

Django admin ir signalai

Albertas Gimbutas

¹Matematikos ir informatikos institutas
Vilniaus universitetas

2014 pavasaris

Modelių administravimo iniciavimas

```
from django.contrib import admin
from myapp.models import MyModel, AnotherModel

admin.site.register(MyModel)
```

```
class AnotherModelAdmin(admin.ModelAdmin):
    fields = ('title', 'description')
    exclude = ('created',)

admin.site.register(MyModel, AnotherModel)
```

- django tutorial: Admin

Modelių administravimo deklaravimas

```
class FlatPageAdmin(admin.ModelAdmin):
    fieldsets = (
        (None, {
            'fields': (('url', 'title'),
                       'content',
                       'sites')
        }),
        ('Advanced options', {
            'classes': ('collapse',),
            'fields': ('enable_comments', 'template_name')
        }),
    )
```

- django: Admin

AdminModel atributų ir metodų paskirtis

`list_display`

Modelio reprezentacijoje rodytinų
laukų sąrašas

`readonly_fields`

Šablono bloko žymė

`inlines`

Susijusių objektų rodymas ir kūrimas
tame pačiame puslapyje

`formfield_overrides`

Formos laukų atributų perdengimas

`form`

Naudotinos formos klasė

`list_filter`

Atributų sąrašas pagal kuriuos galima filtruoti

`ordering`

Objektų pateikimo eiliškumas

Metodai:

`get_queryset()`

Objektų pateikimo eiliškumas

`has_add_permission()`

Ar naudotojui leidžiama pridėti

`has_change_permission()`

Ar naudotojui leidžiama redaguoti

`has_delete_permission()`

Ar naudotojui leidžiama trinti

šio modelio objektus

- ➊ ✓ Žodžio įvedimas, žodžių sąrašo peržiūra
- ➋ ✓ Vertimo įvedimas
- ➌ Vertimo patikimas/nepatikimas
- ➍ Vertimo komentavimas
- ➎ Atsitiktinių žodžių pateikimas vertinimui/vertimui
- ➏ Žodžio vertimo pateikimas atsižvelgiant į vertimo kokybę
- ➐ Naudotojai:
 - ➊ ✓ Prisijungimas
 - ➋ Registracija
 - ➌ Galimybė pažymėti terminą kaip vartotiną (LKK naudotojas)