# Django OpenID Provider Documentation

***Release v0.6***

**Roman Barczyński**

September 02, 2016

## Contents

# 1 Introduction

Django OpenID Provider application acts as OpenID provider (in lame terms Server) for your *django.contrib.auth* accounts.

If you have your own Django powered website you might want to use your admin account to authenticate on other sites like stackoverflow or even in other django websites you manage (with django_openid_auth for example).

Reasons to do that:

- forget about dozens of passwords on different sites,
- manage and revoke OpenID authorization for sites you log on,
- http://openidexplained.com/

## 1.1 Features

- Automatic redirect to login page for unauthorized users.
- Semi-automated creation of OpenID identifiers (leave OpenID field empty).
- Decision page for adding trust_root to one's OpenID trusted sites.

## 1.2 New releases and bug reporting

Code and issue tracker is hosted on bitbucket.org/romke/django_openid_provider/

# 2 Requirements

The Python OpenID library is required to run `openid_provider`. By default it'll use `openid.store.filestore.FileOpenIDStore` for persistent storage of OpenID related data.

To change the file system location that the default storage uses, you can optionally provide a `OPENID_PROVIDER_FILESTORE_PATH` setting.

In case you don't want to store the OpenID related data on a file system, it's also possible to make use of the `DjangoOpenIDStore` contained in the django_openid_auth app. Simply add an `OPENID_PROVIDER_STORE` setting to your settings:

```
OPENID_PROVIDER_STORE = 'django_openid_auth.store.DjangoOpenIDStore'
```

This is especially useful in case your site is deployed in shared hosting environments.

# 3 Basic Installation

1. Copy/install `openid_provider` into your project directory (or link to it).
2. Add `'openid_provider'` to `INSTALLED_APPS` and its dependencies:

```
    INSTALLED_APPS = (
        # ...
        'django.contrib.auth',
        'django.contrib.sessions',
```

```
        'openid_provider',
        # ...
    )
```

3. Add `openid_provider/urls.py` to your urlpatterns, e.g.:

```
    urlpatterns = patterns('',
        # ...
        url(r'^openid/', include('openid_provider.urls')),
        # ...
    )
```

4. To create required tables in your database, run:

```
    python manage.py syncdb
```

   or (if you are using south):

```
    python manage.py migrate openid_provider
```

# 4 Upgrading

## 4.1 From <= 0.5

If you are using south you must fake initial migration:

```
python manage.py migrate openid_provider 0001 --fake
```

# 5 What is not provided

This application does not include most basic template every django project should have: `base.html`. You should have `base.html` file in one of your settings.TEMPLATE_DIRS directories and it should contain 3 base blocks:

- title
- extrahead
- content

(see DosAndDontsForApplicationWriters and django template inheritance)

If your base template is named differently you should override `openid_provider/base.html` to contain something like:

```
{% extends "your_base_template_name.html" %}
```

If your base template have different blocks you could easily remap those:

```
{% block your_content_block_name %}{% block content %}{% endblock %}{% endblock %}
```

# 6 Usage

After instalation and database sync login to your django admin site. You'll notice new application and it's model `OpenIDs`.

Click *add* and at least select `User` from select box. If you want that user to have human-friendly openid identifier enter it into `Openid` field. Hit *Save* and your first OpenID is ready.

Your OpenID Claimed Identifier URL will be:

```
<your host>/<your openid base url>/<contents of Openid field>/
```

When you will use it to login on the web, Relaying Party [1] will redirect you to your site. If you aren't logged in on your site you'll first have to log in, then you'll be asked if you want to add Relaying Party to your "Trusted Roots" [2]. Add it and next time you'll log in it'll be automatic - you won't even notice your site appearance if you'll be still logged in.

## 6.1 Customization

You can grant your users way to customize their OpenID by preparing form or you can automate it's creation.

After creating user you can:

```
from django.contrib.auth.models import User
user = User.objects.get(pk=...)
# create openid for that user:
user.openid_set.create(openid=user.username)
```

## 6.2 Revoking Relaying Party authorization

At any time you can revoke any Relaying Party authorization to use your OpenID by removing it from "Trusted Roots" list associated with your OpenID:

```
# note that there can be few openids associated with each user
openid = OpenID.objects.filter(user=request.user, default=True)[0]

# remove all relaying parties with google.com in URL
openid.trustedroot_set.filter(trust_root__contains='google.com').delete()
```

# 7 History

## 7.1 v0.6

Released 2014-03-31.

- When pressing "No" on decide page authentication is now cancelled as it supposed to be, fixes #6.
- Added South migrations

  **Warning:** you must fake initial migration if you already have working openid_provider installation.

- Django 1.6 compatibility (use openid toPostArgs/decodeRequest internal serializer).

[1] that's fancy name for site where you want to login via OpenID.
[2] list of sites authorized to use your OpenID as login information.

### 7.2 v0.5

Released 2013-12-23.

- Security enchancement: OpenID Authentication 2.0 9.2.1 was implemented, fixes #4.
- Fixed landing page view to handle redirect URL GET params correctly.
- Added OPENID_PROVIDER_SREG_DATA_CALLBACK setting for custom SREG callback functions.
- Added AX support, new OPENID_PROVIDER_AX_DATA_CALLBACK callback.
- Added Django 1.5 support.
- Added Django 1.5 custom user model support.
- Fixed response page javascript to submit the correct form when there is more than one.

### 7.3 v0.4

Released 2010-12-30.

- CSRF enabled sites support (thx to Jannis).
- SREG support (thx to Jannis).

### 7.4 v0.3

Released 2010-06-08.

- Better Djangos DosAndDontsForApplicationWriters compliance.
- Added MANIFEST.in and setup.py (thx Bruno).

### 7.5 v0.2

Released 2010-02-03.

- Better support for multiple identities - you can select one as default (if none identity is claimed in request).
- Added checking of claimed identity ownership.
- Added documentation.

### 7.6 v0.1

Released 2009-11-23.

- Initial version.

## 8 Credits

Django OpenID Provider is developed by Roman Barczyński based on code from:

- simon (django code snippets)
- python-openid-2.2.4 examples/djopenid

## 8.1 Contributors

Bruno Renié

- initial setup.py and MANIFEST.in

Jannis Leidel

- code cleanup (tab2spaces, PEP8),
- sreg support,
- CSRF exempt for openid_server,
- simplified host resolution,
- ability to specify OPENID_PROVIDER_STORE in settings.

# 9 LICENSE