Урок 8

urllib2 + xml

```
import urllib2
from xml.dom import minidom
url = 'http://www.litres.ru/genres_list/'
doc = urllib2.urlopen(url)
dom = minidom.parse(doc)
items = dom.getElementsByTagName('subgenres')
for cat in items:
        print 'doing -'+cat.getAttribute('title')
logger.error("Done")
```

Обработка исключений

```
import urllib2
from xml.dom import minidom
from urllib2 import HTTPError
url = 'http://www.litres.ru/genres_listeee/'
try:
        doc = urllib2.urlopen(url)
        dom = minidom.parse(doc)
        items = dom.getElementsByTagName('subgenres')
        for cat in items:
        print 'doing -'+cat.getAttribute('title')
except HTTPError:
        print 'something went wrong!!!'
else:
        print('well done')
finally:
        print('I closed file.')
```

Знакомство с supervisor

pip install supervisor

sudo apt-get install -y supervisor
Supervisor is a client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems.
supervisord - демон, который управляет (запуск/остановка/перезапуск) процессами;
supervisorctl - скрипт, который общается с **supervisord** с помощью XML-RPC;

touch supervisord.conf

```
#!/usr/bin/python

import  time

def log_info(msg):
        f = open('log.txt','a')
        f.write(msg)
        f.close()
    if __name__ == "__main__":
        while True:
        log_info('I am alive.')
        print 'I am alive'
        time.sleep(1)
```

Знакомство с supervisor

```
[program:testapp]
command=./app.py
autorestart=true
redirect_stderr=true
stdout_logfile=/home/proft/temp/app_sd.log
```

Пишем команды на Django.

app/managment/commands/mycommand.py

./manage.py mycommand

```
# -*- coding: utf-8 -*-
import logging
logging.basicConfig()
from optparse import make_option
from django.core.management.base import BaseCommand
from catalog.models import Mymodel
logger = logging.getLogger(__name__)
logger.setLevel(logging.DEBUG)

class Command(BaseCommand):
        def handle(self, *args, **options):
                logger.info("Start.....")
                logger.info("Done")
```

Скин для админки.

```
pip install django-grappelli
INSTALLED_APPS = (
    'grappelli',
    'django.contrib.admin',
)
```

before `django.contrib.admin`!

```
urlpatterns = patterns('',
    (r'^grappelli/', include('grappelli.urls')), # grappelli URLS
    (r'^admin/',  include(admin.site.urls)), # admin site
)
TEMPLATE_CONTEXT_PROCESSORS = (
    ...
    "django.core.context_processors.request",
)
python manage.py collectstatic
```

Работа с изображениями.

easy-thumbnails

sorl-thumbnail==11.12.1b

django-image-cropping

models.py

```
from django.utils.safestring import mark_safe
from easy_thumbnails.files import get_thumbnailer
image  = models.ImageField(upload_to='news', verbose_name=u'Изображение', blank=True)
cropping = ImageRatioField('image', '195x130')
@property
def thumb(self):
    try:
        thumbnail_url = get_thumbnailer(self.image).get_thumbnail({ 'size': (195, 130), 'box': self.cropping, 'crop': True,}).url
            return mark_safe(u'<img src="%s" />' % thumbnail_url)
    except:
        return ''
```

Работа с изображениями.

admin.py

```
from image_cropping import ImageCroppingMixin
class NewsAdmin(ImageCroppingMixin, admin.ModelAdmin):
```