
Django Rosetta Documentation

Release 0.7.13

Marco Bonetti

Apr 07, 2017

Contents

1	Features	3
2	Contents:	5
2.1	Installation	5
2.2	Settings	6
2.3	Usage	7
2.4	Version History	10

Rosetta is a [Django](#) application that eases the translation process of your Django projects.

Because it doesn't export any models, Rosetta doesn't create any tables in your project's database. Rosetta can be installed and uninstalled by simply adding and removing a single entry in your project's *INSTALLED_APPS* and a single line in your main `urls.py` file.

CHAPTER 1

Features

- Database independent
- Reads and writes your project's *gettext* catalogs (po and mo files)
- Installed and uninstalled in under a minute
- Uses Django's admin interface CSS

Installation

Requirements

- As of version 0.7.13, Rosetta supports Django 1.8 through 1.11.

Install Rosetta

1. `pip install django-rosetta`
2. Add 'rosetta' to the `INSTALLED_APPS` in your project's `settings.py`
3. Add an URL entry to your project's `urls.py`, for example:

```
from django.conf import settings

if 'rosetta' in settings.INSTALLED_APPS:
    urlpatterns += patterns('',
        url(r'^rosetta/', include('rosetta.urls')),
    )
```

Note: you can use whatever you wish as the URL prefix.

To uninstall Rosetta, simply comment out or remove the 'rosetta' line in your `INSTALLED_APPS`

Testing

```
pip install tox && tox
```

Security

Because Rosetta requires write access to some of the files in your Django project, access to the application is restricted to the administrator user only (as defined in your project's Admin interface)

If you wish to grant editing access to other users:

1. Create a *'translators'* group in your admin interface
2. Add the user you wish to grant translating rights to this group

Settings

Rosetta can be configured via the following parameters, to be defined in your project settings file:

- `ROSETTA_MESSAGES_PER_PAGE`: Number of messages to display per page. Defaults to 10.
- `ROSETTA_ENABLE_TRANSLATION_SUGGESTIONS`: Enable AJAX translation suggestions. Defaults to `False`.
- `YANDEX_TRANSLATE_KEY`: Translation suggestions from Yandex [Yandex.Translate API](#). To use this service you must first [obtain an AppID key](#), then specify the key here. Defaults to `None`.
- `AZURE_CLIENT_ID` and `AZURE_CLIENT_SECRET`: Translation suggestions using the Microsoft Azure API. To use this service, you must first [register for the service](#), then specify the 'Customer ID' and 'Primary Account Key' respectively, which you can find on your [account information page on Azure Marketplace](#).
- `ROSETTA_MESSAGES_SOURCE_LANGUAGE_CODE` and `ROSETTA_MESSAGES_SOURCE_LANGUAGE_NAME`: Change these if the source language in your PO files isn't English. Default to `'en'` and `'English'` respectively.
- `ROSETTA_WSGI_AUTO_RELOAD` and `ROSETTA_UWSGI_AUTO_RELOAD`: When running WSGI daemon mode, using `mod_wsgi 2.0c5` or later, this setting controls whether the contents of the gettext catalog files should be automatically reloaded by the WSGI processes each time they are modified. For performance reasons, this setting should be disabled in production environments. Default to `False`.
- `ROSETTA_EXCLUDED_APPLICATIONS`: Exclude applications defined in this list from being translated. Defaults to `()`.
- `ROSETTA_REQUIRES_AUTH`: Require authentication for all Rosetta views. Defaults to `True`.
- `ROSETTA_POFILE_WRAP_WIDTH`: Sets the line-length of the edited PO file. Set this to 0 to mimic `makemessage's --no-wrap` option. Defaults to 78.
- `ROSETTA_STORAGE_CLASS`: See the note below on Storages. Defaults to `rosetta.storage.CacheRosettaStorage`
- `ROSETTA_ACCESS_CONTROL_FUNCTION`: An alternative function that determines if a given user can access the translation views. This function receives a `user` as its argument, and returns a boolean specifying whether the passed user is allowed to use Rosetta or not.
- `ROSETTA_LANGUAGE_GROUPS`: Set to `True` to enable language-specific groups, which can be used to give different translators access to different languages. Instead of creating a global `translators` group, create individual per-language groups, e.g. `translators-de`, `translators-fr`, and assign users to these.
- `ROSETTA_CACHE_NAME`: When using `rosetta.storage.CacheRosettaStorage`, you can store the Rosetta data in a specific cache. This is particularly useful when your default cache is a `django.core.cache.backends.dummy.DummyCache` (which happens on pre-production environments). If unset, it will default to `rosetta` if a cache with this name exists, or `default` if not.

- `ROSETTA_POFILENAMES`: Defines which po file names are exposed in the web interface. Defaults to `('django.po', 'djangojs.po')`
- `ROSETTA_EXCLUDED_PATHS`: Exclude paths defined in this list from being searched (usually ends with “locale”). Defaults to `()`
- `ROSETTA_AUTO_COMPILE`: Determines whether the MO file is automatically compiled when the PO file is saved. Defaults to `True`.
- `ROSETTA_ENABLE_REFLANG`: Enables a selector for picking a reference language other than English. Defaults to `False`.
- `ROSETTA_SHOW_AT_ADMIN_PANEL`: Adds a handy link to Rosetta at the bottom of the Django admin apps index. Defaults to `False`.

Storages

To prevent re-reading and parsing the PO file catalogs over and over again, Rosetta stores them in a volatile location. This can be either the HTTP session or the Django cache.

Django 1.4 has introduced a signed cookie session backend, which stores the whole content of the session in an encrypted cookie. Unfortunately this doesn't work with large PO files, as the limit of 4096 chars that can be stored in a cookie is easily exceeded.

In this case the Cache-based backend should be used (by setting `ROSETTA_STORAGE_CLASS = 'rosetta.storage.CacheRosettaStorage'`). Please make sure that a proper `CACHES` backend is configured in your Django settings if your Django app is being served in a multi-process environment, or the different server processes, serving subsequent requests, won't find the storage data left by previous requests.

Alternatively you can switch back to using the Session based storage by setting `ROSETTA_STORAGE_CLASS = 'rosetta.storage.SessionRosettaStorage'` in your settings. This is perfectly safe on Django 1.3. On Django 1.4 or higher make sure you have `DON'T` use the `signed_cookies` `SESSION_BACKEND` with this Rosetta storage backend or funky things might happen.

TL;DR: if you run Django with `gunicorn`, `mod-wsgi` or other multi-process environment, the Django-default `CACHES LocMemCache` backend won't suffice: use `memcache` instead, or you will run into issues.

Usage

Generate a batch of files to translate

See [Django's documentation on Internationalization](#) to set up your project to use `i18n` and create the `gettext` catalog files.

Translate away!

Start your Django development server and point your browser to the URL prefix you have chosen during the installation process. You will get to the file selection window.

Rosetta

Home > Language selection

English				
File	Messages	Translated	Obsolete	Progress
/Volumes/woot/Users/mbi/Code/momentn/trunk/media/locale/en/LC_MESSAGES/djangojs.po	2	0	0	0.00%
French				
File	Messages	Translated	Obsolete	Progress
/Volumes/woot/Users/mbi/Code/momentn/trunk/momentn/locale/fr/LC_MESSAGES/django.po	250	9	0	3.00%
/Volumes/woot/Users/mbi/Code/momentn/trunk/media/locale/fr/LC_MESSAGES/djangojs.po	2	0	0	0.00%
German				
File	Messages	Translated	Obsolete	Progress
/Volumes/woot/Users/mbi/Code/momentn/trunk/momentn/locale/de/LC_MESSAGES/django.po	250	1	0	0.00%
/Volumes/woot/Users/mbi/Code/momentn/trunk/media/locale/de/LC_MESSAGES/djangojs.po	2	0	0	0.00%
Dutch				
File	Messages	Translated	Obsolete	Progress
/Volumes/woot/Users/mbi/Code/momentn/trunk/momentn/locale/nl/LC_MESSAGES/django.po	250	3	0	1.00%
/Volumes/woot/Users/mbi/Code/momentn/trunk/media/locale/nl/LC_MESSAGES/djangojs.po	2	0	0	0.00%

Select a file and translate each untranslated message. Whenever a new batch of messages is processed, Rosetta updates the corresponding *django.po* file and regenerates the corresponding `mo` file.

This means your project's labels will be translated right away, unfortunately you'll still have to restart the web server for the changes to take effect. (NEW: if your web server supports it, you can force auto-reloading of the translated catalog whenever a change was saved. See the note regarding the `ROSETTA_WSGI_AUTO_RELOAD` variable in `conf/settings.py`.)

If the webserver doesn't have write access on the catalog files (as shown in the screen shot below) an archive of the catalog files can be downloaded.

Rosetta
Pick another file / Download this catalog

Home > French > /Users/marco/Code/rosetta/locale/fr/LC_MESSAGES/django.po > Progress: 100.00%
⚠ File is read-only: download the file when done editing!

Translate into French

Display: Untranslated only
Translated only
Both

English	French	Occurrences(s)
Home	Racine	templates/rosetta/languages.html:2 templates/rosetta/pofile.html:13
Language selection	Sélection de la langue	templates/rosetta/languages.html:2
File	Fichier	templates/rosetta/languages.html:12
Messages	Messages	templates/rosetta/languages.html:13
Translated	Traduits	templates/rosetta/languages.html:14
Obsolete	Obsolètes	templates/rosetta/languages.html:15
Progress	Progression	templates/rosetta/languages.html:16
Pick another file	Choisir un autre fichier	templates/rosetta/pofile.html:7
Download this catalog	Télécharger ce catalogue	templates/rosetta/pofile.html:8
Progress: %(percent_translated)s %	Progression: %(percent_translated)s%	templates/rosetta/pofile.html:16

Skip to page: 1 2 3 21/21 messages
Save and translate next block

Translating Rosetta itself

By default Rosetta hides its own catalog files in the file selection interface (shown above.) If you would like to translate Rosetta to your own language:

1. Create a subdirectory for your locale inside Rosetta's locale directory, e.g. `rosetta/locale/XX/LC_MESSAGES`
2. Instruct Django to create the initial catalog, by running `django-admin.py makemessages -l XX` inside Rosetta's directory (refer to [Django's documentation on i18n](#) for details)
3. Instruct Rosetta to look for its own catalogs, by appending `?rosetta` to the language selection page's URL, e.g. `http://127.0.0.1:8000/rosetta/pick/?rosetta`
4. Translate as usual
5. Send a pull request if you feel like sharing

Acknowledgments

- Rosetta uses the excellent [polib](#) library to parse and handle gettext files.

Version History

Version 0.7.13

- Search in comments, too (PR #174, thanks @torchingloom)
- Added `ROSETTA_SHOW_AT_ADMIN_PANEL` setting to display add a link to Rosetta from the admin app index page. (PR #176, thanks @scream4ik)
- Test against Django 1.11
- Template cleanup (Issue #181, thanks @Ecno92)

Version 0.7.12

- Fix `IndexError` in `fix_nls` when translation is just a carriage return (PR #168, thanks @nealtodd)
- Remove float formatting of integer percent translated (PR #171, thanks @nealtodd)
- Fixed a comment (PR #170, thanks @dnaranjo89)
- Test against Django 1.10
- Dropped support for goslate and the Google translate API

Version 0.7.11

- Make MO file compilation optional (PR #166, Issue #155, thanks @nealtodd)
- Fix an invalid page get parameter by falling back to page 1 (PR #165, thanks @nealtodd)
- Adds reference language selector (PR #60, thanks @hsoft)

Version 0.7.10

- Re-released 0.7.9 to include a missing image (Issue #162, thanks @legios89)

Version 0.7.9

- Use language code without country specification for Yandex dest lang (PR #152, thanks @nealtodd)
- Support discovering locale directories like `zh_Hans(xx_Xxxx)` (Fixes #133 via PR #153 and #133, thanks @glasslion and @dohsimpson)
- Ship Django's original search icon as a static asset (Fixes #157, thanks @facconi)
- Added a warning about translation via the Google Translate service being deprecated in the next version

Version 0.7.8

- Adds missing includes in `MANIFEST.in`
- Support for running tests via `setuptools`
- Updated `microsoftranslator` dependency version

Version 0.7.7

- Supported Django versions are now 1.7, 1.8 and 1.9
- Added proper documentation
- Fixed typo in documentation (PR #130, thanks @dfrdmn)
- Fixes the Fuzzy toggle link by adding an actual toggle checkbox (Issue #132, thanks @EmilStenstrom)
- Better handling of Custom User Models while checking whether the current User is authorized to translate (Issue #131, thanks @EmilStenstrom)
- Include the testproject in the sdist tarball to allow Debian to run tests during installation (Issue #137, thanks @fladi)
- Display an explicit error message to the enduser when saving the POfile fails for some reason (Issue #135, thanks @pgcd)
- Added support for PEP 3101 string formatting (PR #140, thanks @adamjforster)
- Added support for composite locales, e.g. 'bs-Cyrl-BA' (Issue #142, thanks @felarov)
- Fixed a misplaced CSRF token (PR #145, thanks @pajod)

Version 0.7.6

- Added support for the Free Google Translate API (PR #117, thanks @cuchac)
- Probable fix for apps defined by their AppConfig causing havoc in Django 1.7 and later (Issues #113 and #125)
- Test configuration improved to test against Django 1.8 beta 1 and Django 1.7.5
- Require polib >= 1.0.6 (PR #127, thanks @NotSqrt)
- Test against Django 1.8 final

Version 0.7.5

- Fixed external JavaScript import to be url scheme independent (PR #101, thanks @tsouvarev)
- Fixed a test
- Added support for excluding certain locale paths from the list of PO catalogs (PR #102, thanks @elpaso)
- Added support for translator groups (PR #103, thanks @barklund)
- Removed Microsoft Translator as a shipped lib, relying on an external version instead
- Improved the app loading mechanism to cope with Django 1.7's new AppConfig (thanks @artscoop)
- Fixed a couple inconsistencies in the German translation. (thanks @benebun)
- Use content_type instead of mimetype in HttpResponse. (Issue #115, thanks @vesteinn)
- Don't assume that request.user has settable properties, this was a silly idea anyway (Issue #114, thanks @steve-jalim)
- Preserve HTML code when receiving translations from the Yandex translation service (Issue #116, thanks @marcbelmont)
- Use TOX for testing
- Test against Django 1.8a

Version 0.7.4

- New ROSETTA_POFILENAMES setting. (PR #44, thanks @wrboyce)
- Updated Czech translation (#97, #99 thanks @cuchac)
- Fixed gettext standard compliance of all shipped translations
- No longer ship polib, rely on the Cheeseshop instead

Version 0.7.3

- Fix for test settings leaking onto global settings: LANGUAGES was overridden and not set back (Issue #81 - Thanks @zsoldosp)
- Test against Django 1.6.1
- Missing context variable in catalog list (Issue #87 - Thanks @kunitoki)
- Added support for Yandex translation API (Issue #89 - Thanks @BlackWizard) See supported languages and limitations here: <https://github.com/mbi/django-rosetta/pull/89>
- Added support for the Azure translation API, replacing the BING API. (Issue #86, thanks @davidkuchar and @maikelwever)
- Removed support for the signed_cookies SESSION_ENGINE + SessionRosettaStorage in Django 1.6, because serialization of POFiles would fail
- Simplified the group membership test (Issue #90 - Thanks @dotsbb)
- Better serving of admin static files. (Issue #61, thanks @tback)
- Dropped Django 1.3 support

Version 0.7.2

- Fix for when settings imports unicode_literals for some reason (Issue #67)
- Fixed mess with app_id between pages (Issue #68, thanks @tsouvarev)
- Added Farsi translation. Thanks, @amiraliakbari
- Improved the permission system, allowing for more advanced permission mechanisms. Thanks, @codeinthehole and @tangentlabs
- Fixed the ordering of apps in the language selection screen. (Issue #73, thanks @tsouvarev, @kanu and everyone else involved in tracking this one down)
- Support for complex locale names. (Issue #71, Thanks @strycore)
- Configurable cache name (Issue #75, Thanks @Karmak23)

Version 0.7.1

- Fix: value missing in context

Version 0.7.0

- Support for Django 1.5 and HEAD, support for Python 3.
- Upgraded bundled polib to version 1.0.3 - <http://pypi.python.org/pypi/polib/1.0.3>
- Support timezones on the last modified PO header. Thanks @jmoiron (Issue #43)
- Actually move to the next block when submitting a lot of translations (Issue #13)
- Add msgctxt to the entry hash to differentiate entries with context. Thanks @metalpriest (Issue #39)
- Better discovery of locale files on Django 1.4+ Thanks @tjjs (Issues #63, #64)
- List apps in alphabetical order

Version 0.6.8

- Switched to a pluggable storage backend model to increase compatibility with Django 1.4. Cache and Session-based storages are provided.

Version 0.6.7

- Added a testproject to run tests
- Updated french translation. Thanks, @BertrandBordage
- Merged @sleepyjames' PR that fixes an error when pofile save path contains '.po' in the path
- Merged @rory's PR to correctly handle plural strings that have a leading/trailing newline (Issue #34)

Version 0.6.6

- Django 1.4 support (Issue #30, #33)
- Better handling of translation callbacks on Bing's translation API and support of composite locales (Issue #26)

Version 0.6.5

- Updated polib to 0.7.0
- Added ROSETTA_POFILE_WRAP_WIDTH setting to track the line-length of the updated Po file. (Issue #24)
- Renamed the `messages` `context` variable to `` `rosetta_messages` prevent conflicts with `django.contrib.messages` (Issue #23)

Version 0.6.4

- Added ROSETTA_REQUIRES_AUTH option to grant access to non authenticated users (False by default)

Version 0.6.3

- Support for the Bing translation API service to replace Google's service which is no longer free.