



무료 전자 책

배우기

# django-admin

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#django-  
admin

.....	1
<b>1: django-admin</b> .....	<b>2</b>
.....	2
.....	2
.....	2
Examples.....	2
.....	2
.....	4
.....	4
django .....	4
<b>2:</b> .....	<b>5</b>
.....	5
Examples.....	5
.....	5
:	5
:	5
:	5
(app, model Django admin).....	5
.....	7
!	9
.....	12

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [django-admin](#)

It is an unofficial and free django-admin ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official django-admin.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# 1: django-admin

Django Admin    CRUD . . . . .

Django . . . . .

- 
- 
- 
- 

1.10	2106-08-01
1.9	2015-12-01
1.8	2015-04-01
1.7	2014-09-02
1.6	2013-11-06
1.5	2013-02-26
1.4	2012-03-23
1.3	2011 3 23
1.2	2010-05-17
1.1	2009-07-29
1.0	2008-09-03

## Examples

. . .

```
# settings.py

# `django.contrib.admin` and its dependancies.
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    ...
]
```

```

MIDDLEWARE = [
    ...
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    ...
]

TEMPLATES = [
    {
        ...,
        'OPTIONS': {
            'context_processors': [
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                ...
            ],
        },
    },
]

```

Django> = 1.9 urls.py .

## 1.9

```

from django.conf.urls import url
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
]

```

## 1.9

```

from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', include(admin.site.urls)),
]

```

include include Django 1.9 .

.

```
$ python manage.py migrate
```

.

```
$ python manage.py createsuperuser
```

.

```
$ python manage.py runserver
```

( <http://127.0.0.1:8000/admin/> ) .

```
admin . admin.py manage.py startapp admin.py app . .
```

```
#myapp/admin.py
from django.contrib import admin
from myproject.myapp.models import MyModel

admin.site.register(MyModel)
```

**ModelAdmin** . :

```
class MyCustomAdmin(admin.ModelAdmin):
    list_display = ('name', 'age', 'email') # fields to display in the listing
    empty_value_display = '-empty-' # display value when empty
    list_filter = ('name', 'company') # enable results filtering
    list_per_page = 25 # number of items per page
    ordering = ['-pub_date', 'name'] # Default results ordering

# and register it
admin.site.register(MyModel, MyCustomAdmin)
```

```
admin.register .
```

```
@admin.register(MyModel)
class MyCustomAdmin(admin.ModelAdmin)
    ...
```

**Django Admin** . .

```
admin . admin.py manage.py startapp admin.py . .
```

```
#myapp/admin.py
from django.contrib import admin
from django.contrib.auth.models import User

admin.site.unregister(User)
```

**django**

```
from django.contrib.auth.models import User
class UserAdmin(admin.ModelAdmin):
    list_display = ('email', 'first_name', 'last_name')
    list_filter = ('is_staff', 'is_superuser')

admin.site.unregister(User)
admin.site.register(User, UserAdmin)
```

django User Model Admin UserAdmin . admin.py User Model . Model ModelAdmin .

**django-admin** : <https://riptutorial.com/ko/django-admin/topic/3499/django-admin->

---

## 2:

get\_price (self) . get\_price . . get\_price .

## Examples

Laravel PHP . Django Django Django HTML CRUD . " . ?" " Yeap!" .

. .

" " ?

(,) . (python manage.py shell) .

```
# python manage.py shell (interactive console)

from math import ceil

from my_app.models import Product

DISCOUNT = 10 # percentage

for product in Product.objects.filter(is_active=True):
    """ Set discount to ALL products that are flagged as active """
    multiplier = DISCOUNT / 100. # DISCOUNT / 100 in python 3 (without dot)
    old_price = product.price
    new_price = ceil(old_price - (old_price * multiplier)) # seller wins :)
    product.price = new_price
    product.save(update_fields=['price'])
```

? ? ? ! . FTW. .

:

- 3.4.3
- 1.10
- SQLite (Python , )

:

- 
- ()

:

- Django Admin .

# (app, model Django admin)

, manage.py stock :

```
python manage.py createapp stock
```

Django .models.py models.py .

```
# models.py

from django.db import models

class Product(models.Model):
    name = models.CharField('Product name', max_length=100) # required field
    price = models.PositiveIntegerField('Product price')

    def __str__(self): # __unicode__ in Python 2
        return self.name
```

Product . [INSTALLED\\_APPS](#) .

settings.py INSTALLED\_APPS .

```
# settings.py

INSTALLED_APPS = [
    # ... previous Django apps
    'stock.apps.StockConfig',
]
```

:

```
python manage.py makemigrations
python manage.py migrate
```

migrate id, name **W** price product . !

<your\_project\_name>/<your\_project\_name>/ [ROOT\\_URLCONF](#) Django URL .

```
# urls.py

urlpatterns = [
    # ... other URLs
    url(r'^admin/', admin.site.urls),
]
```

Django . stock/admin.py .

```
# admin.py
```



```
from django.contrib import admin

from .models import Product

@admin.register(Product)
class ProductAdmin(admin.ModelAdmin):
    pass
```

```
python manage.py runserver
```

127.0.0.1:8000/admin/ . **C**reate- **R**ead- **U**pdate- **D**elete -- Product ! / . User . :

```
python manage.py createsuperuser
```

,, ( ) .

---

( ). Django admin .

100 . name price .

```
python manage.py shell :
```

```
# python manage.py shell

from stock.models import Product

for i in range(1, 101):
    p = Product.objects.create(name='Product %s' % i, price=i)
```

for **100** ( ) ( ) Product 1 , Product 2 , ... Product 100 1 , 2 , ..., 100 .

127.0.0.1:8000/admin/ Products .

# Django administration

## Site administration

### AUTHENTICATION AND AUTHORIZATION

**Groups**

[+ Add](#) [✎ Change](#)

**Users**

[+ Add](#) [✎ Change](#)

### STOCK

**Products**

[+ Add](#) [✎ Change](#)

100 .

# Django administration

Home > Stock > Products

Select product to change

Action:

**PRODUCT**

**Product 100**

**Product 99**

**Product 98**

**Product 97**

**Product 96**

**Product 95**

**Product 94**

**Product 93**

**Product 92**

**Product 91**

**Product 90**

Django `change list` `change list` . `stock/admin.py` .

```
@admin.register(Product)
class ProductAdmin(admin.ModelAdmin):
    list_display = ('name', 'price')
```

.



, ., 30 % .

`change list` Action ? Django **D** elete .

..stock/admin.py .

```
# admin.py

@admin.register(Product)
class ProductAdmin(admin.ModelAdmin):
    list_display = ('name', 'price')
    actions = ['discount_30']

    def discount_30(self, request, queryset):
        from math import ceil
        discount = 30 # percentage

        for product in queryset:
            """ Set a discount of 30% to selected products """
            multiplier = discount / 100. # discount / 100 in python 3
            old_price = product.price
            new_price = ceil(old_price - (old_price * multiplier))
            product.price = new_price
            product.save(update_fields=['price'])
        discount_30.short_description = 'Set 30%% discount'
```

- ProductAdmin (actions). ( ).
- ProductAdmin action-function. ModelAdmin (self), HTTP request queryset (queryset - products) .
- (short\_description) (% %).

function-action ( ) 30 % . update\_fields ( UPDATE ) UPDATE update\_fields save() .  
2 .

change list delete . Set 30% discount Go . !

## Select product to change

Action:   0 of 100 selected

<input type="checkbox"/>	Product 100
<input type="checkbox"/>	Product 99
<input type="checkbox"/>	Product 98
<input type="checkbox"/>	Product 97
<input type="checkbox"/>	Product 96
<input type="checkbox"/>	Product 95

`admin.py`

[https://riptutorial.com/ko/django-admin/topic/7747/-](https://riptutorial.com/ko/django-admin/topic/7747/)

---

S. No		Contributors
1	django-admin	<a href="#">Antoine Pinsard</a> , <a href="#">Community</a> , <a href="#">Iker</a> , <a href="#">NeErAj KuMaR</a> , <a href="#">Roald Nefs</a> , <a href="#">Udi</a>
2		<a href="#">ira</a> , <a href="#">nik_m</a> , <a href="#">Stryker</a>