

Django Admin站点使用

一、创建项目

使用pycharm创建Django项目,使用命令创建应用 `python manage.py startapp goods` (创建应用goods)

二、配置文件

Settings.py文件INSTALLED_APPS中加载应用配置实现注册应用,并配置数据库,使用pymysql驱动,在项目同名文件夹下的__init__.py文件初始化mysql驱动

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'test_2',  
        'HOST': 'localhost',  
        'PORT': 3306,  
        'USER': 'root',  
        'PASSWORD': 'mysql'  
    }  
}
```

配置mysql数据库

```
1 # 初始化mysql驱动  
2 import pymysql  
3  
4 pymysql.install_as_MySQLdb()  
5
```

初始化mysql驱动

三、goods应用定义模型



```
class Shop(models.Model):
    """ 商铺模型 """
    name = models.CharField(max_length=20)
    location = models.CharField(max_length=20)
    create_date = models.DateField()
    is_delete = models.BooleanField(default=False)

    class Meta:
        db_table = 'shop'

    def __str__(self):
        return self.name

class Goods(models.Model):
    """ 商品模型表 """
    choices_gender = (
        (0, '男'),
        (1, '女')
    )
    name = models.CharField(max_length=20)
    brand_name = models.CharField(max_length=20)
    show_date = models.DateField()
    gender = models.IntegerField(choices=choices_gender, default=0)
    price = models.DecimalField(decimal_places=2, max_digits=7) # 最多7位数, 保留两位小数
    comment = models.CharField(max_length=200, null=True, blank=True)
    shop = models.ForeignKey('Shop', on_delete=models.CASCADE)
    is_show = models.BooleanField(default=True)

    class Meta:
        db_table = 'goods'

    def __str__(self):
        return self.name
```

四、数据库迁移,创建对象

数据库迁移：`python manage.py makemigrations` 生成迁移文件

`python manage.py migrate` 同步到数据库

进入交互环境,创建对象 `python manage.py shell` (导入模型)

启动 python 两种方法,1.pyhton 2.python manage.py shell,都会启动交互解释器,但是第二种启动解释器会告诉 Django 使用哪个配置文件,Django 框架大部分子系统都依赖于配置文件.

```
from goods.models import Shop, Goods
```

```
Shop.objects.create(name='速写专卖店', location='广东深圳', create_date='2018-5-30')  
<Shop: 速写专卖店>
```

```
Goods.objects.create(name='风衣', brand_name='速写', show_date='2018-11-11', gender=1, price=999.99, comment='抵挡寒风', shop=(Shop.objects.get(id=1)))  
<Goods: 风衣>
```

五、创建超级管理员

```
Terminal: Local x Local (2) x Local (3) x +
D:\Site>python manage.py createsuperuser
Username: admin
Email address: admin@163.com
Password:
Password (again):
The password is too similar to the email address.
```

六、界面本地化

```
107
108 LANGUAGE_CODE = 'zh-hans'
109
110 TIME_ZONE = 'Asia/Shanghai'
111
```

七、注册模型

在应用goods下的admin.py文件中注册模型,可以实现对数据进行增删改管理操作

```
1 from django.contrib import admin
2 from .models import Shop, Goods
3
4 # Register your models here.
5
6 # 注册模型
7 # admin.site.register([Shop, Goods])
8 admin.site.register(Shop)
9 admin.site.register(Goods)
10
```

导入模型

注册模型

八、站点显示备注信息

为模型字段选项添加verbose_name显示字段备注信息,verbose_name_plural显示表名备注信息

```
6 class Shop(models.Model):
7     """ 商铺模型 """
8     name = models.CharField(max_length=20, verbose_name='店铺名称')
9     location = models.CharField(max_length=20, verbose_name='店铺地址')
10    create_date = models.DateField(verbose_name='开业时间')
11    is_delete = models.BooleanField(default=False, verbose_name='逻辑删除')
12
13    class Meta:
14        db_table = 'shop'
15        verbose_name_plural = '店铺'
16
17    def __str__(self):
18        return self.name
```

字段备注信息

表名备注信息,表名带复数形式

首页 > Goods > 商品 > 裙子

修改 goods

显示字段的备注名

商品名称:

裙子

品牌:

速写

上市日期:

2018/06/04

今天 |

群体:

女 ▼

价格:

388.00

商品描述:

店铺:

速写专卖店 ▼

☒ 是否展示

修改应用名:在应用下的 app.py 文件下添加 **verbose_name**



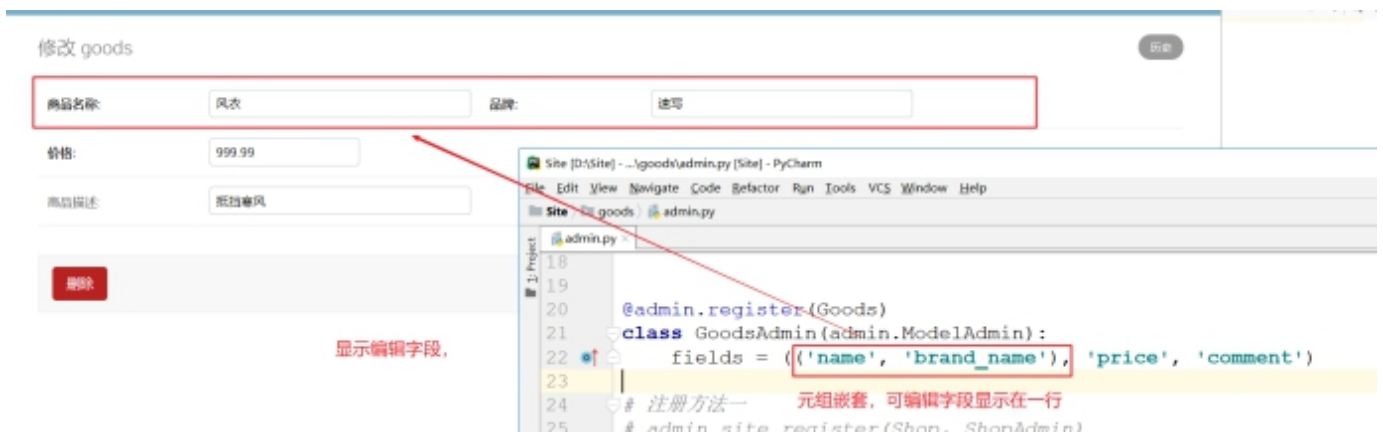
九、定义与使用admin管理类

ModelAdmin类控制admin站点的展示效果,我们可以自定义管理类,继承ModelAdmin

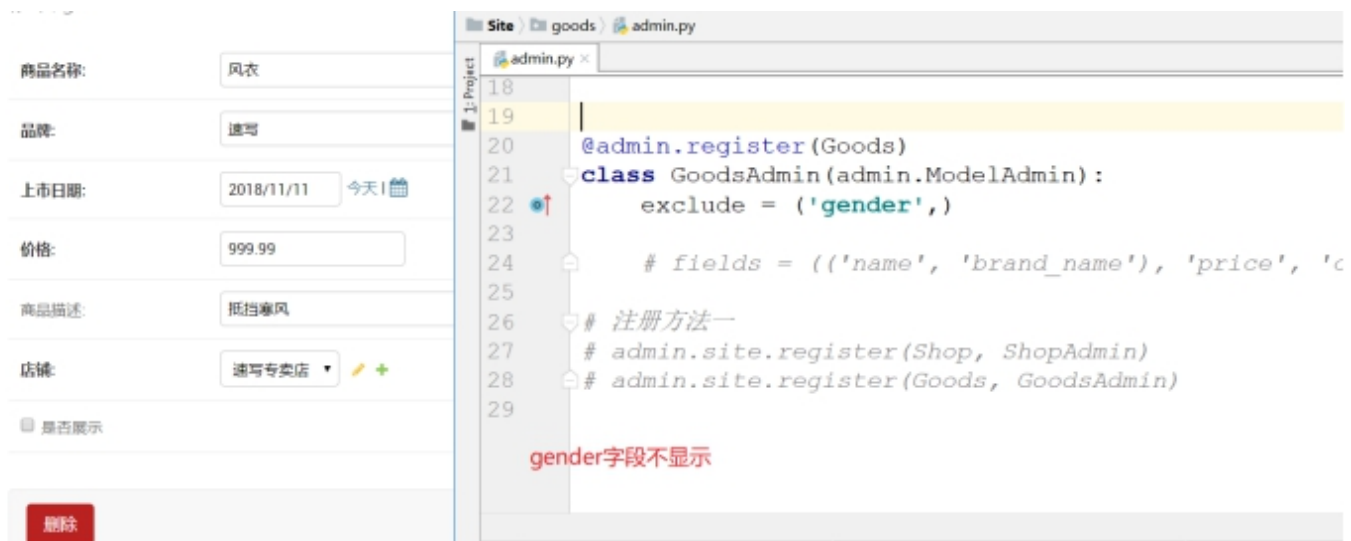
```
12 # 自定义站点管理类
13
14 # 注册方法二
15 @admin.register(Shop)
16 class ShopAdmin(admin.ModelAdmin):
17     pass
18
19 # 两种注册方法
20 @admin.register(Goods)
21 class GoodsAdmin(admin.ModelAdmin):
22     pass
23
24 # 注册方法一
25 # admin.site.register(Shop, ShopAdmin)
26 # admin.site.register(Goods, GoodsAdmin)
27
```

十、自定义编辑页面显示

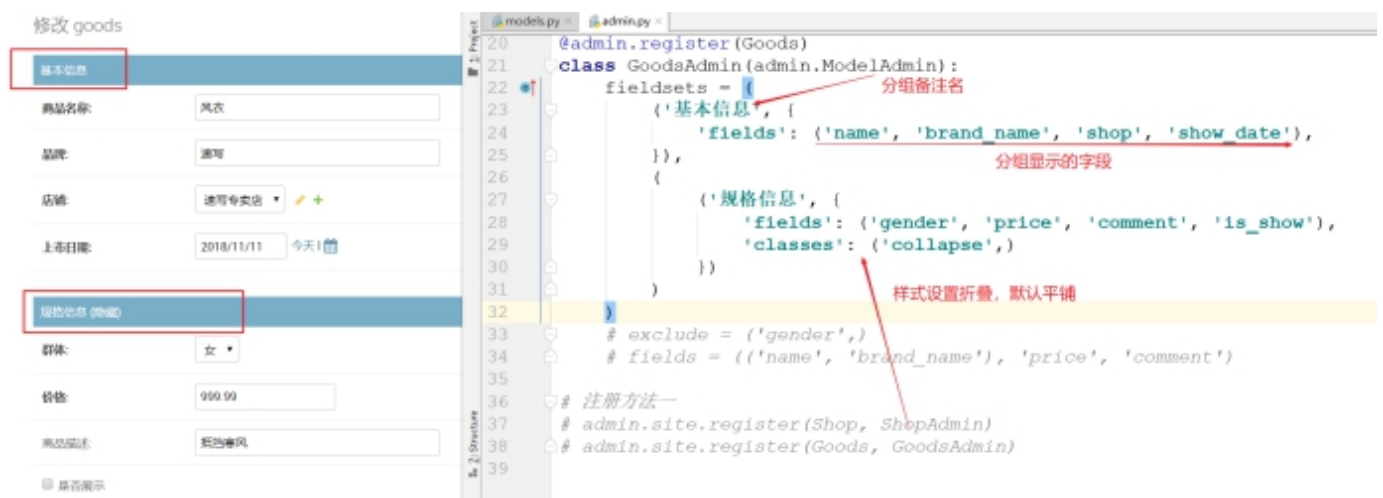
fields:编辑页面显示可编辑字段,如果嵌套,会在同一行显示多个字段



exclude: 编辑页面不显示可编辑字段



fieldsets: 编辑页面对编辑字段进行分组



fields 和 fieldsets 只能两者选之一，

如果字段为日期或时间类型,并选项auto_now或auto_now_add设置为True,那么这个字段是不可编辑的,默认不会显示,也不能手动添加

十一、InlineModelAdmin内联(关联对象)

在管理界面能够在与父模型相同的页面上编辑模型,在一对多的关系中,我们可以在一的一方编辑多的对象.

两个子类:**TabularInline** 以表格的形式嵌入
StackedInline 以块状形式嵌入

这两种的方法仅仅只是用于渲染他们的模板

```
12 # 自定义站点管理类
13 class GoodsTabularInline(admin.TabularInline):
14     # 关联模型
15     model = Goods
16
17
18 class GoodsStackedInline(admin.StackedInline):
19     # 关联模型
20     model = Goods
21
22
23 # 注册方法二
24 @admin.register(Shop)
25 class ShopAdmin(admin.ModelAdmin):
26     inlines = [
27         # GoodsStackedInline # 以块状形式嵌入
28         GoodsTabularInline # 以表格形式嵌入
29     ]
30     pass
31
```



商品名称	品牌	上市日期	群体	价格	商品描述
风衣	速写	2018/11/11 今天!	女	999.99	抵挡寒风
裙子	速写	2018/06/04 今天!	女	388.00	
			男		

十二、自定义列表页显示

`list_display` : 定义列表页显示的列, 未定义 `list_display` 默认显示的是定义模型的 `__str__` 方法返回的对象字符串

```
@admin.register(Goods)
class GoodsAdmin(admin.ModelAdmin):
    list_display = ('name', 'brand_name', 'show_date', 'gender',
                   'price', 'shop', 'comment', 'is_show')
```

动作: 执行 2 个中 0 个被选

商品名称	品牌	上市日期	群体	价格	店铺	商品描述	是否展示
裙子	速写	2018年6月4日	女	388.00	速写专卖店	-	●
风衣	速写	2018年11月11日	女	999.99	速写专卖店	抵挡寒风	●

2 商品

显示自定义的字段

bool类型字段显示一个开关

外键显示关联对象的 `__str__` 返回的字符串

如果字段是 `ForeignKey`, Django 将显示 `__str__` 相

关对象(在我们的Goods 模型 中,shop是Goods模型的外键,那么shop字段显示的就是我们Shop模型中的__str__()对象)

如果字段是 BooleanFiled,Django 将显示一个开关,而不是 True 或 False

如果字段的值为 None,Django 默认显示 - 破折号, `empty_value_display` 属性设置值为 None 的显示

商品名称	品牌	上市日期	群体	价格	店铺	商品描述	是否展示
裙子	速写	2018年6月4日	女	388.00	速写专卖店	未填	✔
风衣	速写	2018年11月11日	女	999.99	速写专卖店	抵御寒风	❌

2 商品

设置值为None的显示

```
class GoodsAdmin(admin.ModelAdmin):
    list_display = ('name', 'brand_name', 'show_date', 'gender',
                    'price', 'shop', 'comment', 'is_show')
    empty_value_display = '未填'
```

`list_display_links`:设置进入编辑页面的链接,默认第一列链接进入到编辑页面,将其设置为 None 完全没有链接,将其设置为 `list_display` 要转换为链接的列的列表或元组, `list_display_links` 必须和

list_display 配合使用

```
@admin.register(Goods)
class GoodsAdmin(admin.ModelAdmin):
    list_display = ('name', 'brand_name', 'show_date', 'gender',
                   'price', 'shop', 'comment', 'is_show')
    empty_value_display = '未填'
    list_display_links = ('name', 'brand_name')
    fieldsets = (
        ('基本信息', {
            'fields': ('name', 'brand_name', 'shop', 'show_date'),
```

动作: 执行 2个中0个被选

<input type="checkbox"/>	商品名称	品牌	上市日期	性别	价格	店铺	商品描述	是否展示
<input type="checkbox"/>	裙子	速写	2018年6月4日	女	388.00	速写专卖店	未填	<input checked="" type="checkbox"/>
<input type="checkbox"/>	风衣	速写	2018年11月11日	女	999.99	速写专卖店	抵挡寒风	<input type="checkbox"/>

2 商品

有链接进入到编辑页面

list_editable:允许字段在列表页进行编辑,必须和list_display配合使用,无法编辑未显示的字段, list_editable 列表字段,不能和list_display_links 字段名冲突,不能有进入到编辑页面的链接.

```
@admin.register(Goods)
class GoodsAdmin(admin.ModelAdmin):
    list_display = ('name', 'brand_name', 'show_date', 'gender',
                   'price', 'shop', 'comment', 'is_show')
    empty_value_display = '未填'
    list_display_links = ('name', 'brand_name')
    list_editable = ('price', 'comment', 'is_show')
    fieldsets = (
        ('基本信息', {
```

<input type="checkbox"/>	商品名称	品牌	上市日期	群体	价格	店铺	商品描述	是否展示
<input type="checkbox"/>	裙子	速写	2018年6月4日	女	388.00	速写专卖店		<input checked="" type="checkbox"/>
<input type="checkbox"/>	风衣	速写	2018年11月11日	女	999.99	速写专卖店	抵挡寒风	<input type="checkbox"/>

2 商品

可在列表页面进行编辑

保存

list_filter 列表页显示右侧的过滤器, 关联对象过滤: 外键名__关联对象字段名

```
list_editable = ('price', 'comment', 'is_show')
list_filter = ('name', 'gender', 'shop_name')
fieldsets = (
    ('基本信息', {
        'fields': ('name', 'brand_name', 'shop', 'show_date')
    }),
    (
```

显示过滤器, 指定过滤的字段, 外键过滤: 外键_字段

选择 goods 来修改

动作 执行 2 个中 0 个被选

<input type="checkbox"/>	商品名称	品牌	上市日期	群体	价格	店铺	商品描述	是否展示
<input type="checkbox"/>	裙子	速写	2018年6月4日	女	388.00	速写专卖店		<input checked="" type="checkbox"/>
<input type="checkbox"/>	风衣	速写	2018年11月11日	女	999.99	速写专卖店	抵挡寒风	<input type="checkbox"/>

2 商品

保存

过滤器

以 商品名称

全部
裙子
风衣

以 群体

全部
男
女

以 店铺名称

全部
速写专卖店

list_per_page: 控制列表页显示的条数, 默认100

list_max_show_all: 控制 ' 显示全部 ' 的列表页的

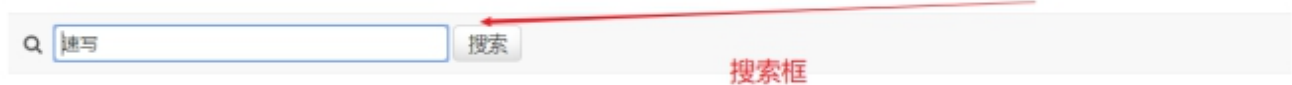
显示条数,默认 200 条

search_fields : 在列表页启用搜索框

search_fields 字段名列表,

关联对象外键查询 search_fields=['外键名_字段名']

选择 goods 来修改



```
list_editable = ('price', 'comment', 'is_show')
list_filter = ('name', 'gender', 'shop_name')
search_fields = ('name', 'shop_name')  # 关联对象外键查询
fieldsets = (
    ('基本信息', {  # 定义查询的字段
        'fields': ('name', 'brand_name', 'shop', 'show_da
    )},
```

十三、 AdminSite 站点信息

调整 adminsite 基本信息

admin.site.site_title= 站点标题

admin.site.site_header= 站点头

admin.site.index_title= 首页标题

