django-dbtemplates Documentation

Release dev

Jannis Leidel and contributors

Jan 27, 2019

Contents

`dbtemplates` is a Django app that consists of two parts:

1. It allows you to store templates in your database
2. It provides template loader that enables Django to load the templates from the database

It also features optional support for *versioned storage* and *django-admin command*, integrates with Django's *caching system* and the *admin actions*.

Please see https://django-dbtemplates.readthedocs.io/ for more details.

The source code and issue tracker can be found on Github: https://github.com/jazzband/django-dbtemplates

Contents:

CHAPTER 1

Setup

1. Get the source from the Git repository or install it from the Python Package Index by running `pip install django-dbtemplates`.
2. Follow the instructions in the INSTALL file
3. Edit the settings.py of your Django site:
   - Add `dbtemplates` to the `INSTALLED_APPS` setting

     Check if `django.contrib.sites` and `django.contrib.admin` are in `INSTALLED_APPS` and add if necessary.

     It should look something like this:

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.admin',
    'django.contrib.flatpages',
    # ..
    'dbtemplates',
)
```

   - Add `dbtemplates.loader.Loader` to the `TEMPLATES.OPTIONS.loaders` list in the settings.py of your Django project.

     It should look something like this:

```
TEMPLATES = [
  {
      'BACKEND': 'django.template.backends.django.DjangoTemplates',
      'DIRS': [ # your template dirs here
      ],
      'APP_DIRS': False,
```

(continues on next page)

(continued from previous page)

```
    'OPTIONS': {
        'context_processors': [
            'django.contrib.auth.context_processors.auth',
            'django.template.context_processors.debug',
            'django.template.context_processors.i18n',
            'django.template.context_processors.media',
            'django.template.context_processors.static',
            'django.template.context_processors.tz',
            'django.contrib.messages.context_processors.messages',
            'django.template.context_processors.request',
        ],
        'loaders': [
            'django.template.loaders.filesystem.Loader',
            'django.template.loaders.app_directories.Loader',
            'dbtemplates.loader.Loader',
        ],
    },
},
```

]

The order of `TEMPLATES.OPTIONS.loaders` is important. In the former example, templates from the database will be used as a fallback (ie. when the template does not exists in other locations). If you want the template from the database to be used to override templates in other locations, put `dbtemplates.loader.Loader` at the beginning of `loaders`.

4. Sync your database `python manage.py migrate`
5. Restart your Django server

CHAPTER 2

Usage

Creating database templates is pretty simple: Just open the admin interface of your Django-based site in your browser and click on "Templates" in the "Database templates" section.

There you only need to fill in the `name` field with the identifier, Django is supposed to use while searching for templates, e.g. `blog/entry_list.html`. The `content` field should be filled with the content of your template.

Optionally, by leaving the `content` field empty you are able to tell `dbtemplates` to look for a template with the `name` by using Django's other template loaders. For example, if you have a template called `blog/entry_list.html` on your file system and want to save the templates contents in the database, you just need to leave the content field empty to automatically populate it. That's especially useful if you don't want to copy and paste its content manually to the textarea.

CHAPTER 3

Advanced features

3.1 Caching

dbtemplates uses Django's default caching infrastructure for caching, and operates automatically when creating, updating or deleting templates in the database.

To enable one of them you need to specify a setting called DBTEMPLATES_CACHE_BACKEND to one of the valid values Django's CACHE_BACKEND can be set to. E.g.:

```
DBTEMPLATES_CACHE_BACKEND = 'memcached://127.0.0.1:11211/'
```

Note: Starting in version 1.0 dbtemplates allows you also to set the new dict-based CACHES setting, which was introduced in Django 1.3.

All you have to do is to provide a new entry in the CACHES dict named 'dbtemplates', e.g.:

```
CACHES = {
    'dbtemplates': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211',
    }
}
```

Please see the cache documentation if you want to know more about it.

3.2 Versioned storage

dbtemplates comes prepared to use the third party Django app django-reversion, that once installed besides dbtemplates allows you to jump back to old versions of your templates. It automatically saves every state when you save the template in your database and provides an easy to use interface.

Please refer to django-reversion's documentation for more information about how it works.

Hint: Just visit the "History" section of each template instance and browse its history.

3.2.1 Short installation howto

1. Get the source from the django-reversion project site and put it somewhere on your *PYTHONPATH*.
2. Add `reversion` to the `INSTALLED_APPS` setting of your Django project
3. Sync your database with `python manage.py syncdb`
4. Set `DBTEMPLATES_USE_REVERSION` setting to `True`

3.3 Management commands

`dbtemplates` comes with two Django management commands to be used with `django-admin.py` or `manage.py`:

- `sync_templates`

  Enables you to sync your already existing file systems templates with the database. It will guide you through the whole process.

- `create_error_templates`

  Tries to add the two templates `404.html` and `500.html` that are used by Django when a error occurs.

- `check_template_syntax`

  New in version 1.2.

  Checks the saved templates whether they are valid Django templates.

3.4 Admin actions

`dbtemplates` provides two admin actions to be used with Django>=1.1.

- `invalidate_cache`

  Invalidates the cache of the selected templates by calling the appropriate cache backend methods.

- `repopulate_cache`

  Repopulates the cache with selected templates by invalidating it first and filling then after that.

- `check_syntax`

  New in version 1.2.

  Checks the selected tempaltes for syntax errors.

CHAPTER 4

Settings

4.1 DBTEMPLATES_ADD_DEFAULT_SITE

`dbtemplates` adds the current site (`settings.SITE_ID`) to the database template when it is created by default. You can disable this feature by setting `DBTEMPLATES_ADD_DEFAULT_SITE` to `False`.

4.2 DBTEMPLATES_AUTO_POPULATE_CONTENT

`dbtemplates` auto-populates the content of a newly created template with the content of a template with the same name the other template loader. To disable this feature set `DBTEMPLATES_AUTO_POPULATE_CONTENT` to `False`.

4.3 DBTEMPLATES_CACHE_BACKEND

The dotted Python path to the cache backend class. See *Caching* for details.

4.4 DBTEMPLATES_USE_CODEMIRROR

A boolean, if enabled triggers the use of the CodeMirror based editor. Set to `False` by default.

4.5 DBTEMPLATES_USE_TINYMCE

New in version 1.3.

A boolean, if enabled triggers the use of the TinyMCE based editor. Set to `False` by default.

4.6 DBTEMPLATES_USE_REVERSION

A boolean, if enabled triggers the use of `django-reversion`.

4.7 DBTEMPLATES_MEDIA_PREFIX

The URL prefix for `dbtemplates`' media – CSS and JavaScript used by the CodeMirror based editor. Make sure to use a trailing slash, and to have this be different from the `STATIC_URL` setting (since the same URL cannot be mapped onto two different sets of files).

> **Warning:** Starting in version 1.0, `dbtemplates` uses the `STATIC_URL` setting, originally introduced by the django-staticfiles app. The app has since been added to Django itself and isn't needed if you use Django 1.3 or higher. Please refer to the contrib docs in that case.

CHAPTER 5

Changelog

5.1 v3.0 (2019-01-27)

Warning: This is a backwards-incompatible release!

- Dropped support for Django < 1.11.
- Added support for Django 2.0 and 2.1.
- Added support for Python 3.7.
- Recompiled Russian locale.
- Fixed byte string in migration file that caused the migration system to falsely think that there are new changes.
- Fixed string representation of template model, e.g. to improve readability in choice fields.

5.2 v2.0 (2016-09-29)

Warning: This is a backwards-incompatible release!

- Moved maintenance to the Jazzband
- Dropped support for Python 2.6
- Added support for Python 3.4 and 3.5
- Dropped support for Django < 1.8
- Removed South migrations. Please use Django's native migration system instead
- Removed the example project since it's out-of-date quickly

5.3 v1.3.2 (2015-06-15)

- support for Django 1.8 (not full, but usable)
- support for RedactorJS

thanks for contrib - @eculver, @kmooney, @volksman

5.4 v1.3.1 (2012-05-23)

- Minor release to move away from nose again and use own django-discover-runner.

5.5 v1.3 (2012-05-07)

- Dropped support for Django < 1.3 **backwards incompatible**
- Dropped using versiontools in favor of home made solution.
- Added optional support for TinyMCE editor instead of the CodeMirror editor (just enable `DBTEMPLATES_USE_TINYMCE`).
- Fixed compatibility to Django 1.4's handling of the `DATABASES` setting. Should also respect database routers now.
- Fixed an issue of the cache key generation in combination with memcache's inability to stomach spaces.
- Moved test runner to use nose and a hosted CI project at Travis: http://travis-ci.org/jazzband/django-dbtemplates

5.6 v1.2.1 (2011-09-07)

- Fixed a wrong use of the non-lazy localization tools.
- Fixed bugs in the documentation.
- Make use of django-appconf and versiontools.

5.7 v1.2 (2011-08-15)

- Refactored the template loader to be even more cache effective.
- Added `check_template_syntax` management command and admin action to make sure the saved templates are valid Django templates.

5.8 v1.1.1 (2011-07-08)

- Fixed bug in cache loading (again).
- Fixed bugs in the documentation.

Note: Since `dbtemplates` removed support for Django lower than 1.2 you have to use the template loader class in the `TEMPLATE_LOADERS` (`'dbtemplates.loader.Loader'`) and **not** the previosly included function that ended with `load_template_source`.

5.9 v1.1 (2011-07-06)

- **BACKWARDS-INCOMPATIBLE** Requires Django 1.2 or higher. For previous Django versions use an older versions of `dbtemplates`, e.g.:

```
$ pip install "django-dbtemplates<1.1"
```

- Added South migrations.

Note: If you are using South in your Django project, you can easily enable dbtemplates' migrations, *faking* the first migration by using the `--fake` option of South's `migrate` management command:

```
$ manage.py migrate --fake 0001 dbtemplates
```

Then run the rest of the migrations:

```
$ manage.py migrate dbtemplates
```

- Removed uniqueness on the `name` field of the `Template` model. This is needed because there isn't a `unique_together` for M2M fields in Django such as the `sites` field in the `Template` model.
- Made the `sites` field optional to support a way to apply a template to all sites.
- Added `--delete` option to `sync_templates` managment command to delete the file or database entry after syncing (depending on used `--overwrite` mode).
- Updated translations.
- Fixed issue with incorrectly splitting paths in `sync_templates`.
- Extended tests.
- Fixed issue with cache settings handling.

5.10 v1.0.1 (2011-04-14)

- Minor bugfixes with regard to the new cache handling.

5.11 v1.0 (2011-04-11)

Warning: This is the first stable release of django-dbtemplates which comes with a series of backwards incompatible changes.

- Removed own caching mechanism in favor of Django based caching mechanism. The `DBTEMPLATES_CACHE_BACKEND` is expected to be a valid cache backend URI, just like Django's own `CACHE_BACKEND` setting. In Django >= 1.3 an `'dbtemplates'` entry in the `CACHES` setting is also considered valid.
- Added tox configuration to test `dbtemplates` on Python 2.5, 2.6 and 2.7 with Django 1.1.X, 1.2.X and 1.3.X.
- Added Transifex configuration.
- Use `STATIC_URL` setting instead of `MEDIA_URL` for the media prefix. Also moved files from media/* to static/* to follow convention introduced in Django 1.3.
- Use ReadTheDocs for documentation hosting.

5.12 v0.8.0 (2010-11-07)

- Added Finnish translation (by jholster)
- Added –overwrite and –app-first options to sync_templates command (by Alex Kamedov).

5.13 v0.7.4 (2010-09-23)

- Fixed tests.

5.14 v0.7.3 (2010-09-21)

- Added `DBTEMPLATES_AUTO_POPULATE_CONTENT` setting to be able to disable to auto-populating of template content.
- Fixed cosmetic issue in admin with collapsable fields.

5.15 v0.7.2 (2010-09-04)

- Moved to Github again. Sigh.

5.16 v0.7.1 (2010-07-07)

- Fixed problem with the CodeMirror textarea, which wasn't completely disabled before.
- Fixed problem with the `DBTEMPLATES_MEDIA_PREFIX` setting, which defaults now to `os.path.join(settings.MEDIA_ROOT, 'dbtemplates')` now.

  In other words, if you don't specify a `DBTEMPLATES_MEDIA_PREFIX` setting and have the CodeMirror textarea enabled, dbtemplates will look in a subdirectory of your site's `MEDIA_ROOT` for the CodeMirror media files.

5.17 v0.7.0 (2010-06-24)

- Added CodeMirror-based syntax highlighting textarea, based on the amaxing work by Nic Pottier. Set the `DBTEMPLATES_USE_CODEMIRROR` setting to `True` to enable it.
- Make use of the full width in plain textarea mode.
- Added Chinese translation
- Added support for Django 1.2
- Updated French translation
- Added `DBTEMPLATES_USE_REVERSION` setting to be able to explicitely enable reversion support. (Default: `False`)

5.18 v0.6.1 (2009-10-19)

- Fixed issue with default site of a template, added ability to disable default site (`DBTEMPLATES_ADD_DEFAULT_SITE`).

5.19 v0.6.0 (2009-10-09)

- Updated and added locales (Danish, Brazilian Portuguese)
- Fixes an ambiguity problem with the cache invalidation
- Added `invalidate_cache` and `repopulate_cache` admin actions
- Added Sphinx documentation

5.20 v0.5.7

- Updates to the docs
- switch back to Bitbucket
- fixed tests
- Added Italian translation
- list of sites the template is used on
- fixed bug in `create_error_template` command.

5.21 v0.5.4

- Made loader and cache backends site-aware.
- The filesystem cache backend now saves the files under `<dir>/<site_domain>/<file_name>`.
- The Django cache backend the Site id in the cache key
- Template is now saved explicitly to backend if not existent in cache (e.g. if deleted manually or invalidated).

5.22 v0.5.3

- Removed automatic creation of 404.html and 50v0.html templates and added a new management command for those cases called `create_error_templates`
- Also reverted move to Bitbucket

5.23 v0.5.2

- Fixed a problem with `django.contrib.sites` when its table hasn't been populated yet on initialization of dbtemplates. Thanks for the report, Kevin Fricovsky
- Added an example Django project and docs for it

5.24 v0.5.1

- Removed unneeded code that registered the model with reversion.
- Updated docs a bit.
- Moved codebase to Bitbucket.
- Removed legacy `sync_templates.py` script, use `django-admin.py sync_templates` from now on.

5.25 v0.5.0

- Added support for django-reversion
- added feature that populates the content field automatically when left empty by using Django's other template loaders
- added caching backend system with two default backends:
  - `FileSystemBackend`
  - `DjangoCacheBackend`

  More about it in the blog post and in the docs.

5.26 v0.4.7

- Minor bugfix

5.27 v0.4.6

- Minor doc change and PyPI support

5.28 v0.4.5

- fixed the –force option of the sync_templates command

5.29 v0.4.4

- fixed error in custom model save() after changes in Django r8670.

5.30 v0.4.3

- removed oldforms code

5.31 v0.4.2

- added Hebrew translation (by mkriheli)

5.32 v0.4.1

- added French (by Roland Frederic) and German locale

5.33 v0.4.0

- adds better support for newforms-admin
- don't forget to load the dbtemplates.admin, e.g. by using django.contrib.admin.autodiscover() in you urls.py

5.34 v0.3.1

- adds a new management command *sync_templates* for bidirectional syncing between filesystem and database (backwards-compatible) and FilesystemCaching (thanks, Arne Brodowski!)

5.35 v0.2.5

- adds support for newforms-admin

CHAPTER 6

Support

Please leave your questions and messages on the designated site:

http://github.com/jazzband/django-dbtemplates/issues/