## ORM

python manage.py makemigrations appname

python manage.py migrate

ORM:

objeccts relation mapping

（不设置主键 会默认使用一个）

gid = models.AutoField(primary\_key=True)

max\_length=240

default="商品描述,商品描述"

unique=True

create\_time = models.DateTimeField(auto\_now\_add=True)

One ToOne:

OneToMany:

goods = models.ForeignKey(Goods,  
 related\_name='comments',  
 on\_delete=models.CASCADE) # User 关联到用户  
parent\_comment = models.ForeignKey('self',  
 related\_name='children',  
 on\_delete=models.CASCADE,  
 blank=True, null=True)

ManyToMany:

<https://www.cnblogs.com/haozike/p/python_django_ManyToMany.html>

## 序列化器：

覆写 update()和create()方法

# **Django ORM常用操作介绍:**

https://www.django.cn/article/show-15.html

# Django rest:

## 1、数据库模型，数据库字段，数据库外键，relative\_name字段，自关联外键

## 2、序列化器

## 3、url路由，视图集ViewSet

## 4、url请求静态文件配置，处于debug模式下才可使用：

# 例如一个请求 http://localhost:8080/static/js/xxx.js 匹配到url前缀 xxxx/static/ 取出将剩余/js/xxx.js 到底下STATICFILES\_DIRS设置的目录进行查找  
STATIC\_URL = '/static/'  
  
# Add for vuejs  
# 在生产环境 Debug=True 的情况下才能使用  
# 匹配url前缀/static/ 之后,在fronted/dist/static目录下 查询 /js/xxx.js  
STATICFILES\_DIRS = [  
 os.path.join(BASE\_DIR, 'fronted/dist/static'),  
]

## 5、Csrf问题（跨站请求伪造）

from rest\_framework.authentication import SessionAuthentication

class CsrfExemptSessionAuthentication(SessionAuthentication):  
  
 def enforce\_csrf(self, request):  
 return # To not perform the csrf check previously happening

from rest\_framework.viewsets import ModelViewSet

class GoodsViewSet(mixins.ListModelMixin,  
 mixins.RetrieveModelMixin,  
 viewsets.GenericViewSet):

class UserViewSet(ModelViewSet):

authentication\_classes = (CsrfExemptSessionAuthentication,)

## 6、用户权限认证：

### 1、登录

from django.contrib import auth  
from django.contrib.auth import authenticate, login, logout

### 2、登出

auth.logout(request)

request.session.flush() # 清除缓存 一次性将session的东西都删除

### 3、判断user是否已经登录

request.user.is\_authenticated:

## 7、axios、django跨域请求处理：

### django：

django-cors-headers

(官网：<https://github.com/ottoyiu/django-cors-headers>)

### 1、模块支持： pip install django-cors-headers

### 2、

INSTALLED\_APPS = (

...

'corsheaders',

...

)

### 3、

MIDDLEWARE = [ # Or MIDDLEWARE\_CLASSES on Django < 1.10

...

'corsheaders.middleware.CorsMiddleware',

'django.middleware.common.CommonMiddleware',

...

]

#'corsheaders.middleware.CorsMiddleware'

要在'django.middleware.common.CommonMiddleware'(django原来就自带的)前面，

(因为django MIDDLEWARE 按照配置的顺序执行 中间件，其实把

'corsheaders.middleware.CorsMiddleware' 放到最上层更好,因为其他中间件可能会处理response 导致一些不可预料的后果)

### 4、站点白名单设置，使这些站点能访问server

CORS\_ORIGIN\_WHITELIST = (

'google.com',

'hostname.example.com',

'127.0.0.1:9000',

'localhost:8080',

'localhost:8000/',

'127.0.0.1:8000/',

'127.0.0.1:8080/',

)

### **5、CORS\_ALLOW\_CREDENTIALS**

CORS\_ALLOW\_CREDENTIALS = True # 设置响应头 Access-Control-Allow-Credentials:true

默认为False,设置为True才能允许requests带有cookies

### #**CORS\_ORIGIN\_ALLOW\_ALL = True #允许所有站点访问**

**官方网站还有很多配置属性：允许什么方法、使用正则匹配、允许什么请求头**

## axios:

// 带上cookie请求  
axios.defaults.withCredentials = **true**;

## 8、django的QuerySet update之后查询集就为空了

## 9、父向子组件通信：props:[]

## 子向父组件通信：$emit

## 10、vue cli3 打包后 'render' undefine

vue的组件没有写

**export default** {  
 name:"xxxx"  
}

## 11、vue打包后的问题：

1、打包后图片不显示：

路径问题 如果是,在vue cli3打包的时候图片的路径并没有被改变

1、直接写相对路径引用一个图片：

<div class="demo-carousel">  
 <img src="../assets/汽车8.jpg">  
</div>

打包后编译成：



这种情况vue cli3在打包时会对assets目录下的文件会根据vue.config.js的配置进行处理，将src映射好

2、动态从后端获取img\_url：

<div style="text-align: center;overflow: hidden">  
 <img :src="`${baseUrl}${goods.img\_url}`" />  
 <!--去看vue cli3 官方文档 静态文件部分-->  
</div>

编译后：



而后端对于静态资源的请求标准格式：

/static/img/xxx.jpg、/static/img/xxx.png 所以会获取不到

解决方案：

1、（默认情况 如果没有在vue.config.js里修改配置的话）在pullic下的静态文件vue cli3打包后并不做任何操作，只把其放到dist根目录下

那么在后端添加对/xxx.jpg、/xxx.png 形式的url进行路由解析并返回正确的所请求的资源就可以解决问题

2、在配置axios时使用拦截器,如果是/xxx.jpg、/xxx.png这种格式的url将其处理成/static/img/xxx.jpg、/static/img/xxx.png,

并且手动将图片复制到 /dist/static/img/目录下，这样后端在处理这种url的请求时会在/dist/static/img/ 目录下寻找xxx.jpg、xxx.png

3、直接将数据库里存的图片的img\_url改为：/static/img/xxx.jpg、/static/img/xxx.png,然后把图片复制到，/dist/static/img/ 目录下

4、nginx进行静态资源请求的处理?这个咋做

2、打包后运行不正常( store.js 仓库失效 不知道为什么 获取不到 )

3、打包后刷新页面（会出现404 not found

原因：

直接发送类似http请求给服务器：

<http://localhost:8080/search-goods>

因为是用vue做的页面路由，后端并不能解析这个http请求

vue单页的router是没有请求http而是在一页上操作，虽然上边网址栏的http地址会改变，但实际上却没有向服务器进行http请求

解决方案：？

## 11、后端每个Response都要写返回状态码，否则前端无法知道此次请求是成功还是失败

## 12、静态资源最好是英文名的，中文可能前后端编码不一致的话会产生乱码或者解析不出正确的中文文件名