

Урок 13

Кастомизируем админку

Подмена главного шаблона

admin/custom_index.html

```
{% extends "admin/index.html" %}
```

```
{% block branding %}
```

```
    {{block.super}}
```

```
    <a class="btn btn-primary" href="/admin/extra/">My extra link</a>
```

```
{% endblock %}
```

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

```
admin.site.index_template = 'admin/custom_index.html'
```

```
admin.autodiscover()
```

СВОИ ВИДЫ В АДМИНКЕ

```
from django.conf.urls import patterns
from django.http import HttpResponseRedirect

def my_view(request):
    return HttpResponseRedirect("Hello!")

def get_admin_urls(urls):
    def get_urls():
        my_urls = patterns(
            (r'^my_view/$', admin.site.admin_view(my_view))
        )
        return my_urls + urls
    return get_urls

admin_urls = get_admin_urls(admin.site.get_urls())
admin.site.get_urls = admin_urls
```

Своя ссылка в списке

```
class CurrencyAdmin(admin.ModelAdmin):  
    list_display = ( 'date', 'rub2dram', 'dram2usd', 'my_link' )  
  
    def my_link(self, obj):  
        return '<a href="/admin/recount/%s">%s</a>' % (obj.id, 'Recount')  
my_link.allow_tags = True
```

Работа с формами

```
from django import forms

class NameForm(forms.Form):
    your_name = forms.CharField(label='Your name', max_length=100)

from django.shortcuts import render
from django.http import HttpResponseRedirect

def get_name(request):
    if request.method == 'POST':
        form = NameForm(request.POST)
        if form.is_valid():
            return HttpResponseRedirect('/thanks/')
    else:
        form = NameForm()

    return render(request, 'name.html', {'form': form})
```

Работа с формами

```
<form action="/your-name/" method="post">
  {% csrf_token %}
  {{ form }}
  <input type="submit" value="Submit" />
</form>
```

EmailField() BooleanField(required=False) ChoiceField(choices) DateField DateTimeField DecimalField

EmailField FileField FilePathField FloatField ImageField IntegerField IPAddressField

GenericIPAddressField MultipleChoiceField TypedMultipleChoiceField NullBooleanField

RegexField SlugField TimeField URLField

Формы модели

```
#in forms.py

class UserForm(forms.ModelForm):
    class Meta:
        model = User
        fields = ["username", "email"]
```

```
#in template

<form method="post">
    {{ uform.as_p }}
    <input type="submit" ...>
</form>
```

Шаблонные теги

```
catalog/templatetags/catalog_tag.py
# -*- coding: utf-8 -*-
from django import template
from catalog.models import Catalog
register = template.Library()
from django.utils.translation import ugettext as _
@register.inclusion_tag("catalog/catalog.html")
def catalog_tag(t='paper'):
    out = {}
    c = Catalog.objects.filter(category_type=t).order_by('sorting')
    out['items'] = c
    if t == 'magazine':
        out['title'] = _(u'Журналы')
    if t == 'paper':
        out['title'] = _(u'Газеты')
    return out
```


Шаблонные теги

catalog/templates/catalog.html

```
<h2> {{ title }} </h2>
```

```
<ul class="nav nav-pills nav-stacked">
```

```
    {% for i in items %}
```

```
        <li> <a href="{{i.get_absolute_url}}">{{i}}</a> </li>
```

```
    {% endfor %}
```

```
</ul>
```

```
{% load catalog_tag %}
```

```
{% catalog_tag t=magazine %}
```

Деплой фабрикой

```
pip install fabric
project/fabfile.py
# -*- coding: utf-8 -*-
from fabric.api import *
import os
from contextlib import contextmanager as _contextmanager

def production_env():
    """Окружение для продакшена"""
    env.user = 'webmaster' # На сервере будем работать из под пользователя "webmaster"
    env.project_root = '/home/webmaster/wc_ve/chat' # Путь до каталога проекта (на сервере)
    env.password = '1q2w3e'
    env.activate = 'source /home/webmaster/wc_ve/bin/activate'
env.hosts = ['chat.mirbu.com']
```

Деплой фабрикой

```
@_contextmanager
```

```
def virtualenv():
```

```
    with cd(env.project_root):
```

```
        with prefix(env.activate):
```

```
            yield
```

```
def deploy():
```

```
    production_env() # Инициализация окружения
```

```
    with virtualenv():
```

```
        run('git pull') # Пуляемся из репозитория
```

```
        run('pip install -r requirements.txt') # ставим пакеты
```

```
        run('bower install')
```

```
        run('./manage.py collectstatic --noinput') # Собираем статику
```

```
        run('./manage.py migrate')
```

```
        run('find . -name "*.mo" -print -delete') # Чистим старые скомпиленные файлы gettext'a
```

```
        run('./manage.py compilemessages') # Собираем новые файлы gettext'a
```