
wagtail-srcset Documentation

Release 0.1.5

Johannes Spielmann

Apr 24, 2023

Contents

1	wagtail-srcset	3
1.1	Quickstart	3
1.2	Features	4
1.3	Todo	4
1.4	Documentation	4
1.5	What is this all about?	4
1.6	Running Tests	7
1.7	Running the Example App	8
1.8	Run Linter	8
1.9	Coverage	8
1.10	Start a Notebook Server	8
1.11	Build Documentation	8
1.12	Publish Package	8
2	Installation	9
3	Usage	11
4	Contributing	13
4.1	Types of Contributions	13
4.2	Get Started!	14
4.3	Pull Request Guidelines	15
4.4	Tips	15
5	Credits	17
5.1	Development Lead	17
5.2	Contributors	17
6	History	19
6.1	0.3.0 (2023-04-24)	19
6.2	0.2.0 (2022-11-03)	19
6.3	0.1.5 (2020-05-21)	19
6.4	0.1.4 (2020-04-22)	20
6.5	0.1.3 (2020-04-13)	20
6.6	0.1.2 (2020-04-05)	20
6.7	0.1.0 (2019-XX-YY)	20

Contents:

CHAPTER 1

wagtail-srcset

HTML5 image srcset support for Wagtail

1.1 Quickstart

Install wagtail-srcset:

```
pip install wagtail-srcset
```

Add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    "wagtail_srcset.apps.WagtailSrcsetConfig",  
    ...  
)
```

Use it in your templates:

```
{% load wagtail_srcset_tags %}  
{% srcset_image img width-600 %}
```

Or via wagtails “as” syntax:

```
{% load wagtail_srcset_tags %}  
{% srcset_image img width-60 as thumbnail %}  
<a href="{{ thumbnail.original }}">
```

(continues on next page)

(continued from previous page)

```

</a>
```

1.2 Features

- Generate srcset attribute dynamically, based on the size of an image and the width attribute of the template tag if SRCSET_DYNAMIC is True
- You can specify default images sizes in DEFAULT_SRCSET_RENDITIONS
- wagtails WAGTAILIMAGES_JPEG_QUALITY is used for jpeg quality when set

1.3 Todo

- Dont just support the width resize-rule add fill, max, min etc

1.4 Documentation

The full documentation is at <https://wagtail-srcset.readthedocs.io>.

1.5 What is this all about?

I tried to use wagtail as a basis for a personal blog engine (yeah I know). Playing around with some images I noticed that they looked not as sharp as on my old page and I wondered why. Finally I found out that wagtail images with width-600 for example are implicitly upscaled on modern display devices. For a more detailed description and demonstration with an actual image, take a look at the image below and maybe view it at 100% scale.

Difference between native wagtail and wagtail_srcset image tag

Both images have a width of 600px browser pixels. But 1px on a legacy device and 1px on a modern high pixel density device are not the same. Phones, tablets or modern computer displays often use up to 3 hardware pixels to show 1 browser pixel.

This is the original wagtail image created with `{% image img width-600 %}`. It looks blurry because the original image is at first scaled to 600px width by wagtail and stored in a image rendition. Then this 600px image is shown on a high pixel density screen. Since those screens use more hardware pixels to display those 600 browser pixels, the image is upscaled implicitly by a factor of 2 to 3. Ouch.



This is the image generated by `{% srcset_image img width-600 %}` which should be sharper on modern display devices, because now 1 browser image pixel should roughly map to one pixel on the device again. We present the browser with different image size alternatives and the browser chooses the most appropriate size for its device's pixel density which could mean that a 1800px width image is displayed for a 600 browser pixel width image.



Here are two amplified sections, to make the difference more visible:





This package aims to provide a new image tag for wagtail that produces sharp looking images by generating a srcset attribute that includes larger images for higher pixel density devices.

If you are concerned about the increased image size I would recommend to use more aggressive lossy compression instead of upscaling.

1.6 Running Tests

Does the code actually work?

```
# create a virtualenv and activate it
git clone https://github.com/ephes/wagtail_srcset.git
cd wagtail_srcset
python -m pip install flit
flit install -s
pytest
```

1.7 Running the Example App

```
# activate virtualenv where wagtail_srcset is installed
python manage.py runserver --settings example.settings 0.0:8000
```

1.8 Run Linter

```
# activate virtualenv where wagtail_srcset is installed
flake8 wagtail_srcset tests
```

1.9 Coverage

```
# activate virtualenv where wagtail_srcset is installed
pytest --cov=wagtail_srcset --cov-report=html
```

1.10 Start a Notebook Server

```
# activate virtualenv where wagtail_srcset is installed
DJANGO_ALLOW_ASYNC_UNSAFE=true python manage.py shell_plus --notebook
```

1.11 Build Documentation

```
# activate virtualenv where wagtail_srcset is installed
rm docs/modules.rst docs/wagtail_srcset.rst
sphinx-apidoc -o docs/ wagtail_srcset
make -C docs clean
make -C docs html
open docs/_build/html/index.html
```

1.12 Publish Package

```
# activate virtualenv where wagtail_srcset is installed
flit publish
```

CHAPTER 2

Installation

At the command line:

```
$ pipx install wagtail-srcset
```


CHAPTER 3

Usage

To use wagtail-srcset in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    "wagtail_srcset.apps.WagtailSrcsetConfig",  
    ...  
)
```

Add wagtail-srcset's URL patterns:

```
from wagtail_srcset import urls as wagtail_srcset_urls  
  
urlpatterns = [  
    ...  
    path("", include(wagtail_srcset_urls)),  
    ...  
]
```


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/ephes/wagtail_srcset/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

wagtail-srcset could always use more documentation, whether as part of the official wagtail-srcset 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/ephes/wagtail_srcset/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 *wagtail-srcset* for local development.

1. Fork the *wagtail-srcset* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/wagtail-srcset.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 wagtail-srcset
$ cd wagtail-srcset/
$ 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 wagtail_srcset 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/ephes/wagtail-srcset/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_wagtail_srcset
```


5.1 Development Lead

- Jochen Wersdörfer <jochen-wagtailsrct@wersdoerfer.de>
- Johannes Spielmann <j@spielmannsolutions.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.3.0 (2023-04-24)

- Add support for Wagtail>=5
- Drop support for Wagtail<3.0
- Added workaround for image_node.filter_spec -> image_node.filter_specs change in Wagtail 5
- Some small spelling fixes and extended docstrings
- pre-commit autoupdate
- make tox check against python (3.9, 3.10, 3.11) and wagtail (3, 4, 5)
- fixed codecov.io workflow
- added supported python and wagtail versions to readme via badges

6.2 0.2.0 (2022-11-03)

- Replace poetry with flit
- Make tox run tests with flit for last 3 Python versions
- Updated dependencies

6.3 0.1.5 (2020-05-21)

- Bumped down python requirement to 3.6 (since f-strings are used) - request by CVYY39
- Updated dependencies

6.4 0.1.4 (2020-04-22)

- Updated dependencies

6.5 0.1.3 (2020-04-13)

- Add original url to rendition generated by “as” syntax

6.6 0.1.2 (2020-04-05)

- Added support for wagtails “as” syntax

6.7 0.1.0 (2019-XX-YY)

- First release