src/router目录的index.js中，我们把新建的Home组件，配置到vue-router路由中：

（代码）

7）、在src/main.js文件中，导入element-ui、vue-resource库。

（代码）

8）如果出现跨域问题，需要在Django层注入header，用Django的第三方包django-cors-headers来解决跨域问题：

直接安装**：（npm install element-ui）**

settings.py 修改：

9）、在前端工程frontend目录下，输入npm run dev启动node自带的服务器，浏览器会自动打开， 我们能看到页面：

**出现{ parser: "babylon" } is deprecated;**