
django-withings Documentation

Release 0.0.1

Issac Kelly, Rebecca Lovewell, Dan Poirier, Percy Perez, Brad Pitc

September 05, 2014

1	Getting started	3
2	Settings	5
2.1	WITHINGS_CONSUMER_KEY	5
2.2	WITHINGS_CONSUMER_SECRET	5
2.3	WITHINGS_LOGIN_REDIRECT	5
2.4	WITHINGS_LOGOUT_REDIRECT	5
2.5	WITHINGS_SUBSCRIBE	5
2.6	WITHINGS_ERROR_TEMPLATE	6
2.7	WITHINGS_DECORATOR_MESSAGE	6
3	Views, decorators, and templates	7
4	Template tags	9
4.1	is_integrated_with_withings	9
5	Utilities	11
5.1	is_integrated	11
6	Links to related information	13
7	Release history	15
7.1	0.0.1	15
8	Indices and tables	17

Contents:

Getting started

1. Add *django-withings* to your Django site's requirements, however you prefer, and install it. It's installable from [PyPI](#).

2. Add *withingsapp* to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS += ['withingsapp']
```

3. Add the *django-withings* URLs to your `URLconf`:

```
url(r'^withings/', include('withingsapp.urls')),
```

3. Register your site at the [Withings developer site](#) to get a key and secret.

4. Add settings for `WITHINGS_CONSUMER_KEY` and `WITHINGS_CONSUMER_SECRET`:

```
WITHINGS_CONSUMER_KEY = 'abcdefg123456'  
WITHINGS_CONSUMER_SECRET = 'abcdefg123456'
```

5. If you need to change the defaults, add settings for `WITHINGS_LOGIN_REDIRECT`, `WITHINGS_LOGOUT_REDIRECT`, and/or `WITHINGS_ERROR_TEMPLATE`.

6. To display whether the user has integrated their Withings, or change a template behavior, use the `is_integrated_with_withings` template filter. Or in a view, call the `withingsapp.utils.is_integrated()` function. You can also use the decorator `withingsapp.decorators.withings_integration_warning()` to display a message to the user when they are not integrated with Withings.

7. To send the user through authorization at the Withings site for your app to access their data, send them to the `withingsapp.views.login()` view.

Settings

2.1 WITHINGS_CONSUMER_KEY

The key assigned to your app by Withings when you register your app at [the Withings developer site](#). You must specify a non-null value for this setting.

2.2 WITHINGS_CONSUMER_SECRET

The secret that goes with the WITHINGS_CONSUMER_KEY. You must specify a non-null value for this setting.

2.3 WITHINGS_LOGIN_REDIRECT

Default `''`

The URL which to redirect the user to after successful Withings integration, if no forwarding URL is given in the 'withings_next' session variable.

2.4 WITHINGS_LOGOUT_REDIRECT

Default `''`

The URL which to redirect the user to after removal of Withings account credentials, if no forwarding URL is given in the 'next' GET parameter.

2.5 WITHINGS_SUBSCRIBE

Default `False`

When this setting is True, we will subscribe to user data. Withings will send notifications when the data changes and we will queue tasks to get the updated data. When requests for withings data are made to withingsapp, we will always pull the latest data from our own database instead of getting it directly from Withings. To use this feature, you will need to setup a celery worker to handle the tasks.

Example celery configuration with rabbitmq as a backend:

```
WITHINGS_QUEUE = "withingsapp_get_data" CELERY_ROUTES = { 'with-  
ingsapp.tasks.update_withings_data_task':  
    { 'queue': WITHINGS_QUEUE } }  
BROKER_URL = "amqp://user:password@your_host:5672/your_vhost" import djcelery djcelery.setup_loader()
```

2.6 WITHINGS_ERROR_TEMPLATE

Default `'withings/error.html'`

The template used to report an error integrating the user's Withings.

2.7 WITHINGS_DECORATOR_MESSAGE

Default `'This page requires Withings integration.'`

The default message used by the `withingsapp.decorators.withings_integration_warning()` decorator to inform the user about Withings integration. If a callable is provided, it is called with the request as the only parameter to get the final value for the message.

Views, decorators, and templates

There are several views and decorators your site will use to drive Withings integration.

Template tags

4.1 `is_integrated_with_withings`

Utilities

5.1 is_integrated

Links to related information

- [Django-withings home](#)
- [Withings developer site](#)
- [Python-withings library](#)

Release history

7.1 0.0.1

Initial release.

Django-withings is a Django app for integrating a user's Withings data into your site.

It handles the details of getting your app authorized to access your user's Withings data via the Withings web API.

Indices and tables

- *genindex*
- *modindex*
- *search*