nider Documentation

Release 0.5.0

Vladyslav Ovchynnykov

Aug 13, 2020

Contents

1 Installation **3**
- 1.1 Stable release . . . 3
- 1.2 From sources . . . 3

2 Usage **5**
- 2.1 Image units . . . 5
- 2.2 Image content . . . 8
- 2.3 Initializing an image . . . 8
- 2.4 Drawing on the image . . . 9
- 2.5 Adding watermarks . . . 11

3 nider package **13**
- 3.1 nider.core module . . . 13
- 3.2 nider.models module . . . 13
- 3.3 nider.exceptions module . . . 13

4 Contributing **15**
- 4.1 Types of Contributions . . . 15
- 4.2 Get Started! . . . 16
- 4.3 Pull Request Guidelines . . . 17
- 4.4 Tips . . . 17

5 Credits **19**
- 5.1 Development Lead . . . 19
- 5.2 Contributors . . . 19

6 History **21**
- 6.1 0.1.0 (2017-07-27) . . . 21
- 6.2 0.2.0 (2017-08-12) . . . 21
- 6.3 0.3.0 (2017-08-17) . . . 21
- 6.4 0.4.0 (2017-09-14) . . . 21

7 Indices and tables **23**

Python Module Index **25**

Index **27**

Nider is an approach to make generation of text based images simple yet flexible.

CHAPTER 1

Installation

1.1 Stable release

To install nider, run this command in your terminal:

```
$ pip install nider
```

This is the preferred method to install nider, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

1.2 From sources

The sources for nider can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/pythad/nider
```

Or download the tarball:

```
$ curl  -OL https://github.com/pythad/nider/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 2

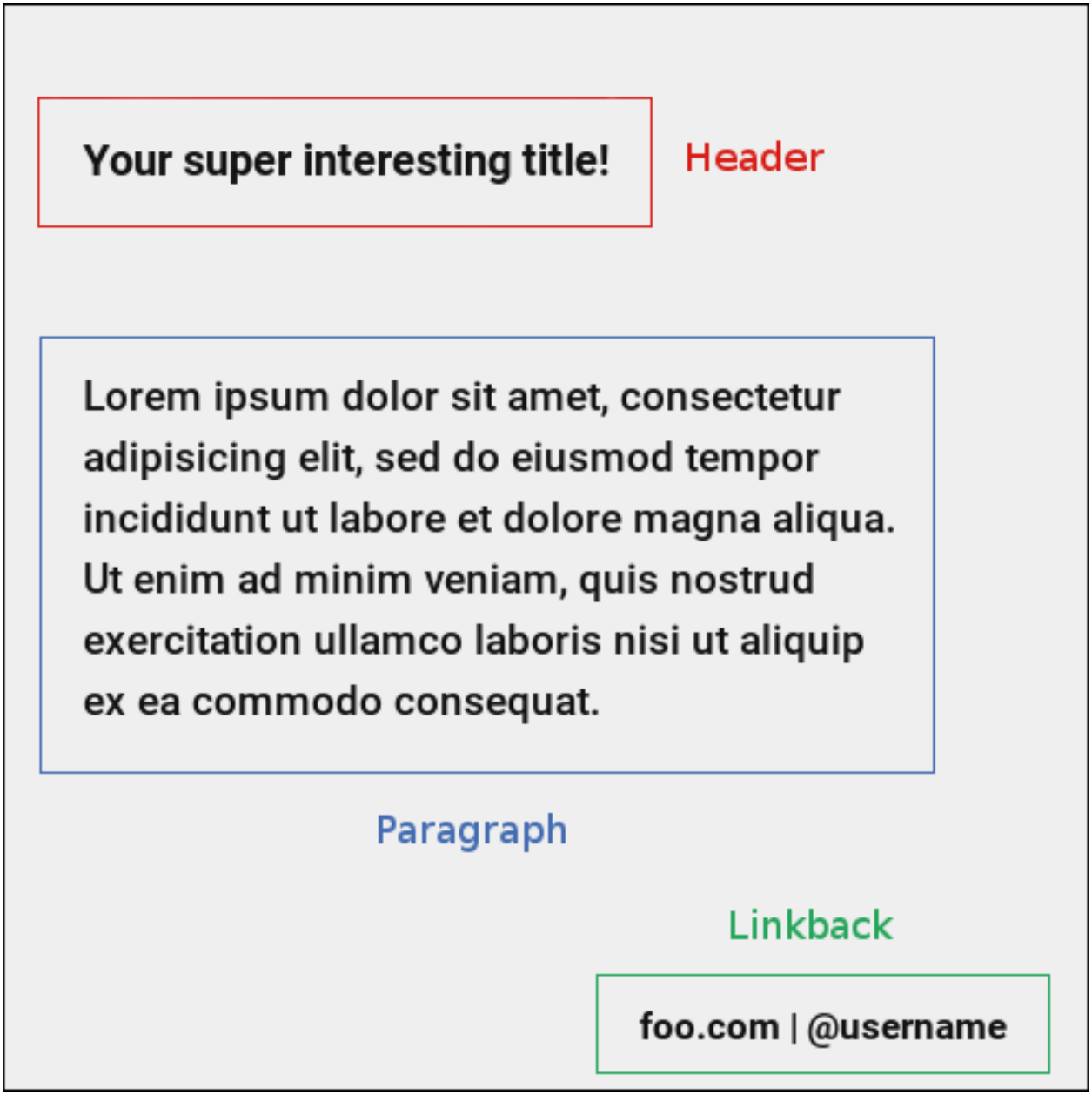
Usage

This article is a tutorial for `nider` package and at the same time it is a full reference of all `nider` models and possibilities.

2.1 Image units

There are three main units each `nider.Image` can consist of:

- header
- paragraph
- linkback



Each of the units is represented by a class in `nider.models`:

- `nider.models.Header`
- `nider.models.Paragraph`
- `nider.models.Linkback`

2.1.1 `nider.models.Header`

Example

```
from nider.core import Font
from nider.core import Outline
```

(continues on next page)

(continued from previous page)

```
from nider.models import Header


header = Header(text='Your super interesting title!',
                font=Font('/home/me/.local/share/fonts/Roboto/Roboto-Bold.ttf', 30),
                text_width=40,
                align='left',
                color='#ededed',
                outline=Oultine(2, '#222')
                )
```

2.1.2 nider.models.Paragraph

This class has the same attribures and behaviour as `nider.models.Header`.

Example

```
from nider.core import Font
from nider.core import Outline

from nider.models import Paragraph


para = Paragraph(text='Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed␣
↪do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim␣
↪veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo␣
↪consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum␣
↪dolore eu fugiat nulla pariatur.',
                 font=Font('/home/me/.local/share/fonts/Roboto/Roboto-Bold.ttf', 30),
                 text_width=65,
                 align='left',
                 color='#ededed'
                 outline=Oultine(1, '#000')
                 )
```

2.1.3 nider.models.Linkback

Example

```
from nider.core import Font
from nider.core import Outline

from nider.models import Linkback


linkback = Linkback(text='foo.com | @username',
                    font=Font('/home/me/.local/share/fonts/Roboto/Roboto-Bold.ttf',␣
↪30),
                    color='#ededed',
                    outline=Oultine(2, '#000')
                    )
```

Note: Parameters `color` and `outline.color` are optional for any unit. They can be generated automatically by `nider`. `nider` analyzes background color of either a texture or of an image and chooses an opposite one to it. So if your image in mainly dark , white text color will be auto generated and set. The same applies to outline color.

Although it's a nice feature for backgrounds you have no control over, we'd recommend to provide colors explicitly.

2.2 Image content

In order to aggregate all of the units together you need to create an instance of `nider.models.Content` class.

2.2.1 `nider.models.Content`

Example

```
from nider.models import Content
from nider.models import Linkback
from nider.models import Paragraph


para = Paragraph(...)

linkback = Linkback(...)

content = Content(para, linkback=linkback, padding=60)
```

2.3 Initializing an image

After the content is prepared it's the right time to initialize an image. In `nider` a basic image is represented by `nider.models.Image`.

2.3.1 `nider.models.Image`

Example

```
from nider.models import Content
from nider.models import Image


content = Content(...)

img = Image(content,
            fullpath='example.png',
            width=500,
            height=500
            )
```

Social media images

`nider` comes with some pre-built models that can be used to generate images for some social networks. These are subclasses of `nider.models.Image` with changed size.

Instagram

- `nider.models.InstagramSquarePost` - 1080x1080 image
- `nider.models.InstagramPortraitPost` - 1080x1350 image
- `nider.models.InstagramLandscapePost` - 1080x566 image

Facebook

- `nider.models.FacebookSquarePost` - 470x470 image
- `nider.models.FacebookLandscapePost` - 1024x512 image

Twitter

- `nider.models.TwitterPost` - 1024x512 image
- `nider.models.TwitterLargeCard` - 506x506 image

I highly recommend reading this post if you are curious about what are the right image sizes for social media images.

2.4 Drawing on the image

Having an instance of `nider.models.Image` we are ready to create a real image.

`nider` comes with 3 options of drawing your image:

- `Image.draw_on_texture` - draws preinitialized image and its attributes on a texture.

Note: You don't need to create textured images by pasting texture mulpitle times in Photoshop or Gimp. `nider` takes care of filling image of any size with textrure you privide.

- `Image.draw_on_bg` - Draws preinitialized image and its attributes on a colored background. nider uses a color you provide to fill the image and then draws the content.
- `Image.draw_on_image` - Draws preinitialized image and its attributes on an image. Content will be drawn directly on the image you provide.

2.4.1 `Image.draw_on_texture`

Example

```
from nider.models import Content
from nider.models import Image


content = Content(...)

img = Image(content,
            fullpath='example.png',
            width=500,
            height=500
            )

img.draw_on_texture('example_texture.png')
```

Check the full example here .

`nider` comes with a huge bundle of textures. As for now you need to copy them to your machine if you want to use any of them.

2.4.2 `Image.draw_on_bg`

Example

```
from nider.models import Content
from nider.models import Image


content = Content(...)

img = Image(content,
            fullpath='example.png',
            width=500,
            height=500
            )

img.draw_on_bg('#efefef')
```

Check the full example here .

2.4.3 `Image.draw_on_image`

Examples

```
from nider.models import Content
from nider.models import Image


content = Content(...)

img = Image(content,
            fullpath='example.png',
            width=500,
```

(continues on next page)

(continued from previous page)

```
            height=500
            )

img.draw_on_image('example_bg.jpg')
```

Using filters and enhancements:

```
img.draw_on_image('example_bg.jpg',
                  image_enhancements=((ImageEnhance.Contrast, 0.75),
                                      (ImageEnhance.Brightness, 0.5)),
                  image_filters=((ImageFilter.BLUR),),
                  )
```

Check the full example here .

That's it. After any of draw methods has been called and successfully completed the new image will be saved to `Image.fullpath`.

2.5 Adding watermarks

`nider` comes with built-in support for adding watermarks to your images.

First of all you need to create an instanse of `nider.models.Watermark` class.

2.5.1 Example

```
watermark = Watermark(text='COPYRIGHT',
                      font=Font('/home/me/.local/share/fonts/Roboto/Roboto-Bold.ttf'),
                      color='#111',
                      cross=True,
                      rotate_angle=-45,
                      opacity=0.35
                      )
```

After this you can either add watermark to you `Content` instance and draw watermark on `nider` generated images:

```
from nider.models import Content
from nider.models import Image
from nider.models import Watermark


watermark = Watermark(...)

content = Content(..., watermark=watermark)

img = Image(content,
            fullpath='example.png',
            width=500,
            height=500
            )
```

(continues on next page)

(continued from previous page)

```
img.draw_on_bg('#efefef')
```

or you can add a watermark to an existing image using `nider.tools.add_watermark()`:

2.5.2 Example

```
from nider.models import Watermark

from nider.tools import add_watermark


watermark = Watermark(...)
add_watermark('path/to/my/img', watermark)
```

CHAPTER 3

nider package

3.1 nider.core module

3.2 nider.models module

3.3 nider.exceptions module

exception nider.exceptions.**AutoGeneratedUnitColorUsedWarning**(*unit*, *color_used*)
Warning raised when auto generated unit color was used

exception nider.exceptions.**AutoGeneratedUnitOutlinecolorUsedWarning**(*unit*, *color_used*)
Warning raised when auto generated unit's outline color was used

exception nider.exceptions.**DefaultFontWarning**
Warning raised when default font was used

exception nider.exceptions.**FontNotFoundException**(*fontpath_provided*)
Exception raised when font cannot be found

exception nider.exceptions.**FontNotFoundWarning**(*fontpath_provided*)
Warning raised when font cannot be found

exception nider.exceptions.**ImageGeneratorException**
Base class for exceptions raised by nider

exception nider.exceptions.**ImageGeneratorWarning**
Base class for warnings raised by nider

exception nider.exceptions.**ImageSizeFixedWarning**
Warning raised when the size of the image has to be adjusted to the provided content's size because the content takes much space

exception nider.exceptions.**InvalidAlignException**(*align_provided*, *available_aligns=None*)

Exception raised when align is not supported by nider

CHAPTER 4

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/pythad/nider/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" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

nider could always use more documentation, whether as part of the official nider 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/pythad/nider/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 *nider* for local development.

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

```
$ git clone git@github.com:your_name_here/nider.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 nider
$ cd nider/
$ 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 nider tests
$ python setup.py test or 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 3.4 and 3.5. Check https://travis-ci.org/pythad/nider/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 discover tests
```

CHAPTER 5

Credits

5.1 Development Lead

- Vladyslav Ovchynnykov <ovd4mail@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.1.0 (2017-07-27)

- First release on PyPI.

6.2 0.2.0 (2017-08-12)

- Added `PIL.ImageEnhance` and `PIL.ImageFilter` built-in support
- Enabled auto color generation for unit colors

6.3 0.3.0 (2017-08-17)

- Dropped shadow support for units
- Added outline support for units
- Made unit's font config as a separate class

6.4 0.4.0 (2017-09-14)

- Added ability to add watermarks to images

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

n

nider.exceptions, 13

Index

A

AutoGeneratedUnitColorUsedWarning, 13
AutoGeneratedUnitOutlinecolorUsedWarning, 13

D

DefaultFontWarning, 13

F

FontNotFoundException, 13
FontNotFoundWarning, 13

I

ImageGeneratorException, 13
ImageGeneratorWarning, 13
ImageSizeFixedWarning, 13
InvalidAlignException, 13

N

nider.exceptions (*module*), 13