
django-websettings Documentation

Release 1.0

Hiroki KIYOHARA

December 16, 2013

Contents

1	Resources	3
2	At a glance	5
3	Contents	7
3.1	Getting Started	7
3.2	Advanced	10

Django application to provide a web interface to set a yet another django's settings.

If you want to learn about django-websettings, please read *[introduction of django-websettings](#)* first.

Resources

- Docs
- PyPI
- Code
- Testing

At a glance

Setting values:

```
# In websettings.py file
HTT_DRUM_PLAYER = 'Ritsu Tainaka'
HTT_BASS_PLAYER = 'Mio Akiyama'
```

Getting setting values:

```
>>> from websettings import websettings
>>> websettings.HTT_BASS_PLAYER
'Mio Akiyama'
```

On web interface:

WebSettings

HTT_DRUM_PLAYER

HTT_BASS_PLAYER

Changed the value:

WebSettings

HTT_DRUM_PLAYER

HTT_BASS_PLAYER

And then:

```
>>> from websettings import websettings
>>> websettings.HTT_BASS_PLAYER
'Jun Suzuki'
```

That's it! You like this and want to learn more, please read *introduction of django-websettings* first.

Contents

3.1 Getting Started

This documentation provide the basically usage of django-websettings.

3.1.1 Installation

Environment

django-websettings can be run on the 1.4, 1.5 of Django and 2.7, 3.2, 3.3 of CPython, 2.0 of PyPy.

If you want to use Django 1.4, you should choice Python 2.7, because Django 1.4 does not support Python 3.

Install

To install django-websettings, you should just install `django-websettings` on own environment.

```
$ pip install django-websettings
```

Websettings file

You shuoud put `websettings.py` file contains the setting keys and default values.

It recommended to put it under project directory. Let's put empty `websettings.py`

```
$ cd /path/to/yourproject/  
$ touch websettings.py
```

And then, you should tell the position of `websettings.py` by specifying a `WEBSETTINGS_MODULE` value in your project's `settings.py` file:

```
# In your settings.py file.  
WEBSETTINGS_MODULE = 'yourproject.websettings'
```

And add line to `INSTALLED_APPS` to your `settings.py`:

```
INSTALLED_APPS = (  
    ...  
    'websettings',  
    ...  
)
```

Then sync to your DB:

```
$ python manage.py syncdb
```

Web interface URL

The url to web interface to set setting values can be included like this:

```
# In your urls.py file.  
url(r'^websettings/', include('websettings.urls'))
```

The admin user can only access this page.

django-websettings does not provide the login and logout page, so it is also necessary to provide them.

3.1.2 Tutorial

Now, you can use some great features provided by django-websettings.

Setting websettings

You can write values your websettings.py, like this:

```
# In your websettings.py  
HTT_DRUM_PLAYER = 'Ritsu Tainaka'  
HTT_BASS_PLAYER = 'Mio Akiyama'
```

Please be careful in following:

- Values can accept only strings
- All of these keys must explain by UPPER_CASE

And then, you can get these values from websettings.websettings module. You can try getting it in your Django shell:

```
>>> from websettings import websettings  
>>> websettings.HTT_DRUM_PLAYER  
'Ritsu Tainaka'  
>>> websettings.HTT_BASS_PLAYER  
'Mio Akiyama'
```

Using Web interface

Now, you can change these setting values from web interface generated automatically.

Let's access to the URL `/websettings/` after logged in by admin user. you will see a page, like this:

WebSettings

HTT_DRUM_PLAYER

HTT_BASS_PLAYER

- It provides a list of settings you wrote in websettings.py.
- Value of setting is handled as default value of each forms.
- The value accept only string up to 255 character.

And then, you can change the values by using it. If you want to fire Mio and hire Jun as bassist, you can click `Edit` button and change the value.

WebSettings

HTT_DRUM_PLAYER

HTT_BASS_PLAYER

Ok, and submit the form. Let's try getting `HTT_BASS_PLAYER` value again in shell:

```
>>> websettings.HTT_BASS_PLAYER
'Jun Suzuki'
```

Yay!

3.1.3 Conclusion

That's basic usage and feature of `django-websettings`.

You set default values, use setting values, then fix values from web interface if you feel like it.

3.2 Advanced

3.2.1 Printing correct value of websettings

You can check the current value of settings by using `print_websettings` management command:

```
$ python manage.py print_websettings
HTT_DRUM_PLAYER      = 'Ritsu Tainaka'
HTT_BASS_PLAYER      = 'Mio Akiyama'
```

Also you can use these keyword arguments:

- `format`:
 - `json`: printing result as json.
 - `simple`: printing result as simple format like assignment of python.
- `indent`: accept a number to specify indent size (only available as json format)

3.2.2 Deleting trash values

There is an anxiety leaving trash data on backend:

```
>>> from websettigs import websettings
>>> websettings.clear_trash()
```

3.2.3 Writing own backend

django-websettings is designed to correspond to multiple backend to store settings.

This guide explain the way to write your own backend and use it.

Interface guide for backend

A backend is a just one module contains three functions.

- `getsetting(key)`: a function to get the setting value by *key*.
- `setsetting(key, value)`: a function to set *value* correspond to *key*.
- `exclude_clear(keys)`: a function to clear setting values exclude *keys*.

When fail getting or setting the values, the function should raise `AttributeError`.

Using your backend

You can specify the back end module to set a `WEBSETTINGS_BACKEND` in your project's `setting.py` file:

```
# In your settings.py file
WEBSETTINGS_BACKEND = 'path.to.backend_module'
```

A DB backend (`websetting.backends.db`) is used by default.