

---

# **Tower**

***Release 0.3.4***

July 14, 2016



|          |                                   |           |
|----------|-----------------------------------|-----------|
| <b>1</b> | <b>Requirements</b>               | <b>3</b>  |
| <b>2</b> | <b>Installation</b>               | <b>5</b>  |
| <b>3</b> | <b>Configure</b>                  | <b>7</b>  |
| <b>4</b> | <b>Usage</b>                      | <b>9</b>  |
| <b>5</b> | <b>A note on whitespace</b>       | <b>11</b> |
| <b>6</b> | <b>A note on <i>safe</i>-ness</b> | <b>13</b> |
| <b>7</b> | <b>Run tests</b>                  | <b>15</b> |



Tower provides some additional functionality to Jinja and it's i18n extension, and the Babel library.

- Author: Wil Clouser and [contributors](#)
- Licence: BSD
- Compatibility: Python 2.6 and 2.7, Django 1.4, 1.5 and 1.6
- Requirements: django, babel, jinja2, jingo and translate-toolkit
- Project URL: <https://github.com/clouserw/tower>
- Documentation: <http://tower.readthedocs.org/en/latest/>

Specifics:

- Pulls strings from a variety of sources: Python, JavaScript, and .lhtml files.
- Collapses whitespace in all strings to prevent unwieldy msgids.
- Supports Gettext context (msgctxt) in all gettext, and ngettext calls.
- Supports merging PHP and Python .pot files. This is temporary. If you want to support that for an extended time look at phppo2pypo in the [Translate Toolkit](#).



---

## Requirements

---

- Django
- Babel
- Jinja2 and Jingo
- translate-toolkit

See [requirements.txt](#) for details.



---

## Installation

---

Install from pypi with:

```
pip install tower
```

Source code is at <http://github.com/clouserw/tower>.

Install from GitHub with:

```
pip install -e git://github.com/clouserw/tower.git#egg=tower
```



---

## Configure

---

Add to `INSTALLED_APPS` in your Django settings file:

```
INSTALLED_APPS = (
    # ...
    'tower',
    # ...
)
```

Then configure.

`django.conf.settings.DOMAIN_METHODS`

**Default:** None—you must set this.

The value is a dict of domain to file spec and extraction method tuples.

For example, this creates a domain “messages” and in that domain extracts all the l10n strings from `.py` and `.html` files:

```
DOMAIN_METHODS = {
    'messages': [
        ('fjord/**/*.py', 'tower.extract_tower_python'),
        ('fjord/**/*.html', 'tower.extract_tower_template'),
    ]
}
```

Use `tower.extract_tower_python` for Python files and `tower.extract_tower_template` for Jinja2 templates.

The l10n strings will be saved in a `.pot` file with the name of the domain. In the above example, it’d be `messages.pot`.

`django.conf.settings.STANDALONE_DOMAINS`

**Default:** `['messages']`

By default, all domains specified in `DOMAIN_METHODS` get merged into one big `.pot` file. If you don’t want that, you can specify which domains get their own `.pot` file with `STANDALONE_DOMAINS`.

`django.conf.settings.TOWER_KEYWORDS`

Gettext keywords are the actual names of the functions your application uses to denote strings which should be extracted. Babel defines a sane default list in `DEFAULT_KEYWORDS` which includes the regulars like `_()`, `gettext()`, `ngettext()`, etc. If your app uses different functions you’ll need to define them here. This setting is a dictionary and the value follows [gettext’s keywordspec](#).

If you set this, it is *merged* with Babel’s `DEFAULT_KEYWORDS` setting.

Example:

```
# We lazy load strings in our classes so we can cache them and they are
# still localized in the right language for the end user. This will let
# us use _lazy('Localized string').
TOWER_KEYWORDS = {
    '_lazy': None,
}
```

`django.conf.settings.TOWER_ADD_HEADERS`

**Default:** False

If you have trouble extracting strings with Tower, try setting this to True.

`django.conf.settings.ROOT`

This points to the source code directory where you want your `locale/` directory to be.

`django.conf.settings.path`

This is a function that takes arbitrary set of args and combines them with `ROOT` to form a new path.

Example:

```
import os

# The settings file is in fjord/fjord/settings/base.py. From
# base, up two directories is the initial fjord directory
# which is where all the source code and the fjord Django
# project are.
ROOT = os.path.dirname(os.path.dirname(__file__))

path = lambda *args: os.path.abspath(os.path.join(ROOT, *args))
```

`django.conf.settings.TOWER_INSTALL_JINJA_TRANSLATIONS`

**Default:** True

By default tower will ensure it's `gettext` and `ngettext` functions are installed into Jinja2 on every call to `tower.activate()`. You likely want this, but if you need to provide your own `gettext` and `ngettext` functions, set this to `False`, and in your project call `jingo.env.install_gettext_translations` or `jingo.env.install_gettext_callables`.

---

**Usage**

---

**Extract:**

```
./manage.py extract
```

**Merge:**

```
./manage.py merge
```



---

## A note on whitespace

---

When tower extracts strings, it collapses whitespace. This makes it easier for localizers. It also means you need to use `ugettext`, `ungettext`, `ugettext_lazy` and `ungettext_lazy` from tower. Otherwise the msgids being passed in won't have their whitespace stripped and thus won't match anything in your .mo file.



---

## A note on safe-ness

---

L10n strings are marked “safe” for Jinja2 automatically, so they will not be HTML-escaped:

```
{{ _('Hello <strong>World</strong>') }}
```

This works as expected. When interpolating into an L10n string, however, it will be marked as “unsafe” and escaped, unless you use jingo’s `|fe()` helper (which will escape the arguments but not the string they are interpolated into). Like this:

```
{{ _('Hello <strong>{0}</strong>') |fe(user.nickname) }}
```



---

## Run tests

---

Run:

```
python run_tests.py
```

To test on all supported versions of Python and Django:

```
$ pip install tox  
$ tox
```