

Урок 12

Тэги в django

```
pip install django-taggit
from taggit.managers import TaggableManager
```

```
class Food(models.Model):
    # ... fields here
```

```
    tags = TaggableManager()
```

```
>>> apple = Food.objects.create(name="apple")
>>> apple.tags.add("red", "green", "delicious")
>>> apple.tags.all()
[<Tag: red>, <Tag: green>, <Tag: delicious>]
>>> apple.tags.remove("green")
>>> apple.tags.all()
[<Tag: red>, <Tag: delicious>]
>>> Food.objects.filter(tags__name__in=["red"])
[<Food: apple>, <Food: cherry>]
```

Загрузка файлов

```
import os
```

```
def get_upload_path(instance, filename):
```

```
    return os.path.join( "user_%d" % instance.owner.id, "car_%s" % instance.slug, filename)
```

```
photo = models.ImageField(upload_to=get_upload_path)
```

Компрессия статики

```
pip install django_compressor
```

```
INSTALLED_APPS = (  
    # other apps  
    "compressor",  
)  
  
STATICFILES_FINDERS = (  
    # other finders..  
    'compressor.finders.CompressorFinder',  
    ...)
```

- Define `COMPRESS_ROOT` in settings if you don't have already `STATIC_ROOT` or if you want it in a different folder.

```
{% load compress %}  
{% compress <js/css> [<file/inline> [block_name]] %}  
<html of inline or linked JS/CSS>  
{% endcompress %}
```

Менеджеры модели

```
from django.db import models

class ContactManager(models.Manager):

    def get_contact_list(self, user_id, tpa_id) :

        from main.models import ChatUser2Room

        contacts = None

        try:

            contacts = ChatUser2Room.objects.raw("""

                SELECT DISTINCT u1.room_id AS id, u2.user_id AS user_id, us.name AS name

                .....

                WHERE  u1.user_id = '%s' AND u1.room_id = u2.room_id AND u1.tpa_id = u2.tpa_id AND u1.tpa_id = '%s' AND u1.deleted = FALSE AND u2.user_id != '%s'

            """, params=[user_id, tpa_id, user_id])

        except ObjectDoesNotExist:

            pass

        return contacts

from main.managers import ContactManager

objects = models.Manager() # The default manager.

contacts = ContactManager() # The specific manager.
```

Сигналы

- `django.db.models.signals.pre_save` & `django.db.models.signals.post_save`
- Sent before or after a model's `save()` method is called.
- `django.db.models.signals.pre_delete` & `django.db.models.signals.post_delete`
- Sent before or after a model's `delete()` method or queryset's `delete()` method is called.
- `django.db.models.signals.m2m_changed`
- Sent when a `ManyToManyField` on a model is changed.
- `django.core.signals.request_started` & `django.core.signals.request_finished`
- Sent when Django starts or finishes an HTTP request.

Слушаем сигналы

To receive a signal, you need to register a *receiver* function that gets called when the signal is sent by using the `Signal.connect()` method:

```
def my_callback(sender, **kwargs):  
    print("Request finished!")
```

```
from django.core.signals import request_finished
```

```
request_finished.connect(my_callback)
```

```
from django.db.models.signals import pre_save
```

```
from django.dispatch import receiver
```

```
from myapp.models import MyModel
```

```
@receiver(pre_save, sender=MyModel)
```

```
def my_handler(sender, **kwargs):
```

Создаем свой сигнал

```
import django.dispatch

pizza_done = django.dispatch.Signal(providing_args=["toppings", "size"])

class PizzaStore(object):
    ...

    def send_pizza(self, toppings, size):
        pizza_done.send(sender=self.__class__, toppings=toppings, size=size)
    ...
```


Виджет color-picker

pip install django-paintstore

```
INSTALLED_APPS = (
```

```
    # ...
```

```
    'paintstore',
```

```
)
```

```
from paintstore.fields import ColorPickerField
```

```
color = ColorPickerField()
```