django-docusign Documentation

Release 0.9

Benoît Bryon

January 20, 2015

Contents

django-docusign is a Django application for DocuSign's digital signature SAAS platform.

It uses pydocusign as *Python* client. It implements django-anysign API.

Project status

django-docusign is under active development. The project is not mature yet, but authors already use it! It means that, while API and implementation may change (improve!) a bit, authors do care of the changes.

Also, help is welcome! Feel free to report issues, request features or refactoring!

Resources

- Documentation: https://django-docusign.readthedocs.org
- Python package: https://pypi.python.org/pypi/django-docusign/
- Bugtracker: https://github.com/novapost/django-docusign/issues
- Changelog: https://django-docusign.readthedocs.org/en/latest/about/changelog.html
- Roadmap: https://github.com/novapost/django-docusign/milestones
- Code repository: https://github.com/novapost/django-docusign
- Continuous integration: https://travis-ci.org/novapost/django-docusign

Contents

3.1 Install

django-docusign is open-source software, published under BSD license. See *License* for details.

If you want to install a development environment, you should go to *Contributing* documentation.

3.1.1 Prerequisites

- Python [1] 2.7, 3.3 or 3.4. Other versions may work, but they are not part of the test suite at the moment.

3.1.2 As a library

In most cases, you will use *django-docusign* as a dependency of another project. In such a case, you should add `django-docusign` in your main project's requirements. Typically in `setup.py`:

```python
from setuptools import setup

setup(
    install_requires=[
        'django-docusign',
        #...
    ]
    # ...
)
```

Then when you install your main project with your favorite package manager (like pip [2]), *django-docusign* will automatically be installed.

3.1.3 Standalone

You can install *django-docusign* with your favorite Python package manager. As an example with pip [2]:

```
pip install django-docusign
```

---

[1] https://www.python.org/

[2] https://pypi.python.org/pypi/pip/

3.1.4 Check

Check *django-docusign* has been installed:

```
python -c "import django_docusign;print(django_docusign.__version__)"
```

You should get *django_docusign*'s version.

References

3.2 About django-docusign

This section is about the *django-docusign* project itself.

3.2.1 Vision

django-docusign helps you integrate and use Docusign [3] within a Django project.

References

3.2.2 Alternatives and related projects

This document presents other projects that provide similar or complementary functionalities. It focuses on differences or relationships with *django-docusign*.

DocuSign's helper libraries

There are other libraries related to DocuSign [4] service. Check https://www.docusign.com/developer-center/quick-start/api-explorer/helper-library

pydocusign

django-docusign uses pydocusign [5], a Python library for DocuSign [1]. *django-docusign* focuses on integration in *Django*.

django-anysign

django-anysign [6] provides generic tools to integrate online signing in *Django*. *django-docusign* implements *django-anysign* API.

---

[3] https://www.docusign.com/
[4] https://www.docusign.com/
[5] https://pypi.python.org/pypi/pydocusign/
[6] https://pypi.python.org/pypi/django-anysign/

References

3.2.3 License

Copyright (c) 2014, Benoît Bryon. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of django-docusign nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3.2.4 Authors & contributors

Maintainer: Benoît Bryon <benoit@marmelune.net>, as a member of the PeopleDoc [7] team: https://github.com/novapost/

Developers: https://github.com/novapost/django-docusign/graphs/contributors

Notes & references

3.2.5 Changelog

This document describes changes between each past release. For information about future releases, check milestones [8] and *Vision*.

0.9 (2015-01-20)

- Nothing changed yet.

---

[7] http://www.people-doc.com
[8] https://github.com/novapost/django-docusign/milestones

0.8 (2014-11-28)

Introducing DocuSign templates.

- Feature #28 - Introduced support for DocuSign templates to create envelopes. To run the tests, you'll need a new `PYDOCUSIGN_TEST_TEMPLATE_ID` environment variable: DocuSign's UUID of a template which has 2 roles.

0.7 (2014-10-22)

Bugfixes & refactoring, mostly around `SignatureCallbackView`.

- Bug #20 - In `SignatureCallbackView`:
  - `update_signature` is not triggered if signature is not updated, when a signer, not the last, has just signed the envelope.
  - `update_signer` is triggered for every signer when signature and signers have "Sent" status (signature has just been created in embedded mode).
- Features #16 and #20 - In demo application, `SignatureCallbackView` is tested using `pydocusign`'s templates.
- Bug #17 - In demo application, added tests around settings view... and fixed it, so that it works as expected.
- Bug #18 - In demo application, replaced using `iterator.next()` instead of `list(iterator)[0]`.
- Refactoring #19 - Removed deprecated `SignatureCallbackView.clean_status`. Use `pydocusign.DocuSignCallbackParser.envelope_status` instead.

0.6 (2014-10-10)

Custom email subject and blurb.

- Feature #10 - DocuSignBackend.create_signature accepts `subject` and `blurb` arguments, in order to customize email subject and body for DocuSign messages and embedded signing page.

0.5 (2014-10-08)

Improvements around multiple signers.

- Feature #13 - `DocuSignCallbackView` also handles recipient notifications.
- Feature #11 - Improved ordering of signers, better management of DocuSign's `routingOrder`.
- Feature #12 - Demo project shows how to implement signature of multiple signers.
- Bugfix #9 - Fixed default value of `signer_return_url` in `DocuSignBackend.post_recipient_view()`. Default value was not assigned if value was not passed as argument.

0.4 (2014-09-26)

- Feature #7 - Introduced `django_docusign.SignatureCallbackView`, as a generic view to handle DocuSign's notification callbacks. Simplified demo's code.

0.3 (2014-09-18)

Updated demo to manage DocuSign API callbacks.

- Feature #1 - Introduced support for DocuSign API callbacks (envelope updates). At the moment, the feature is mostly made of sample code in the demo project. It shows how *django-docusign* can be used to handle DocuSign's API callbacks. The idea is to let users try the feature then, later, add most useful parts in `django_docusign` package.

0.2 (2014-09-16)

Introduced demo application (with tests).

- Feature #4 - Introduced demo application: illustrate and test API usage.
- Feature #3 - On signature creation, tabs can be computed by backend.

0.1 (2014-08-12)

Initial release.

- Introduced `django_docusign.backend.DocuSignBackend` with "create signature" and "post recipient view" features.

Notes & references

3.3 Contributing

This document provides guidelines for people who want to contribute to the *django-docusign* project.

3.3.1 Create tickets

Please use [django-docusign bugtracker](#) [9] **before** starting some work:

- check if the bug or feature request has already been filed. It may have been answered too!
- else create a new ticket.
- if you plan to contribute, tell us, so that we are given an opportunity to give feedback as soon as possible.
- Then, in your commit messages, reference the ticket with some `refs #TICKET-ID` syntax.

3.3.2 Use topic branches

- Work in branches.
- Prefix your branch with the ticket ID corresponding to the issue. As an example, if you are working on ticket #23 which is about contribute documentation, name your branch like `23-contribute-doc`.
- If you work in a development branch and want to refresh it with changes from master, please [rebase](#) [10] or [merge-based rebase](#) [11], i.e. do not merge master.

---

[9] https://github.com/novapost/django-docusign/issues
[10] http://git-scm.com/book/en/Git-Branching-Rebasing
[11] http://tech.novapost.fr/psycho-rebasing-en.html

3.3.3 Fork, clone

Clone *django-docusign* repository (adapt to use your own fork):

```
git clone git@github.com:novapost/django-docusign.git
cd django-docusign/
```

3.3.4 Usual actions

The *Makefile* is the reference card for usual actions in development environment:

- Install development toolkit with pip [12]: `make develop`.

- Run tests with tox [13]: `make test`.

- Build documentation: `make documentation`. It builds Sphinx [14] documentation in *var/docs/html/index.html*.

- Release *django-docusign* project with zest.releaser [15]: `make release`.

- Cleanup local repository: `make clean`, `make distclean` and `make maintainer-clean`.

See also `make help`.

3.3.5 Use private credentials to run the tests

The test suite contains several integration tests, so it requires valid DocuSign account credentials. The test suite reads environment variables to get the setup. Here is an example to run the tests:

```
PYDOCUSIGN_TEST_ROOT_URL='https://demo.docusign.net/restapi/v2' \
PYDOCUSIGN_TEST_USERNAME='your-username' \
PYDOCUSIGN_TEST_PASSWORD='your-password' \
PYDOCUSIGN_TEST_INTEGRATOR_KEY='your-integrator-key' \
PYDOCUSIGN_TEST_TEMPLATE_ID='UUID-of-your-docusign-template' \
PYDOCUSIGN_TEST_SIGNER_RETURN_URL='http://example.com/signer-return/' \
PYDOCUSIGN_TEST_CALLBACK_URL='http://example.com/callback/' \
make test
```

Notes & references

---

[12] https://pypi.python.org/pypi/pip/
[13] https://pypi.python.org/pypi/tox/
[14] https://pypi.python.org/pypi/Sphinx/
[15] https://pypi.python.org/pypi/zest.releaser/

Indices and tables

- *genindex*
- *modindex*
- *search*