

关注社区微信公众号: PMvideo

细说Django的admin

作者: renpingsheng (/authorarticle.html?author=renpingsheng) 2017-10-19 ☆ 收录到我的专题 (/select.html?articleId=536991)

标签 效果 (<http://www.bijishequ.com/info/search.html?searchText=效果>) 页面 (<http://www.bijishequ.com/info/search.html?searchText=页面>) 显示 (<http://www.bijishequ.com/info/search.html?searchText=显示>) 列表 (<http://www.bijishequ.com/info/search.html?searchText=列表>) 详细 (<http://www.bijishequ.com/info/search.html?searchText=详细>)

在 `admin.py` 中只需要将地 `Model` 中某个类注册,即可在Admin中进行增删查改的功能,例如:

```
admin.site.register(models.UserInfo)
```

这种方式比较简单,如果想要进行更多的定制操作,就需要利用 `ModelAdmin` 进行操作

方式一

```
from django.contrib import admin
from . import models

class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')

admin.site.register(models.Book, BookAdmin)
```

没有设置 `BookAdmin` 时,系统默认为 `ModelAdmin`

`models.Book` 实际上是一个列表,也可以写成

```
from django.contrib import admin
from . import models

class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')

admin.site.register([models.Book,], BookAdmin)
```

这个参数里可以写多个表名,但要注意的是这里面添加的表名都必须要有 `list_display` 中声明要显示的字段

方式二

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')
```

在 `ModelAdmin` 中提供了大量的可定制的功能

定制admin

1.`list_display`:显示列表时,定制显示的列

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')
```

在后台管理页面中,显示书的名称,价格以及出版社等信息

Select book to change

Action: 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH
<input type="checkbox"/>	php	66.00	深圳出版社
<input type="checkbox"/>	golang	78.00	工业出版社
<input type="checkbox"/>	bootstrap	67.00	湖北出版社
<input type="checkbox"/>	saltstack	105.00	人民出版社

在class类中还有如下用法:

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'aaa')

    def aaa(self, obj):
        return obj.title+"--aaaa"
```

显示如下

Select book to change

Action: 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	AAA
<input type="checkbox"/>	php	66.00	php--aaaa
<input type="checkbox"/>	golang	78.00	golang--aaaa
<input type="checkbox"/>	bootstrap	67.00	bootstrap--aaaa

2.list_display_links:列表时,定制列可以点击跳转

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')
    list_display_links = ('title', 'price', 'publish')
```

在后台页面显示的时候,所显示的列为可以点击跳转

Select book to change

Action: 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH
<input type="checkbox"/>	php	66.00	深圳出版社
<input type="checkbox"/>	golang	78.00	工业出版社

3.list_filter:列表时,定制右侧快速筛选

例子一:

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    list_display = ('title', 'price', 'publish')
    list_filter = ('classification', 'publish')
```

效果如下:

ADD BOOK +

FILTER

By classification

All

文学

机械

科学

计算机

By publish

All

人民出版社

工业出版社

重庆出版社

湖北出版社

上海出版社

广州出版社

深圳出版社

例子二:

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')

    class Ugg(admin.SimpleListFilter):

        title=uget('类型')

        parameter_name = "book"

        def lookups(self, request, model_admin):
            """
            显示筛选选项, 列出价格大于100的书籍的类型
            :param request:
            :param model_admin:
            :return:
            """
            return models.Book.objects.filter(price__gt=100).values_list("publish", "classification")

        def queryset(self, request, queryset):
            """
            点击查询时, 进行筛选
            :param request:
            :param queryset:
            :return:
            """
            vl=self.value()
            return queryset

    list_filter = ["publish", Ugg,]
```

效果如下:

Select book to change

ADD BOOK +

Action: Go 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH
<input type="checkbox"/>	php	66.00	深圳出版社
<input type="checkbox"/>	golang	78.00	工业出版社
<input type="checkbox"/>	bootstrap	67.00	湖北出版社
<input type="checkbox"/>	saltstack	105.00	人民出版社
<input type="checkbox"/>	ruby	15.00	工业出版社
<input type="checkbox"/>	markdown	11.00	重庆出版社
<input type="checkbox"/>	jQuery	34.00	重庆出版社
<input type="checkbox"/>	javascript	35.00	上海出版社
<input type="checkbox"/>	c sharp	45.00	人民出版社

FILTER

By publish

All

人民出版社

工业出版社

重庆出版社

湖北出版社

上海出版社

广州出版社

深圳出版社

By 类型

All

机械

文学

机械

4.list_select_related:列表时, 连表查询是否自动select_related

使用联表查询可以提高数据库的查询性能

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')

    list_select_related = ["publish"] # 联表查询出版社的信息
```

5. 分页相关

```
list_per_page=10 # 分页, 每页显示的数据条数
list_max_show_all=100 # 分页, 显示全部数据时, 最多显示的数据条数
paginator=Paginator # 分布插件
```

例如:

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    list_display = ('title', 'price', 'publish')
    list_per_page = 2
```

效果如下:

Select book to change

Action: Go 0 of 2 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH
<input type="checkbox"/>	php	66.00	深圳出版社
<input type="checkbox"/>	golang	78.00	工业出版社

1 2 3 4 5 6 7 8 16 books Show all

6. list_editable:列表时, 可以编辑的列

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish')

    list_editable = ('price', 'publish')
```

效果如下:

Select book to change

Action: 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH
<input type="checkbox"/>	php	<input type="text" value="66.00"/>	<input type="button" value="深圳出版社"/>
<input type="checkbox"/>	golang	<input type="text" value="78.00"/>	<input type="button" value="工业出版社"/>

7. search_fields:列表时，模糊搜索的功能

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title','price','publish')

    list_select_related = ["publish"]

    search_fields = ['title','classification']
```

效果如下:

Select book to change

1 result (16 total)

Action: 0 of 1 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH
<input type="checkbox"/>	golang	78.00	工业出版社

1 book

8. date_hierarchy:列表时，对Date和DateTime类型进行搜索

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title','price','publish','publication_date')

    list_select_related = ["publish"]

    date_hierarchy="publication_date"
```

效果如下:

2016 2017

Action: 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH	PUBLICATION DATE
<input type="checkbox"/>	php	66.00	深圳出版社	June 7, 2017
<input type="checkbox"/>	golang	78.00	工业出版社	May 27, 2017
<input type="checkbox"/>	bootstrap	67.00	湖北出版社	Sept. 5, 2017

< All dates January 2017 February 2017 April 2017 May 2017 June 2017 July 2017 August 2017 September 2017

Action: 0 of 15 selected

<input type="checkbox"/>	TITLE	PRICE	PUBLISH	PUBLICATION DATE
<input type="checkbox"/>	php	66.00	深圳出版社	June 7, 2017
<input type="checkbox"/>	golang	78.00	工业出版社	May 27, 2017

9. preserve_filters:详细页面，删除、修改，更新后跳转回列表后，是否保留原搜索条件

```

from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish', 'publication_date')

    list_select_related = ["publish"]

    search_fields = ['title', 'classification']

    preserve_filters=True

```

效果如下:

Search bar: go Search 1 result (16 total)

Action: ----- Go 0 of 1 selected

	TITLE	PRICE	PUBLISH	PUBLICATION DATE
<input type="checkbox"/>	golang	78.00	工业出版社	May 27, 2017

10. save_as = False: 详细页面，按钮为“Save as new”或“Save and add another”

```

from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish', 'publication_date')

    list_select_related = ["publish"]

    save_as = True

```

效果如下:

Save as new Save and continue editing SAVE

```

from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish', 'publication_date')

    list_select_related = ["publish"]

    save_as = False

```

效果如下:

Save and add another Save and continue editing SAVE

11. save_as_continue = True: 点击保存并继续编辑

用法与上一个用法相同

12. save_on_top = False: 详细页面，在页面上方是否也显示保存删除等按钮

```

from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish', 'publication_date')

    list_select_related = ["publish"]

    save_on_top = False

```

效果如下:

ect more than one.

[Save and add another](#)[Save and continue editing](#)[SAVE](#)

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'publish', 'publication_date')

    list_select_related = ["publish"]

    save_on_top = True
```

效果如下:

[Save and add another](#)[Save and continue editing](#)[SAVE](#)[Save and add another](#)[Save and continue editing](#)[SAVE](#)

13. inlines:详细页面，如果有其他表和当前表做FK，那么详细页面可以进行动态增加和删除

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title','price')

class BookInline(admin.StackedInline):
    extra=0
    model=models.Book

class PublishAdmin(admin.ModelAdmin):
    inlines = [BookInline,]
    list_display = ['name','addr']

admin.site.register(models.Publish,PublishAdmin)
```

效果如下:

Add publish

Name:

Addr:

BOOKS

Book: #1

Title:

Price:

Publication date:

Today |

Note: You are 8 hours ahead of server time.

Classification:

Authors:

小冰

小明

小红帽

小平

小李

小刘

小磊

Hold down "Control", or "Command" on a Mac, to select more than one.

Book: #2

Title:

Price:

Publication date:

Today |

在编辑或添加出版社的时候可以同时添加多本书籍数据,添加书籍信息的时候是纵向添加书籍信息的
在这里 BookInline 这个类还可以继承 TabularInline 这个类实现横向添加书籍信息


```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price')

class BookInline(admin.TabularInline):
    extra=0
    model=models.Book

class PublishAdmin(admin.ModelAdmin):
    inlines = [BookInline,]
    list_display = ['name', 'addr']

admin.site.register(models.Publish, PublishAdmin)
```

效果如下:

Change publish

HISTORY

Name:

深圳出版社

Addr:

深圳

BOOKS

TITLE	PRICE	PUBLICATION DATE	CLASSIFICATION	AUTHORS	DELETE?
<div>Book object</div> <div>php</div>	66.00	<div>2017-06-06</div> <div>Today</div> <div>Note: You are 8 hours ahead of server time.</div>	计算机	<div> <div>小冰</div> <div>小明</div> <div>小红帽</div> <div>小平</div> <div>小李</div> <div>小刘</div> <div>小福</div> </div>	<input type="checkbox"/>
<div></div>		<div></div> <div>Today</div> <div>Note: You are 8 hours ahead of server time.</div>		<div> <div>小冰</div> <div>小明</div> <div>小红帽</div> <div>小平</div> <div>小李</div> <div>小刘</div> <div>小福</div> </div>	<input type="checkbox"/>

+ Add another Book

14. action:列表时，定制action中的操作

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price')

    def func1(self, request, queryset):          # 定义具体的action的方法
        pass

    func1.short_description = "自定义操作"      # 在页面上显示方法的名称

    actions_on_top = True                      # action在页面上方显示

    actions_on_bottom = False                  # action在页面下方显示

    actions_selection_counter = True            # 显示页面上选择数据的条数

    actions=[func1,]                           # 添加定制的动作
```

在action列表中,添加一个func1的功能

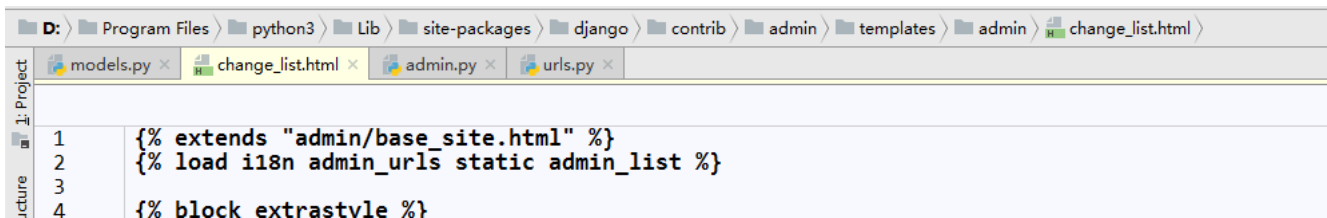
Select book to change

Action: 0 of 16 selected

<input type="checkbox"/>	TITLE	
<input type="checkbox"/>	php	Func1
<input type="checkbox"/>	colony	

15. 定制HTML模板

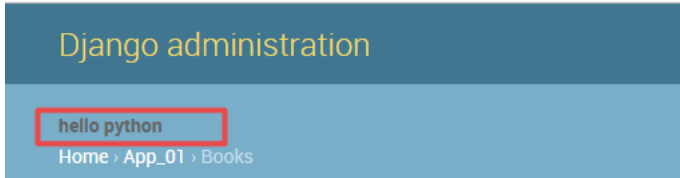
Django后台管理页面默认使用的模板是 `change_list.html`



打开change_list.html页面,添加一行

```
<h3>hello python</h3>
```

刷新浏览器,可以看到



Select book to change

可以设置的选项有:

add_form_template = None	# 添加数据的模板
change_form_template = None	# 修改数据的模板
change_list_template = None	# 显示数据的模板
delete_confirmation_template = None	# 删除数据时确认页面的模板
delete_selected_confirmation_template = None	# 选中之后删除的确认页面
object_history_template = None	# 有ForeignKey时关联数据进行处理的模板

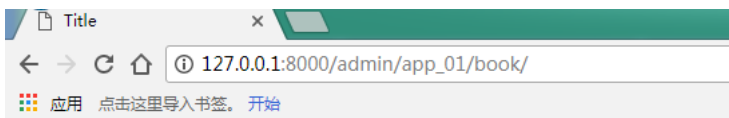
例子:

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price')

    change_list_template = ['change_list1.html']
```

效果如下:



hello python

16. raw_id_fields:详细页面, 针对FK和M2M字段变成以Input框形式

正常显示的时候, FK 和 M2M 字段是通过下拉框来进行操作的

添加 raw_id_fields 配置后,针对 FK 和 M2M 字段变成以 Input 框形式

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'price', 'classification')

    raw_id_fields = ('publish', 'authors')
```

效果如下:

Change book

Title:

Price:

Publish:

深圳出版社 ▼

Publication date:

2017-06-07

 Today |

Note: You are 8 hours ahead of server time.

Classification: Authors:

小冰
小明
小红帽
小平
小李
小刘
小磊

Hold down "Control", or "Command" on a Mac, to select more than one.

Change book

Title:

Price:

Publish:

8

 深圳出版社

Publication date:

2017-06-07

 Today |

Note: You are 8 hours ahead of server time.

Classification: Authors:

17. fields:详细页面时，显示的字段

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    fields = ('title','price')
```

效果如下:

Change book

Title:

Price:

Delete

Save and add another

18. exclude:详细页面时，排除的字段

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    exclude = ('classification',)
```

效果如下:

Change book

Title:	<input type="text" value="php"/>
Price:	<input type="text" value="66.00"/>
Publish:	<input type="text" value="深圳出版社"/>
Publication date:	<input type="text" value="2017-06-07"/> Today
Note: You are 8 hours ahead of server time.	
Authors:	<div><div>小冰</div><div>小明</div><div>小红帽</div><div>小平</div><div>小李</div><div>小刘</div><div>小磊</div></div>
Hold down "Control", or "Command" on a Mac, to select more than one.	

Delete

Save and add another

Save and continue

19. readonly_fields: 详细页面时，只读字段

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    readonly_fields = ('classification',)
```

效果如下:

Change book

Title:

php

Publish:

深圳出版社 ▾

Publication date:

2017-06-07

Today | 📅

Note: You are 8 hours ahead of server time.

Classification:

计算机

Authors:

小冰

小明

小红帽

小平

小李

小刘

小磊

Hold down "Control", or "Command" on a Mac, to select more than one.

Price:

66.00

Delete

Save and add another

Save and c

20. fieldsets:详细页面时，使用fieldsets标签对数据进行分割显示

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    fieldsets = (
        ("基本信息", {
            "fields": ("title", "price", "publish",)
        }),
        ("出版信息", {
            "classes": ("collapse", "wide", "extrapretty"),
            "fields": ("classification", "authors", "publication_date",)
        }),
    ),
)
```

效果如下:

Change book

基本信息

Title:

php

Price:

66.00

Publish:

深圳出版社 ▾

出版信息 (Show)

Delete

Save and add another

Save a

Change book

基本信息

Title:

php

Price:

66.00

Publish:

深圳出版社 ▼

出版信息 (Hide)

Classification:

计算机

Authors:

小冰

小明

小红帽

小平

小李

小刘

小磊

Hold down "Control", or "Command" on a Mac, to select more than one.

Publication date:

2017-06-07

Today | 📅

Note: You are 8 hours ahead of server time.

Delete

21. 详细页面时，M2M显示时，数据移动选择(方向:上下和左右)

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    filter_vertical = ("authors",)
```

效果如下:

Change book

Title:

php

Price:

66.00

Publish:

深圳出版社 ▼

Publication date:

2017-06-07

Today | 📅

Note: You are 8 hours ahead of server time.

Classification:

计算机

Authors:

Available authors ⓘ

🔍 Filter

小明
小李
小磊
小平

Chosen authors ⓘ

小冰
小刘
小红帽

Hold down "Control", or "Command" on a Mac, to select more than one.

Delete

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):

    filter_horizontal = ("authors",)
```

效果如下:

Change book

Title:

php

Price:

66.00

Publish:

深圳出版社

Publication date:

2017-06-07

Today

Note:

You are 8 hours ahead of server time.

Classification:

计算机

Authors:

Available authors

Filter

小明
小红帽
小李
小磊

Choose all

Chosen authors

小冰
小平
小刘

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

Delete

22. ordering:列表时，数据排序规则

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    ordering = ("-price",)
```

效果如下:

Select book to change

ADD B

Action:

Go

 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	CLASSIFICATION	PUBLISH
<input type="checkbox"/>	saltstack	105.00	机械	人民出版社
<input type="checkbox"/>	openstack	89.00	文学	广州出版社
<input type="checkbox"/>	golang	78.00	科学	工业出版社
<input type="checkbox"/>	bootstrap	67.00	计算机	湖北出版社
<input type="checkbox"/>	linux	67.00	机械	广州出版社
<input type="checkbox"/>	php	66.00	计算机	深圳出版社
<input type="checkbox"/>	mysql	55.00	机械	湖北出版社
<input type="checkbox"/>	zabbix	54.00	计算机	上海出版社
<input type="checkbox"/>	swift	51.00	计算机	湖北出版社


```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    ordering = ("price",)
```

效果如下:

Select book to change

ADD BOOK

Action:

Go

 0 of 16 selected

<input type="checkbox"/>	TITLE	PRICE	▲	CLASSIFICATION	PUBLISH
<input type="checkbox"/>	markdown	11.00		科学	重庆出版社
<input type="checkbox"/>	ruby	15.00		文学	工业出版社
<input type="checkbox"/>	perl	28.00		科学	工业出版社
<input type="checkbox"/>	jQuery	34.00		机械	重庆出版社
<input type="checkbox"/>	javascript	35.00		文学	上海出版社
<input type="checkbox"/>	c sharp	45.00		科学	人民出版社
<input type="checkbox"/>	python	48.00		文学	工业出版社
<input type="checkbox"/>	swift	51.00		计算机	湖北出版社

23. view_on_site:编辑时，是否在页面上显示view on set

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    def view_on_site(self, obj):

        return "http://www.baidu.com"
```

点击 view on set 按钮,浏览器会指向 "http://www.baidu.com"

HISTORYVIEW ON SITE ➤

24. radio_fields:详细页面时，使用radio显示选项（FK默认使用select）

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    radio_fields = {"publish": admin.VERTICAL}
```

效果如下:

Change book

Title:

php

Price:

66.00

Publish:

☐ 人民出版社

☐ 工业出版社

☐ 重庆出版社

☐ 湖北出版社

☐ 上海出版社

☐ 广州出版社

☒ 深圳出版社

Publication date:

2017-06-07

Today |

Note: You are 8 hours ahead of server time.

Classification:

计算机

Authors:

小冰

小明

小红帽

小平

小李

小刘

小磊

Hold down "Control", or "Command" on a Mac, to select more than one.

Delete

25. `show_full_result_count = True`:列表时，模糊搜索后面显示的数据个数样式

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    search_fields = ('title', 'classification')

    show_full_result_count = True
```

效果如下:

Select book to change

ADD

Q

p

Search

7 results (16 total)

Action:

Go

0 of 7 selected

<input type="checkbox"/>	TITLE	PRICE	CLASSIFICATION	PUBLISH
<input type="checkbox"/>	php	66.00	计算机	深圳出版社
<input type="checkbox"/>	bootstrap	67.00	计算机	湖北出版社
<input type="checkbox"/>	javascript	35.00	文学	上海出版社
<input type="checkbox"/>	c sharp	45.00	科学	人民出版社
<input type="checkbox"/>	openstack	89.00	文学	广州出版社
<input type="checkbox"/>	perl	28.00	科学	工业出版社
<input type="checkbox"/>	python	48.00	文学	工业出版社

26. `formfield_overrides = {}`:详细页面时，指定现实插件

27. `prepopulated_fields = {}`:添加页面，当在某字段填入值后，自动会将值填充到指定字段

```
from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    search_fields = ('title', 'classification')

    prepopulated_fields = {"title": ("classification",)}
```

效果如下:

Add book

Title:

aaaaaaaaaaaaaaaaaaaa

Price:

Publish:

----- ▾

Publication date:

Today |

Note: You are 8 hours ahead of server time.

Classification:

aaaaaaaaaaaaaaaaaaaa

28. `form =.ModelForm`:用于定制用户请求时候表单验证

未进行配置之前:

Add book

Please correct the errors below.

This field is required.

Title:

This field is required.

Price:

This field is required.

Publish:

----- ▾

This field is required.

Publication date:

Today |

Note: You are 8 hours ahead of server time.

This field is required.

Classification:

修改配置文件:

```

from django.contrib import admin
from . import models
from django.forms import.ModelForm

class MyForm(ModelForm):
    class Meta:
        model=models.Book
        fields="__all__"
        error_messages = {
            "title": {'required': "书名不能为空"},
            "price": {'required': "价格不能为空"}
        } # 自定义错误信息, 用户输入错误时显示

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    search_fields = ('title', 'classification')

    form=MyForm

```

效果如下:

Add book

Please correct the errors below.

Title:

书名不能为空

Price:

价格不能为空

Publish:

This field is required.

Publication date:

This field is required.

Today |

Note: You are 8 hours ahead of server time.

29. empty_value_display = "列数据为空时, 显示默认值"

```

from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    search_fields = ('title', 'classification')

    empty_value_display="列数据为空时, 默认显示"

```

总结:

在定制admin的所有的操作的时候都是在配置文件中完成的.

在配置文件中, 可以进行配置的有数据表的字段, 函数和类

在某个页面中, 可以配置使用自定义的模板.

同样的, 请求到达Django的时候, 也可以配置使其执行自定义的函数

ModelAdmin 是放置在 options.py 文件中的.

打开 options.py 文件, 可以看到 ModelAdmin 里有一个叫做 changelist_view 的方法

```

from django.contrib import admin
from . import models

@admin.register(models.Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ("title", "price", "classification", "publish")

    search_fields = ('title', 'classification')

    empty_value_display="列数据为空时, 默认显示"

```

可以看到, BookAdmin 是继承自 ModelAdmin 这个类的, ModelAdmin 里有一个 changelist_view 的方法

所以在 BookAdmin 中也可以自定义一个 changelist_view 的视图函数.