

---

# **testpad Documentation**

***Release 0.1.5***

**Milind Shakya**

October 03, 2016



---

## Contents

---

<b>1</b>	<b>testpad</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Quickstart . . . . .	3
1.3	Features . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
4.4	Tips . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
6.1	0.1.0 (2015-10-02) . . . . .	15
6.2	0.1.5 (2015-11-02) . . . . .	15



Contents:



---

**testpad**

---

Screenshot/Annotate and send to backend for processing.

## 1.1 Documentation

The full documentation is at <https://testpad.readthedocs.org>.

## 1.2 Quickstart

Install testpad:

```
pip install testpad
```

Then use it in a project by updating your settings file (ideally local and test settings and not production settings.):

```
INSTALLED_APPS += (
    'settings_context_processor',
    'testpad',
)
TEMPLATE_CONTEXT_PROCESSORS += ('settings_context_processor.context_processors.settings', )
USE_TESTPAD = True
TESTPAD_AJAX_URL = 'Insert API where screenshot is posted.'

# for settings_context_processor
TEMPLATE_VISIBLE_SETTINGS = (
    'USE_TESTPAD',
    'TESTPAD_AJAX_URL',
)
```

Add this block to your base template:

```
{# JS block for automated QA utility to create bugs in Jira #}
{# Will be turned off for production #}
{% if USE_TESTPAD %}
    {# This is optional. Remove if you already have jquery enabled. #}
    <script src="http://code.jquery.com/jquery-latest.min.js"></script>
    <script src="{{ STATIC_URL }}js/testpad/lib/feedback.js"></script>
    <link rel="stylesheet" href="{{ STATIC_URL }}css/testpad/lib/feedback.min.css" />
    <script type="text/javascript">
        $.feedback({
```

```
    ajaxURL: "{{ TESTPAD_AJAX_URL }}",
    html2canvasURL: "{{ STATIC_URL }}js/testpad/lib/html2canvas.js"
  );
</script>
{%
  endif %}

```

Lastly:

```
python manage.py collectstatic
```

## 1.3 Features

- Captures screenshot based on <https://github.com/ivoviz/feedback> for django apps.
- Uses django\_settings\_context\_processor to make settings constant available in your base templates.
- **Demo:** <http://ivoviz.github.io/feedback/>. Click “Send Feedback” button on lower right.

TestPad client that includes all the javascript magic to make screenshotting and annotations happen.

Django App for <https://github.com/ivoviz/feedback>

## **Installation**

---

At the command line:

```
$ easy_install testpad
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv testpad
$ pip install testpad
```



### Usage

---

To use testpad in a project:

```
import testpad
```



---

## Contributing

---

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at <https://github.com/milin/testpad/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

#### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

#### 4.1.4 Write Documentation

testpad could always use more documentation, whether as part of the official testpad docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/milin/testpad/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/testpad.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 testpad
$ cd testpad/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 testpad tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check [https://travis-ci.org/milin/testpad/pull\\_requests](https://travis-ci.org/milin/testpad/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_testpad
```



## **Credits**

---

### **5.1 Development Lead**

- Milind Shakya <sh.milind@gmail.com>

### **5.2 Contributors**

None yet. Why not be the first?



### History

---

#### 6.1 0.1.0 (2015-10-02)

- First release on PyPI.

#### 6.2 0.1.5 (2015-11-02)

- Updates to directory structure for static files so that css and js files go in proper directories.