



無料電子ブック

学習

django-admin

Free unaffiliated eBook created from
Stack Overflow contributors.

#django-
admin

.....	1
1: django-admin	2
.....	2
.....	2
.....	2
Examples	3
Django.....	3
.....	4
.....	5
django.....	5
2:	6
.....	6
Examples	6
.....	6
.....	6
.....	7
.....	7
Django	7
.....	8
.....	10
.....	13

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

1: django-adminをいめる

Django Adminは、 [Django](#) WebフレームワークのCRUDインタフェースです。ほとんどの、にされますが、くカスタマイズすることができます。ただし、できるユーザーけにされており、があることにするがあります。いずれにしても、できないユーザーにはをえることはにしてはいけません。

Django Adminはなカスタマイズをしますが、あまりにもくのカスタマイズのをとさないようにしてください。もしそうなら、おそらく Django Adminをわずにのカスタムインタフェースをするがます。

リソース

- [Djangoの](#)
- [Django Adminチュートリアル](#)
- [Djangoのドキュメント](#)
- [Djangoのソースコード](#)

バージョン

バージョン	
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	2011-03-23
1.2	2010-05-17
1.1	2009729
1.0	2008-09-03

Examples

Djangoの

Django adminをいめるためになものはすべて、Djangoのデフォルトのプロジェクトレイアウトでにされています。これも

```
# 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でなるurls.pyについては、いバージョンよりもがです。

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`きのバージョンはDjango 1.9でもし`include`が、され、されるです。

まだされていないは、をするがあります。

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

にアクセスするには、スーパーユーザーをするがあります。

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

これがしたら、サーバーをできます

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

ページ <http://127.0.0.1:8000/admin/>にアクセスしてください。

ページにモデルをする

アプリでのモデルをしたでも、ページでできるようにするには、 するがあります。

これは`admin`サブモジュールでわれます。あなたのアプリが`manage.py startapp`をとてされた、`admin.py`ファイルはにあなたの`app`モジュールにかれているはずです。そののはします。

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

admin.site.register(MyModel)
```

すべてのオプションは、`ModelAdmin`サブクラスでされています。いくつかのオプション

```
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)
```

モデルをするよりなは、 `admin.register`デコレータをすること `admin.register`。

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

ページからモデルをする

Django Adminにはデフォルトでいくつかのモデルがされています。ページからモデルをしたいがあります。

これは `admin` サブモジュールでわれます。あなたのアプリが `manage.py startapp` をってされたものであれば、 `admin.py` ファイルはにあなたのアプリケーションモジュールにかれているはずです。それのはします。

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

admin.site.unregister(User)
```

django ユーザモデルをカスタマイズする

```
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がにされているので、カスタムUserAdminをするにをすることがあります。まず `admin.py` で User Model のをしてから、User Model をカスタムModelAdminにするがあります

オンラインで `django-admin` をいめるをむ <https://riptutorial.com/ja/django-admin/topic/3499/django-adminをいめる>

2: の

はあなたがにもししていないときにget_priceselfをとしないとう。はget_priceメソッドをします。これは、モデルのでからじをることができるからです。はっているがある。ここではget_priceメソッドのはされません。

Examples

ディスカウントアクション

ある、[Laravel](#) PHPフレームワークをにっているとししました。Djangoには、[Django admin](#)とばれるデータベースとやりとりするために、[Django](#)にHTML CRUDシステムがすべてまれているとったとき、のがりました。はにった"ののWebアプリケーションのインターフェイスをするにはヶかかりましたが、あなたは1のコードをなくともこれらすべてをっているといえますか"は"イエップ"とえた。

Djangoのは、なです[ジャンゴ](#)くのグッズをします。これらのうちの1つは[アクション](#)です。

しかし、「」はですか

モデルがあり、にいくつかのエントリ、をしているとしします。は、コンソール `python manage.py shell` をせずに、なくとも1つのルールアクションをしたいとしします。

```
# 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'])
```

すべてののにをしたことにながきましたかこのをののにしたいのであればどうでしょうかまたは、のをでしてこのをのにするは、これらすべてはDjangoのをじてあなたはしいをいています。DjangoアクションFTW。なをてみましょう。

- Python 3.4.3
- Django 1.10
- SQLitePythonでビルドされ、なインストールセットアップが

モデル

- たちのeショップにを
- のはではありません
- Djangoインターフェースをして、1つまたはのエントリにをすることができます。

セットアップアプリ、モデル、Django

あなたがにしているとすると、プロジェクトをし、どこディレクトリにmanage.pyのとでアプリをするstockして、

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

Djangoはにディレクトリをします。models.pyファイルをしてする

```
# 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
```

のProductモデルはされましたが、データベースにはまだありません。をINSTALLED_APPSには、アプリがINSTALLED_APPSリストにまれているがあります。

settings.pyファイルをし、INSTALLED_APPSリストのにします

```
# settings.py

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

すぐする

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

migrateコマンドの、データベースにはproductというのテーブルがされ、id、name、price 3つのカラムがまれています。ここまではですね

<your_project_name>/<your_project_name>/ フォルダにある **ROOT_URLCONF** ファイルのをしていない、Django サイトをす URL はのようになります。

```
# urls.py

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

これまでは、Django のアクションについてはもていませんでした。のステップをんで、stock/admin.py ファイルのにしてください

```
# admin.py

from django.contrib import admin

from .models import Product

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

OK。セットアップがしました。すべてがすることをするには、のコマンドをします。

```
python manage.py runserver
```

おみのブラウザで 127.0.0.1:8000/admin/ アクセスしてください。あなたは、あなたの **C**reate- **R**ead- **U**ppdate- **D**elete することができますのある、な、らしい Django のページ、されるはず **Product** モデルをが、のページでユーザー/パスワードのをめられても、もはありません。にログインするための **User** をしていないだけです。にする

```
python manage.py createsuperuser
```

あなたの、メール、ユーザーとパスワードを2すればです。

のをする

これまではモデルをしました、まだエントリはありませんはありません。たちは Django のアクションのをらかにするためにいくつかのエントリがです。

たちは100のをり、それをとてします。しかし、でADD ボタンをして **name** と **price** をするのではなく、たちののをするためのスクリプトをします。

python manage.py shell をし、のようにします。

```
# 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ループがでデータはデータベースにされていることをする100のエントリーをしProduct 1、
Product 2 ...、 Product 100および1、 2、 ...、 100。

Djangoのページからこれらのをするには、もう127.0.0.1:8000/admin/ アクセスしてProductsリンク
をクリックしてください

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

[+ Add](#) [Change](#)

Users

[+ Add](#) [Change](#)

STOCK

Products

[+ Add](#) [Change](#)

された100のをおしみてください

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**

これは、Djangoの`change list`ページとされます。これをもっときれいにできるようにするには、`stock/admin.py` ファイルをしてのようになります。

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

さて、リフレッシュをすと、をする2のがされます。

ピーアクションの

すると、々はモデルをっており、々はエントリをっている。に、したに30のをするアクションを

します。

change list ページのに、Action のラベルがいたボックスがにあることにきましたか Django は、Delete アクションをするエントリにデフォルトアクションをにします。

Django のアクションはなとしてかれています。 stock/admin.py ファイルをし、をしてください

```
# 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'
```

ここですべきがいくつかあります。

- ProductAdmin クラスには、のリストである1つのな actions がありますはアクションをするので
- ProductAdmin クラスのメソッドであるaction-function。これはとしてみるModelAdmin えばself、このため HTTP request オブジェクトと queryset されたオブジェクト・エントリ・のリスト。
- のは、アクションボックスにされたをする short_description ですのをエスケープするためにそこにダブルがあります。

ファンクションアクションでは、されたについてをい、そのを30させました。その、パフォーマンスのから、データベースのupdate_fields リストにまれるフィールドモデルのすべてのフィールドのUPDATEではなくにUPDATEをにするために、update_fields をして save() メソッドをびします。このではパフォーマンスが2しかられませんが、そのがかります。

change list ページでリフレッシュをすると、delete されたアクションのにアクションがされます。にくとのチェックボックスをして、またはすべてののにあるチェックボックスをしていくつかのをし、Set 30% discount アクションをクリックしGo ボタンを。それでおしまい

Django administration

Home › Stock › Products

Select product to change

Action: 0 of 100 selected

☐	Product 100
☐	Product 99
☐	Product 98
☐	Product 97
☐	Product 96
☐	Product 95

もちろん、これはほとんどのではありません。このアクションでは、なるのをする
ことはできません。のをするたびに、`admin.py`ファイルをするがあります。のでは、これをうを
いきます。

オンラインでのをむ <https://riptutorial.com/ja/django-admin/topic/7747/>の

クレジット

S. No		Contributors
1	django-adminをいめる	Antoine Pinsard , Community , Iker , NeErAj KuMaR , Roald Nefs , Udi
2	の	ira , nik_m , Stryker