

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



CUSTOMIZING ADMIN SITE

HOSSEIN FORGHANI

MAKTAB SHARIF

Contents

- ▶ Customizing admin site
- ▶ Customizing admin form, fieldsets, related objects, filter sidebar, search box, and etc.
- ▶ Customizing admin templates
- ▶ Customizing admin actions

Customize the Admin Form

polls/admin.py

```
from django.contrib import admin

from .models import Question

class QuestionAdmin(admin.ModelAdmin):
    fields = ['pub_date', 'question_text']

admin.site.register(Question, QuestionAdmin)
```

Result

[Home](#) › [Polls](#) › [Questions](#) › What's up?

Change question

Date published:

Date:

2015-09-06

Today |



Time:

21:16:20

Now |



Question text:

What's up?

Register Decorator

- ▶ There is also a decorator for registering your ModelAdmin classes:

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

@admin.register(Author)
class AuthorAdmin(admin.ModelAdmin):
    pass
```

Fieldsets

polls/admin.py



```
from django.contrib import admin

from .models import Question

class QuestionAdmin(admin.ModelAdmin):
    fieldsets = [
        (None, {'fields': ['question_text']}),
        ('Date information', {'fields': ['pub_date']}),
    ]

admin.site.register(Question, QuestionAdmin)
```

Result

[Home](#) › [Polls](#) › [Questions](#) › What's up?

Change question

Question text:

What's up?

Date information

Date published:

Date:

2015-09-06

Today | 

Time:

21:16:20

Now | 

Adding Related Objects

polls/admin.py



9

```
from django.contrib import admin

from .models import Choice, Question

class ChoiceInline(admin.StackedInline):
    model = Choice
    extra = 3

class QuestionAdmin(admin.ModelAdmin):
    fieldsets = [
        (None, {'fields': ['question_text']}),
        ('Date information', {'fields': ['pub_date'],
    'classes': ['collapse']})),
    ]
    inlines = [ChoiceInline]

admin.site.register(Question, QuestionAdmin)
```

Add question

Question text:

Date information (Hide)

Date published:

Date:

Today | 

Time:

Now | 

CHOICES

Choice: #1

Choice text:

Votes:

Choice: #2

Choice text:

Votes:

Choice: #3

Choice text:

Votes:

[+ Add another Choice](#)

Tabular Inline

```
polls/admin.py
```

```
class ChoiceInline(admin.TabularInline):  
    #...
```

Result

CHOICES		
CHOICE TEXT	VOTES	DELETE?
<input type="text"/>	<input type="text" value="0"/>	
<input type="text"/>	<input type="text" value="0"/>	
<input type="text"/>	<input type="text" value="0"/>	
+ Add another Choice		

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

filter_horizontal

Restricted services:



Available restricted services ⓘ

Filter

3l

l

A

llation

A

A

A

A

A

A

Appliance removal / disposal

Choose all ⓘ

Chosen restricted services ⓘ

L

M

M

M

3

Remove all ⓘ

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

Customize the Admin Change List

polls/admin.py

```
class QuestionAdmin(admin.ModelAdmin):  
    # ...  
    list_display = ('question_text', 'pub_date',  
                    'was_published_recently')
```

Result

Home > Polls > Questions

Select question to change

Action: 0 of 1 selected

<input type="checkbox"/>	QUESTION TEXT	DATE PUBLISHED	WAS PUBLISHED RECENTLY
<input type="checkbox"/>	What's up?	Sept. 3, 2015, 9:16 p.m.	False

1 question

Improving Method

polls/models.py



```
class Question(models.Model):  
    # ...  
    def was_published_recently(self):  
        now = timezone.now()  
        return now - datetime.timedelta(days=1) <=  
self.pub_date <= now  
        was_published_recently.admin_order_field = 'pub_date'  
        was_published_recently.boolean = True  
        was_published_recently.short_description = 'Published  
recently?'
```


“Filter” sidebar

- ▶ Add the following line to QuestionAdmin:

```
list_filter = ['pub_date']
```


Result

Home › Polls › Questions

Select question to change

ADD QUESTION +

Action:  0 of 1 selected

<input type="checkbox"/>	QUESTION TEXT	DATE PUBLISHED	PUBLISHED RECENTLY?
<input type="checkbox"/>	What's up?	Sept. 3, 2015, 9:16 p.m.	

1 question

FILTER

By date published

[Any date](#)

[Today](#)

[Past 7 days](#)

[This month](#)

[This year](#)

Search Box

- ▶ Add the following line to QuestionAdmin:

```
search_fields = ['question_text']
```

Customize the Admin Look and Feel

- ▶ Open your `settings` file and add a `DIRS` option in the `TEMPLATES` setting
- ▶ Create a directory called `admin` inside templates
- ▶ Copy `admin/base_site.html` from default Django admin template directory into that directory
- ▶ default Django admin template directory:
`django/contrib/admin/templates`
- ▶ Larger projects: put related admin templates in each app

How to Change Admin Site Name

- ▶ Edit the file and replace `{{ site_header | default:_('Django administration') }}`

```
{% block branding %}
<h1 id="site-name"><a href="{% url 'admin:index' %}">Polls
Administration</a></h1>
{% endblock %}
```

Customize the Admin Index Page

- ▶ The template to customize is `admin/index.html`
- ▶ Do the same as with `admin/base_site.html`

مدیریت

مدیریت وبگاه

فعالیت‌های اخیر

فعالیت‌های من

+ بیش از ۳ سال

گزینه

+ ۱ تا ۳ سال

گزینه

+ زیر ۱ سال

گزینه

+ اخیر

گزینه

+ بله

گزینه

+ چقدر جنگو کار کردید؟

سوال

جنگو را دوست دارید؟

سوال

POLLS

سوال‌ها

+ اضافه کردن

تغییر

گزینه‌ها

+ اضافه کردن

تغییر

بررسی اصالت و اجازه‌ها

کاربرها

+ اضافه کردن

تغییر

گروه‌ها

+ اضافه کردن

تغییر

Admin Action

Q Search

Action: ☒ ----- Go 1 of 3 selected

Delete selected users

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input checked="" type="checkbox"/>	adrian	adrian@example.com	Adrian	Holovaty	✖
<input type="checkbox"/>	jacob	jacob@example.com	Jacob	Kaplan-Moss	✔
<input type="checkbox"/>	simon	simon@example.com	Simon	Willison	✖

3 users

Writing actions

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

def make_published(modeladmin, request, queryset):
    queryset.update(status='p')
make_published.short_description = "Mark selected stories as published"

class ArticleAdmin(admin.ModelAdmin):
    list_display = ['title', 'status']
    ordering = ['title']
    actions = [make_published]

admin.site.register(Article, ArticleAdmin)
```


Result

Select article to change

ADD ARTICLE +

Action: ✓ -----

Go

2 of 5 selected

Delete selected articles

Mark selected stories as published

☐

TI



STATUS

☐

A New Human-like Species Discovered in Deep Burial Chamber

Published

☒

Django 1.9 Released

Draft

☐

Mars Is a Real Fixer-Upper of a Planet, Says Elon Musk on Colbert's 'Late Show'

Withdrawn

☒

The Coming of the Glacier Men

Draft

☐

The Last Audio Cassette Factory

Published

5 articles

Actions as ModelAdmin methods

```
class ArticleAdmin(admin.ModelAdmin):  
    ...  
  
    actions = ['make_published']  
  
    def make_published(self, request, queryset):  
        queryset.update(status='p')  
        make_published.short_description = "Mark selected stories as  
published"
```

Messaging after Action

```
from django.contrib import messages
from django.utils.translation import gettext

class ArticleAdmin(admin.ModelAdmin):
    ...


    def make_published(self, request, queryset):
        updated = queryset.update(status='p')
        self.message_user(request, gettext(
            '%d story was successfully marked as published.',
            '%d stories were successfully marked as published.',
            updated,
        ) % updated, messages.SUCCESS)
```

Result

Django administration

WELCOME, **RYAN**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

[Home](#) › [News](#) › [Articles](#)

 2 stories were successfully marked as published.

Select article to change ADD ARTICLE +

Action: 0 of 5 selected

<input type="checkbox"/>	TITLE	STATUS
<input type="checkbox"/>	A New Human-like Species Discovered in Deep Burial Chamber	Published
<input type="checkbox"/>	Django 1.9 Released	Published
<input type="checkbox"/>	Mars Is a Real Fixer-Upper of a Planet, Says Elon Musk on Colbert's 'Late Show'	Withdrawn
<input type="checkbox"/>	The Coming of the Glacier Men	Published

Other Action Functionalities

- ▶ Actions that Provide Intermediate Pages
- ▶ Making actions available site-wide
- ▶ Disabling a site-wide action
- ▶ Disabling all actions for a particular `ModelAdmin`
- ▶ Conditionally enabling or disabling actions
- ▶ Setting permissions for actions
- ▶ Read more in <https://docs.djangoproject.com/en/3.1/ref/contrib/admin/actions/>

Also See

- ▶ `ModelAdmin.ordering`
- ▶ `ModelAdmin.readonly_fields`
- ▶ `ModelAdmin.save_as`
- ▶ `ModelAdmin.sortable_by`
- ▶ And many other options in <https://docs.djangoproject.com/en/3.1/ref/contrib/admin/>

References

- ▶ <https://docs.djangoproject.com/en/3.1/intro/tutorial07/>
- ▶ <https://docs.djangoproject.com/en/3.1/ref/contrib/admin/>
- ▶ <https://docs.djangoproject.com/en/3.1/ref/contrib/admin/actions/>

Any Question?