

Мастерская интернет-разработчика

Введите что-нибудь и нажмите Enter

Найти


[Toggle to English articles](#)

[Блог](#) [Контакты](#) [Портфолио](#) [Сервисы](#) [Закладки](#) [Винилист](#)

Django. Авторизация через социальные сети с помощью django-social-auth

Подобається!

0

Tweet

0

13

[Share this!](#)

Авторизация на сайтах через социальные сети очень удобна, не надо заполнять регистрационные формы и т.д., вам достаточно нажать на соответствующую кнопку и вас авторизует сам сайт, при этом нужные данные он сможет получить с сайта источника (той социальной сети, с которой выполняется авторизация).



На своих проектах я использую клёвую библиотеку [django-social-auth](#), до неё пробовал [django-ulogin](#), но сам сервис **uLogin** мне не понравился ([мои сумбурные заметки](#) об этом). На пыхе также [обсуждали другие библиотеки](#), в частности **loginza**, но отзывы о ней плохие, поэтому даже не пробовал. В любом случае сервисы - это сервисы, и зависимость от сервисов это больше проблема (частые падения сервисов, убогие виджеты и внешний вид без кастомизаций и т.п.) чем зависимость от библиотеки, которая проксирует запросы к конечным "социальным сетям".

Далее будем обсуждать только **django-social-auth**.

Вообще, нам интересно два репозитория:

- Основной, от [omab](#);
- И второй, со свежей поддержкой русских социалок от [krvss](#), в который я пулял фиксы.

Регистрации в сетях

Для начала нам надо получить ключи от необходимых социальных сетей, на [странице проекта в GitHub](#) есть инструкции для множества социальных сетей, но в свое время у меня были трудности и я решил записать последовательность действий для некоторых из них.

Facebook

Зайдите на <https://developers.facebook.com/apps/> и нажмите на [+ Create New App](#). Введите название приложения (название сайта или проекта), после сабмита формы вы увидите реквизиты "App ID" и "App Secret".

Добавьте их в свой `settings.py`, пример:

```
FACEBOOK_APP_ID = '696381432507483'
FACEBOOK_API_SECRET = '15afb0bbeb173aae12e8e875ffccc7a4'
```

Теперь заполните поле "App Domains", укажите через пробел домены (например один локальный, другой продакшен домен). Поставьте галочку на "Website with Facebook Login" и введите адрес для редиректа, я редиректу в корень продакшен сайта.

Twitter

Зайдите на <https://dev.twitter.com/> и введите логин и пароль от вашей учетной записи в **Twitter**. Далее заходите на <https://dev.twitter.com/apps> и жмёте на [Create a new application](#), заполните нужные поля и соглашаетесь с правилами, после чего вы получите "Consumer key" и "Consumer secret".

Добавьте их в `settings.py`, пример:

```
TWITTER_CONSUMER_KEY = 'G2wMq4KYpTmgZDcjg0EzQ'
TWITTER_CONSUMER_SECRET = 'rGHMGib0wIEpoxjXz0ahc2KmvxY8h10DpZ90LwqEjec'
```

По умолчанию вам выдадут Access level "Read-only", для авторизации этого вам хватит. Рекомендую прочитать [The Application Permission Model](#).

Вконтакте

Зайдите на страницу <http://vk.com/developers.php> и нажмите [Создать приложение](#), выберите Тип "Веб-сайт" и введите адрес сайта и имя домена. В ответ получите "ID приложения" и "Защищенный ключ".

Добавьте их в `settings.py`, пример:

```
VK_APP_ID = '1234567'
```

```
VKONTAKTE_APP_ID = VK_APP_ID
VK_API_SECRET = 'Q0owlQE50XRYd2lcnLa'
VKONTAKTE_APP_SECRET = VK_API_SECRET
```

Для чего дублировать переменные, я [спрашивал в блоге автора](#) и получил ответ. Также смотрите [документацию](#) по бекенду.

Google+

Зайдите на страницу <https://code.google.com/apis/console/>, нажмите `Create`, введите требуемые данные и во вкладке `API Access` увидите "Client ID" и "Client secret".

Добавьте их в `settings.py`, пример:

```
GOOGLE_OAUTH2_CLIENT_ID = '123456789.apps.googleusercontent.com'
GOOGLE_OAUTH2_CLIENT_SECRET = 'p0dJSDjs-dAJsdSAdaSDadasdrt'
```

GitHub

Зайдите на страницу <https://github.com/settings/applications/new> и введите логин и пароль от вашей учетной записи в **GitHub**. Введите имя приложения, адреса сайта для "URL" и "Callback URL" (у меня это один адрес корневой страницы сайта). И получите "Client ID" и "Client Secret", после чего добавьте их в `settings.py`:

```
GITHUB_APP_ID = 'da3bad06987041629b96'
GITHUB_API_SECRET = '8bb53dd4a0b1bbc12f77e147c11d5fd6082adb8d'
```

Теперь перейдем к настройке **social_auth**.

Установка и настройка django-social-auth

Для начала установим приложение:

```
pip install django-social-auth
```

Теперь отредактируйте `settings.py`:

```
# Добавляем в AUTHENTICATION_BACKENDS нужные бекенды,
# смотрите полный список https://github.com/omab/django-social-auth/blob/master/doc/configuration.rst
AUTHENTICATION_BACKENDS = (
    'social_auth.backends.twitter.TwitterBackend',
    'social_auth.backends.facebook.FacebookBackend',
    'social_auth.backends.contrib.vk.VKOAuth2Backend',
    'social_auth.backends.google.GoogleOAuth2Backend',
    'social_auth.backends.contrib.github.GithubBackend',
    'django.contrib.auth.backends.ModelBackend',
)

# Добавляем в TEMPLATE_CONTEXT_PROCESSORS процессор "social_auth_by_name_backends"
TEMPLATE_CONTEXT_PROCESSORS = (
    'django.contrib.auth.context_processors.auth',
    'django.core.context_processors.request',
    'social_auth.context_processors.social_auth_by_name_backends',
)

# И добавляем "social_auth" в INSTALLED_APPS
INSTALLED_APPS = (
    ...
    'social_auth',
)
```

Дополнительные настройки **social_auth**, добавьте в `settings.py`:

```
import random
# Если имя не удалось получить, то можно его сгенерировать
SOCIAL_AUTH_DEFAULT_USERNAME = lambda: random.choice(['Darth_Vader', 'Obi-Wan_Kenobi', 'R2-D2', 'C-3P0', 'Yoda'])
# Разрешаем создавать пользователей через social_auth
SOCIAL_AUTH_CREATE_USERS = True

# Перечислим pipeline, которые последовательно буду обрабатывать респонс
SOCIAL_AUTH_PIPELINE = (
    # Получает по backend и uid инстансы social_user и user
```

```
'social_auth.backends.pipeline.social.social_auth_user',
# Получает по user.email инстанс пользователя и заменяет собой тот, который получили выше.
# Кстати, email выдает только Facebook и GitHub, а Vkontakte и Twitter не выдают
'social_auth.backends.pipeline.associate.associate_by_email',
# Пытается собрать правильный username, на основе уже имеющихся данных
'social_auth.backends.pipeline.user.get_username',
# Создает нового пользователя, если такого еще нет
'social_auth.backends.pipeline.user.create_user',
# Пытается связать аккаунты
'social_auth.backends.pipeline.social.associate_user',
# Получает и обновляет social_user.extra_data
'social_auth.backends.pipeline.social.load_extra_data',
# Обновляет инстанс user дополнительными данными с бекенда
'social_auth.backends.pipeline.user.update_user_details'
)
```

Подключите роуты в `urls.py`:

```
urlpatterns = patterns('',
    ...
    url(r'', include('social_auth.urls')),
)
```

Синхронизация БД:

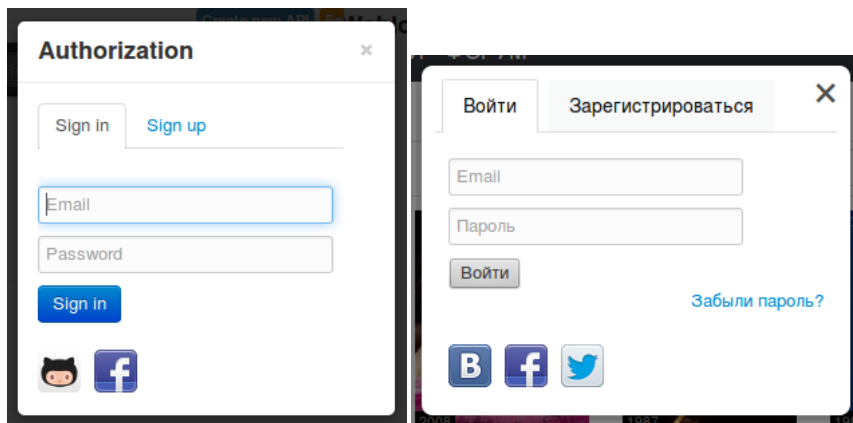
```
./manage.py syncdb
./manage.py migrate social_auth
```

Документацию можно посмотреть тут:

- [Основная документация](#)
- [Документации по pipeline](#)

Виджет для вывода кнопочек

Это просто шаблонный inclusion-тег, который выводит ссылки в виде кнопочек-логотипов социальных сетей, выглядит примерно так:



Сам виджет я не стал публиковать в PyPI, так как не считаю это важным, вам следует просто скопировать исходники отсюда <https://github.com/adw0rd/django-social-auth-widget>

После чего добавьте в `settings.py` следующий список:

```
SOCIAL_AUTH_PROVIDERS = [
    {'id': p[0], 'name': p[1], 'position': {'width': p[2][0], 'height': p[2][1], }},
    for p in (
        ('github', u'Login via GitHub', (0, -70)),
        ('facebook', u'Login via Facebook', (0, 0)),
        ('twitter', u'Login via Twitter', (0, -35)),
    )
]
```

В вашем шаблоне с формой авторизации использовать так:

```
{% load social_auth_widget %}

<form action="" method="post">
```

```

<input name="username" />
<input name="password" />
<input type="submit" value="Sign in" />
{% social_auth_widget %}
</form>

```

Обновление аватарок при авторизации

В статье [Django Social Auth: now with images](#) имеется пример как через сигналы обновлять аватарку пользователю, на мой взгляд, код служит только для примера.

Рубрики: [Программирование](#)

Теги: [Django](#), [Авторизация](#), [Социальные сети](#)

Подобається!

0

Tweet

0

27 февраля 2013 г.

13

[Share this!](#)

Комментарии

[vasa_c](#)

[27 февраля 2013 г. 15:46:05](#)

На uLogin'e есть не слишком приятная особенность. Я могу на самом улогине в своём профиле указать дефолтные параметры, в том числе мыло. И нихрена не проверяется. Могу, например, твоё указать. При авторизации же через фейсбук, я могу выбрать какие параметры я желаю пересылать. Например, отключить мыло. Тогда отключённые параметры будут браться из профиля на ulogin'e. И никак не проверить откуда пришло мыло, с фейсбука или с улогина. В итоге авторизуюсь на фейсбуке под своими данными, а бэкенду приходит, что я под твоими мылом зашёл.

Здесь с этим всё ок?

[adw0rd](#)

[27 февраля 2013 г. 17:33:53](#)

Да, uLogin откровенное гавно.

uLogin всё сильнее расстраивает... то у него в качестве пола от вконтакта "1" - это female, а от facebook "1" - male.

Дата дня рождения в убогом виде, строка "1.2.1970"

Через раз 500-я ошибка

Но самое убогое, пытался авторизоваться через ЖЖ, в качестве имени пользователя он вернул "Мастерская", а в качестве фамилии "интернет-разработчика", а никнейм вообще убил "asnov"

По поводу

Здесь с этим всё ок?

Да, с этим тут все окей, вообще никаких нареканий у меня! Текущие проблемы можно тут посмотреть <https://github.com/omab/django-social-auth/issues> и если что пофиксить самому. Я например вконтакте-бекенд фиксил когда-то. А вот как фиксить сторонний сервис (тот же uLogin) - хз, скорее всего, после подачи реквеста в саппорт - будут тянуть время.

[vasa_c](#)

[28 февраля 2013 г. 12:59:10](#)

оповещения о комментах на мыло у тебя не предусмотрено?

[adw0rd](#)

[28 февраля 2013 г. 13:22:12](#)

Не предусмотрено, [но я об этом думаю](#)

[adw0rd](#)

[10 марта 2013 г. 17:08:00](#)

Смотрите также <https://github.com/omab/python-social-auth>, от автора django-social-auth, вообще это тоже самое, но без Django. То есть можно использовать где угодно. Сам я еще не пользовался, но при необходимости воспользуюсь

12 марта 2013 г. 12:47:56

vasa_c

Запятые лезут в ссылки.

adw0rd

12 марта 2013 г. 13:30:35

Fixed, надо пофиксить будет еще парсер...

tim

1 апреля 2013 г. 21:20:21

Сделал как у вас написано, не получается что-то, хотя пару минут работало. <http://stackoverflow.com/questions/15747567/django-social-auth-dont-log-in>

adw0rd

1 апреля 2013 г. 22:56:10

Я смотрю вы уже решили свой вопрос

Oh, the error is in url(r'^\$', in urls. Only hardcore

noBrain

3 апреля 2013 г. 12:53:41

adw0rd, как обстоят дела со совместимостью с django 1.5.1? Если все нормально, то возникали трудности со взаимодействием с кастомной моделью пользователя?

adw0rd

3 апреля 2013 г. 13:44:45

Как раз сегодня планировал это проверить на проекте py3.3.1 + dj1.5.1

tim

5 апреля 2013 г. 14:41:28

Как избавиться от 500 ошибки? Авторизация проходит успешно (через вк). Но на странице предоставления прав приложения можно нажать "Разрешить" или "Отмена". При нажатии на "Отмена" вылетает AuthCanceled (<http://django-social-auth.readthedocs.org/en/v0.7.22/configuration.html#exceptions>). Почитал тут (<https://github.com/omab/django-social-auth/issues/302#issuecomment-4880887>), то есть надо дебаг false и

```
SOCIAL_AUTH_RAISE_EXCEPTIONS = False
LOGIN_URL = '/user/login/'
LOGIN_REDIRECT_URL = '/profile'
LOGIN_URL = '/login'
LOGIN_ERROR_URL = '/login-error'
```

Но все равно вылетает 500я. Как это можно зафиксировать?

Valeriy

7 апреля 2013 г. 1:33:01

Django 1.5.1, django-social-auth 0.7.22, VKontakteOAuth2Backend и FacebookBackend не перенаправляют на сайт после подтверждения прав (остается на странице подтверждения). Что делать?

adw0rd

14 апреля 2013 г. 17:10:13

tim, посмотрите <http://django-social-auth.readthedocs.org/en/latest/configuration.html#exceptions-middleware>

и в исходниках `social_auth/middleware.py`:

```
def raise_exception(self, request, exception):
    backend = self.backend
    return backend and \
        backend_setting(backend, 'SOCIAL_AUTH_RAISE_EXCEPTIONS') or \
        setting('DEBUG')
```

То есть, чтобы отключить вывод исключений надо помимо `SOCIAL_AUTH_RAISE_EXCEPTIONS=False` иметь `DEBUG=False`

[adw0rd](#)

[14 апреля 2013 г. 17:13:51](#)

noBrain,

adw0rd, как обстоят дела со совместимостью с django 1.5.1? Если все нормально, то возникали трудности со взаимодействием с кастомной моделью пользователя?

У меня на 1.5.1 все хорошо, при этом кастомный юзер и т.д.

[adw0rd](#)

[14 апреля 2013 г. 17:15:26](#)

Valeriy,

остается на странице подтверждения

то есть на странице бекенда? Тогда не знаю, смотрите настройки своих приложений в этих социалках, наверное что-то неправильно настроили, напримр адрес коллбек-страницы

[adw0rd](#)

[5 мая 2013 г. 16:40:38](#)

оповещения о комментах на мыло у тебя не предусмотрено?

сделал <https://github.com/adw0rd/marcus/issues/6>

Андрей

[12 мая 2013 г. 0:50:58](#)

django-social-auth не отображается в IE. Как-то можно пофиксить?

Андрей

[12 мая 2013 г. 0:54:31](#)

Имел ввиду django-social-auth-widget. Удобный, но в IE не работает

[adw0rd](#)

[12 мая 2013 г. 12:22:36](#)

У меня к сожалению нет IE под рукой, но люди говорят что работает

Valeriy

[18 мая 2013 г. 16:13:46](#)

то есть на странице бекенда? Тогда не знаю, смотрите настройки своих приложений в этих социалках, наверное что-то неправильно настроили, напримр адрес коллбек-страницы

adw0rd, а не могли бы вы скинуть готовый настроенный проект с БД SQLite, что бы можно было запустить и посмотреть на работу? Был бы вам очень благодарен.

[adw0rd](#)[18 мая 2013 г. 16:29:05](#)

К сожалению у меня не будет на это времени :(Вы лучше сами это сделайте и скажите что у вас не получается, и кстати в новом django-social-auth нет бекенда "VKontakteOAuth2Backend", теперь там "social_auth.backends.contrib.vk.VKOAuth2Backend".

И соответственно provider "vk-oauth".

ps. Обновил статью в связи с этими изменениями

[Den4uk](#)[24 мая 2013 г. 10:51:06](#)

Спасибо за статью, а кто подскажет где почитать, как дальше работать а точнее взять к примеру фото из соц. сети и подобную информацию??

[adw0rd](#)[24 мая 2013 г. 12:33:01](#)

Про фото я писал в этой же статье, раздел "Обновление аватарок при авторизации"

[adw0rd](#)[24 мая 2013 г. 12:36:20](#)

Далее надо читать API конкретной сети и использовать. Вот библиотеки которые могут вам помочь:

- Twitter - <https://pypi.python.org/pypi/tweepy/> (Недавно я добавил поддержку картинок в статусах <https://github.com/adw0rd/tweepy>)
- Facebook - <https://pypi.python.org/pypi/facebook-sdk>
- Google+ - <https://pypi.python.org/pypi/google-api-python-client>
- Vk - <https://pypi.python.org/pypi/vkontakte>

[Den4uk](#)[24 мая 2013 г. 14:24:36](#)

а через django-social-auth никто не цеплял фото(аватар), порылся в коде модуля там есть конфигурация типа

```
VK_DEFAULT_DATA = ['first_name', 'last_name', 'screen_name', 'nickname', 'photo']
```

только пока не понял как запихнуть ее в модель пользователя

[adw0rd](#)[24 мая 2013 г. 15:08:11](#)

Я про это вам уже и сказал. Обычно делается pipeline, который добавляется в SOCIAL_AUTH_PIPELINE и тот [код](#) будет подгружать фоточки.

VK_DEFAULT_DATA как и VK_EXTRA_DATA запросят если надо доп. прав и всунут в объект info.get('response') доп. данные, для разной сети они разные, могут быть такими: pic_big, userpic, user_photo и т.п.

[adw0rd](#)[24 мая 2013 г. 15:10:32](#)

Вот такой у меня черновик есть:

```
try:
    from urllib.request import urlopen
except ImportError:
    from urllib import urlopen

from django.core.files.storage import default_storage
from django.core.files.base import ContentFile

class UploadSocialAuthAvatar(object):

    def __init__(self, provider, info):
        self.provider = provider
        self.info = info
        self.image_url = None

    def vkontakte_oauth2(self):
        return self.info.get('user_photo')

    def twitter(self):
        image_url = self.info.get('profile_image_url')
        if not 'default_profile' in image_url:
            return image_url.replace('_normal', '_bigger')
        return None
```

```

def google_oauth2(self):
    return self.info.get('picture')

def yandex_oauth2(self):
    return self.info.get('userpic')

def facebook(self):
    return 'http://graph.facebook.com/{0}/picture?type=large'.format(
        self.info.get('id'))

def github(self):
    raise NotImplementedError('GitHub not implemented!')

def process(self):
    """Find and fetch image_url, and save to user.avatar
    """
    social_backend = self.provider.name.replace('-', '_')
    image_url = getattr(self, social_backend)()

    if not image_url:
        return None

    image_content = urlopen(image_url)
    if self.provider.name == 'facebook' and 'image/gif' in str(image_content.info()):
        return None

    return image_content

```

Его можете вызывать из pipeline

Den4uk

[27 мая 2013 г. 14:42:57](#)

Буду разбираться спасибо

Den4uk

[17 июня 2013 г. 21:10:57](#)

А как правильно его прописать в pipeline???

[adw0rd](#)

[18 июня 2013 г. 16:04:26](#)

Например, создаете модуль "users.pipeline":

```

from django.core.files.storage import default_storage
from django.core.files.base import ContentFile

def upload_avatar(request, *args, **kwargs):
    avatar_content = UploadSocialAuthAvatar(
        kwargs['backend'], kwargs['response']).process()
    avatar_name = default_storage.get_available_name(
        request.user.avatar.field.upload_to(
            request.user, "{0}.{1}".format(user.id, avatar_content.headers.subtype)))
    request.user.avatar.save(
        avatar_name, ContentFile(avatar_content.read()))
    request.user.save()

    return HttpResponseRedirect('/')

```

Потом в settings указываете:

```

SOCIAL_AUTH_PIPELINE = (
    'social_auth.backends.pipeline.social.social_auth_user',
    'social_auth.backends.pipeline.associate.associate_by_email',
    'social_auth.backends.pipeline.user.get_username',
    # 'social_auth.backends.pipeline.user.create_user',
    'social_auth.backends.pipeline.social.associate_user',
    'social_auth.backends.pipeline.social.load_extra_data',
    'social_auth.backends.pipeline.user.update_user_details',
    'social_auth.backends.pipeline.misc.save_status_to_session',
    'users.pipeline.upload_avatar',
)

```

Сочинил на ходу, не проверял.

Name or OpenID

Email for notifications

Text

Текст через пустую строку

Превращается в отдельные абзацы

> цитата
> цитата

- список
- список

курсив, **жирный**

some code()

4 пробела

in some language

Отправить на премодерацию