Django Citations Documentation

Release 0.3.0

William Hart

January 02, 2014

Contents

CHAPTER 1

Contents

1.1 Quick start

1.1.1 Installation

1. Install using:

   ```
   pip install django-citations
   ```

2. Add “citations” to your INSTALLED_APPS setting like this:

   ```
   INSTALLED_APPS = (
       ...
       ’citations’,
   )
   ```

3. Set up the database:

   ```
   python manage.py migrate citations
   ```

1.1.2 Basic Usage

You can use the admin panel to create references for your site. These can have a range of details including ISBN, URLs, titles, etc. Each reference must have a unique slug. This is how we will refer to the reference when we cite it within a template.

To refer to a reference in your database, you first need to load the tags in your template:

```
{% load citation_tags %}
```

You can then do the following in your template where you want the reference to appear:

```
{% cite "my_reference_slug" %}
```

You can refer to multiple references at a time by just adding new slugs separated by spaces. Note that quotation marks are required around the slugs, and slug names should not contain spaces:

```
{% cite "my_first_reference_slug" "my_second_reference_slug" %}
```

If you attempt to reference a work that is not in your reference list a TemplateSyntaxError will be raised showing the offending reference slug.

The references will be included in text as numbers - e.g. *[1]*, or *[1, 2]* for multiple references. Reference objects will also be placed in the *reference_list* variable of the template's context.

A reference list can be included at the bottom of the document:

```
{% show_references reference_list %}
```

1.2 Template tags

1.2.1 cite

> {% cite "slugs" ... %}

Use the cite template to cite a reference in the body of the text. Any number of reference slugs can be provided as space separated values inside quotation marks. A *reference_list* variable will be added to the template context which contains all the works cited in the article to date.

1.2.2 show_references

> {% show_references list_of_reference_objects %}

Displays a reference list of all the objects passed in the *list_of_reference_objects* variable.

1.2.3 show_all_references

> {% show_all_references %}

Shows a reference list including all references in the database, sorted alphabetically by author.

1.3 Advanced Tips and Tricks

1.3.1 Importing References from Bibtex

If you have the package *bibtexparser* installed then you will have access in the admin panel to an "Upload Bibtex" button. This feature will upload a bibtex file, scan it for fields that django-citations supports and save new records.

The *bibtexparser* package is provided under the LGPL.

1.3.2 Including a complete bibliography

As of version 0.3, you can include a complete bibliography by using the *show_all_references* template tag. This looks like the following:

```
{% load citation_tags %}

...

{% show_all_references %}
```

1.3.3 Loading templates from database

Sometimes you want to load the text including citations from the database. In this case, you can add a property to your model which builds the text into a template. This could look something like this:

```
from django.template import Template, Context

class MyModel(models.Model):
    body = models.TextField()

    def get_rendered_body(self):
        tpl = Template("{% load citation_tags %}" + self.body + "{% show_references reference_list %
        ctx = Context()
        return tpl.render(ctx)

    rendered_body = property(get_rendered_body)
```

This will include the text as well as a reference list at the bottom. Note that as the *reference_list* variable is located in the local template context then it is not available in the global template context - therefore you must include the reference list in the same template as the citations.

Citations is a simple Django app which lets you use a citations and reference lists within your templates.

- The source code is available from https://github.com/will-hart/django-citations under an MIT license.
- You can find the package on the Python Packing Index https://pypi.python.org/pypi/django-citations/.
- The latest documentation for the *develop* branch is available from http://django-citations.readthedocs.org/en/latest/

CHAPTER 2

Supported Fields

The fields currently supported by the django-citations database include:

- type (currently one of book, electronic journal, journal or website)
- slug (unique ID for the reference, max 128 characters)
- author (max 512 characters)
- title (max 512 characters)
- year (a number)
- series (max 512 characters)
- volume (a number)
- edition (a number)
- isbn (up to 17 characters - can include dashes)
- url (the URL where the resource was accessed online)
- publisher (max 128 characters)
- place (place of publishing, max 128 characters)
- abstract (free text)
- comments (free text)
- keywords (free text)
- accessed (the date the resource was first accessed)

CHAPTER 3

Change Log

Version 0.3

- Improved documentation
- Add new fields to database
- Add bibtex upload function, where *bibtexparser* is installed
- Add *show_all_references* tag for complete bibliography

Version 0.2.1

- Fix tags in README

Version 0.2

- Improved citation database - more fields available
- Improved reference output (uses a Harvard referencing format)

Version 1.0

- Initial version