
dpaste Documentation

Release 2.2

Martin Mahner

January 19, 2017

1	Testing and local development	3
2	Installation	5
3	Integrate dpaste into an existing project	7
4	Settings to override	9
5	API	11

dpaste is a Django based pastebin. It's intended to run separately but its also possible to be installed into an existing Django project like a regular app.

Contents:

Testing and local development

dpaste is continuously tested on [Travis](#). You can also run the test suite locally with `tox`:

```
$ cd dpaste/  
$ pip install tox  
$ tox --skip-missing-interpreters
```

A more manual approach is installing it all by hand in a virtual environment. This is also the preferred way to setup an environment for local development:

```
$ cd dpaste/  
$ pip install -e .  
$ pip install -r requirements.txt  
$ python runtests.py
```

Installation

Ready to contribute? Here's how to set up *dpaste* for local development.

1. Fork the *dpaste* repo on GitHub.
2. Clone your fork locally:

```
$ git clone https://github.com/<your_username>/dpaste.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv dpaste
$ cd dpaste/
$ pip install -r requirements.txt
```

4. Run the commands:

```
$ python manage.py syncdb
$ python manage.py migrate
```

5. Start up the webserver:

```
$ python manage.py runserver
```

Integrate dpaste into an existing project

Dpaste needs at least Django 1.4+ and is tested on Python 2.7 as well as Python 3.3.

Install the latest dpaste release in your environment. This will install all necessary dependencies of dpaste as well:

```
pip install dpaste
```

Add dpaste and mptt to your INSTALLED_APPS:

```
INSTALLED_APPS = (  
    'django.contrib.sessions',  
    'django.contrib.staticfiles',  
    # ...  
  
    'mptt',  
    'dpaste',  
)
```

Add dpaste — and if you want — the dpaste_api to your urlpatterns:

```
urlpatterns = patterns('',  
    # ...  
  
    url(r'pastebin/', include('dpaste.urls.dpaste')),  
    url(r'pastebin/api/', include('dpaste.urls.dpaste_api')),  
)
```

Finally just migrate the database schema:

```
manage.py migrate dpaste
```

3.1 Purge expired snippets

Do not forget to setup a cron job to purge expired snippets. You need to run the management command `cleanup_snippets`. A cron job I use looks like:

```
30 * * * * /srv/dpaste.de/bin/python /srv/dpaste.de/bin/manage.py cleanup_snippets > /dev/null
```

Note also that dpaste does *not* come with Django admin integration. You need to setup an register the models in an `admin.py` yourself.

Note: For further customization see [Settings to override](#).

Settings to override

When dpaste is installed as a standalone service or integrated into an existing project there are various settings you can override to adjust dpaste's behavior without touching the code:

DPASTE_SLUG_LENGTH Integer. Length of the random slug for each new snippet. In the rare case an existing slug is generated again, the length will increase by one more character. Default: 4

DPASTE_SLUG_CHOICES String. A string of characters which are used to create the random slug. Default: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890

DPASTE_LEXER_DEFAULT String. The lexer key that is pre-selected in the dropdown. Note that this is only used if the user has not saved a snippet before, otherwise we use the last-used lexer. Default: `python`

DPASTE_MAX_CONTENT_LENGTH Integer. Maximum number of bytes per snippet. Default: $250 * 1024 * 1024$

DPASTE_MAX_SNIPPETS_PER_USER Integer. Maximum number of snippets we save in the user session and display on the history page. Default: 10

DPASTE_BASE_URL String. The full qualified hostname and path to the dpaste instance. This is used to generate a link in the API response. Default: `https://dpaste.de`

DPASTE_SITE_NAME String. Site name to display in page titles. Default: `dpaste.de`

DPASTE_LEXER_LIST Choices. A tuple of choices of Pygments lexers used in the lexer dropdown. Here is the full [lexer list](#) which is currently used. Example:

```
DPASTE_LEXER_LIST = (
    ('delphi', 'Delphi'),
    ('php', 'PHP'),
    ('text', 'Text'),
)
```

DPASTE_EXPIRE_CHOICES Choices. A tuple of seconds and a descriptive string used in the lexer expiration dropdown. Example:

```
from django.utils.translation import ugettext_lazy as _
DPASTE_EXPIRE_CHOICES = (
    (3600, _(u'In one hour')),
    (3600 * 24 * 7, _(u'In one week')),
    (3600 * 24 * 30, _(u'In one month')),
    (3600 * 24 * 30 * 12 * 100, _(u'100 Years')),
)
```

One-Time snippets are supported. One-Time snippets are automatically deleted once a defined view count has reached (Default: 2). To enable one-time snippets you have to add a choice `onetime` to the expire choices:

```
from django.utils.translation import ugettext_lazy as _
DPASTE_EXPIRE_CHOICES = (
    ('onetime', _(u'One-Time snippet')),
    (3600, _(u'In one hour')),
    (3600 * 24 * 7, _(u'In one week')),
    (3600 * 24 * 30, _(u'In one month')),
)
```

You can also set the maximum view count after what the snippet gets deleted. The default is 2. One view is from the author, one view is from another user:

```
DPASTE_ONETIME_LIMIT = 2
```

Infinite snippets are supported. You can keep snippets forever when you set the choice key to `never`. The management command will ignore these snippets:

```
from django.utils.translation import ugettext_lazy as _
DPASTE_EXPIRE_CHOICES = (
    (3600, _(u'In one hour')),
    (u'never', _(u'Never')),
)
```

DPASTE_EXPIRE_DEFAULT The key of the default value of `DPASTE_EXPIRE_CHOICES`. Default: `3600 * 24 * 30 * 12 * 100` or simpler: `DPASTE_EXPIRE_CHOICES[2][0]`.

DPASTE_ENABLE_GIST **Removed in Version 2.13!** Boolean. Whether to display the Gist button for re-pasting to GitHub. Default: `True`

DPASTE_DEFAULT_GIST_NAME **Removed in Version 2.13!** String. The filename used when pasting a snippet on Github Gist. Default: `dpaste.de_snippet.py`

DPASTE_DEFAULT_GIST_DESCRIPTION **Removed in Version 2.13!** String. The filename used when pasting a snippet on Github Gist. Default: `dpaste.de_snippet.py`

DPASTE_JQUERY_URL String. URL to use for jQuery. Default: `//ajax.googleapis.com/ajax/libs/jquery/1/jquery.js`

dpaste provides a simple API endpoint to create new snippets. All you need to do is a simple `POST` request to the API endpoint `/api/`:

```
POST http://dpaste.de/api/
```

5.1 Available POST data for an API call:

5.1.1 content (required)

The UTF-8 encoded string you want to paste.

5.1.2 lexer (optional)

The lexer string key used for highlighting. See [lexer list](#) for a full list of choices. Default: `python`.

5.1.3 format (optional)

The format of the API response. Choices are:

- `default` — Returns a full qualified URL wrapped in quotes. Example:

```
"https://dpaste.de/xsWd"
```

- `url` — Returns the full qualified URL to the snippet, without surrounding quotes, but with a line break. Example:

```
https://dpaste.de/xsWd\n
```

- `json` — Returns a JSON object containing the URL, lexer and content of the the snippet. Example:

```
{
  "url": "https://dpaste.de/xsWd",
  "lexer": "python",
  "content": "The text body of the snippet."
}
```

5.1.4 expires (optional)

A keyword to indicate the lifetime of a snippet in seconds. The values are predefined by the server. Calling this with an invalid value returns a HTTP 400 BadRequest together with a list of valid values. Default: 2592000. In the default configuration valid values are:

- onetime
- never
- 3600
- 604800
- 2592000

5.1.5 filename (optional)

A filename which we use to determine a lexer, if `lexer` is not set. In case we can't determine a file, the lexer will fallback to `plain` code (no highlighting). A given `lexer` will overwrite any filename! Example:

```
{
  "url": "https://dpaste.de/xsWd",
  "lexer": "",
  "filename": "python",
  "content": "The text body of the snippet."
}
```

This will create a `python` highlighted snippet. However in this example:

```
{
  "url": "https://dpaste.de/xsWd",
  "lexer": "php",
  "filename": "python",
  "content": "The text body of the snippet."
}
```

Since the `lexer` is set too, we will create a `php` highlighted snippet.

Note: Since `lexer` defaults to `python` you have to specifically unset it when using `filename`.

Hint: You need to adjust the setting `DPASTE_BASE_URL` which is used to generate the full qualified URL in the API response. See [Settings](#) to override.

Note: When creating new snippets via the API, they won't be listed on the history page since they are related to a browser session.

5.2 Example code snippets:

A sample Python 2 script to publish snippets:


```
#!/usr/bin/env python

import urllib
import urllib2
import sys

def paste_code():
    request = urllib2.Request(
        'https://dpaste.de/api/',
        urllib.urlencode([('content', sys.stdin.read())]),
    )
    response = urllib2.urlopen(request)
    # Strip surrounding quotes (NB: response has no trailing newline)
    print response.read()[1:-1]

if __name__ == '__main__':
    paste_code()
```

You can simply use curl to publish a whole file:

```
$ alias dpaste="curl -F 'content=<-' https://dpaste.de/api/?format=url"
$ cat foo.txt | dpaste
https://dpaste.de/ke2pB
```


D

DPASTE_BASE_URL, [9](#)
DPASTE_DEFAULT_GIST_DESCRIPTION, [10](#)
DPASTE_DEFAULT_GIST_NAME, [10](#)
DPASTE_ENABLE_GIST, [10](#)
DPASTE_EXPIRE_CHOICES, [9](#)
DPASTE_EXPIRE_DEFAULT, [10](#)
DPASTE_JQUERY_URL, [10](#)
DPASTE_LEXER_DEFAULT, [9](#)
DPASTE_LEXER_LIST, [9](#)
DPASTE_MAX_CONTENT_LENGTH, [9](#)
DPASTE_MAX_SNIPPETS_PER_USER, [9](#)
DPASTE_SITE_NAME, [9](#)
DPASTE_SLUG_CHOICES, [9](#)
DPASTE_SLUG_LENGTH, [9](#)