
mr.docs Documentation

Release 2.1

SVX

October 02, 2016

1	Features	3
----------	-----------------	----------

A container wrapper based on [Alpine Linux](#) for [Sphinx](#) based [reStructuredText](#) [rst] written documentation.

mr.docs is written for writing and testing documentation for [Plone](#). It may or may not fit your use case.

The main purpose of mr.docs is **testing**, this is what mr.docs is written for. However you also can use mr.docs to build your documentation for *production*.

Features

- One installation
- One configuration
- Speed
- Works with different Operating Systems
- Not messing with system python and other dependencies
- ‘Smallish’
- reStructuredText lint
- Sphinx lint
- HTML test-builds
- HTML deployment builds
- Spell-check

1.1 Dependencies

- Docker
- Bash

If you do not have Docker installed yet, please follow the [official install guide](#).

1.2 Getting mr.docs

```
docker pull quay.io/tiramisu/mr.docs
```

1.2.1 Updating

```
docker pull quay.io/tiramisu/mr.docs
```

1.3 Assumptions

mr.docs assumes that the documentation you want to test is located in a directory called `docs` and that the documentation is written in `.rst`.

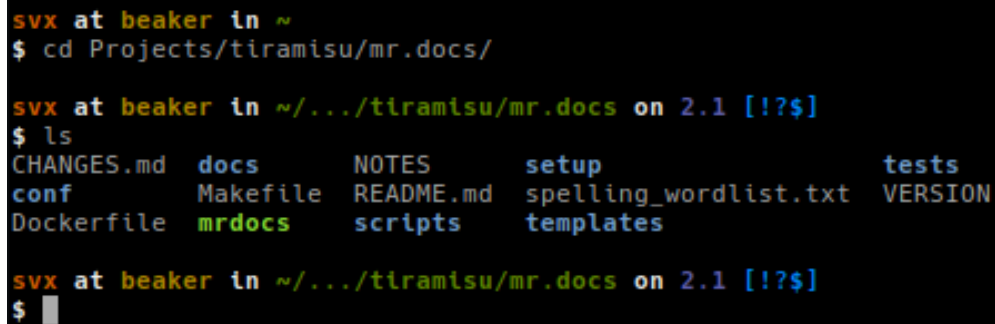
1.3.1 Example

Change into the main directory of your project.

For this example the code repository of the [mr.docs](#) is used.

```
cd Projects/tiramisu/mr.docs/
```

If you do now a `ls`, you will see we have here a directory called *docs*.

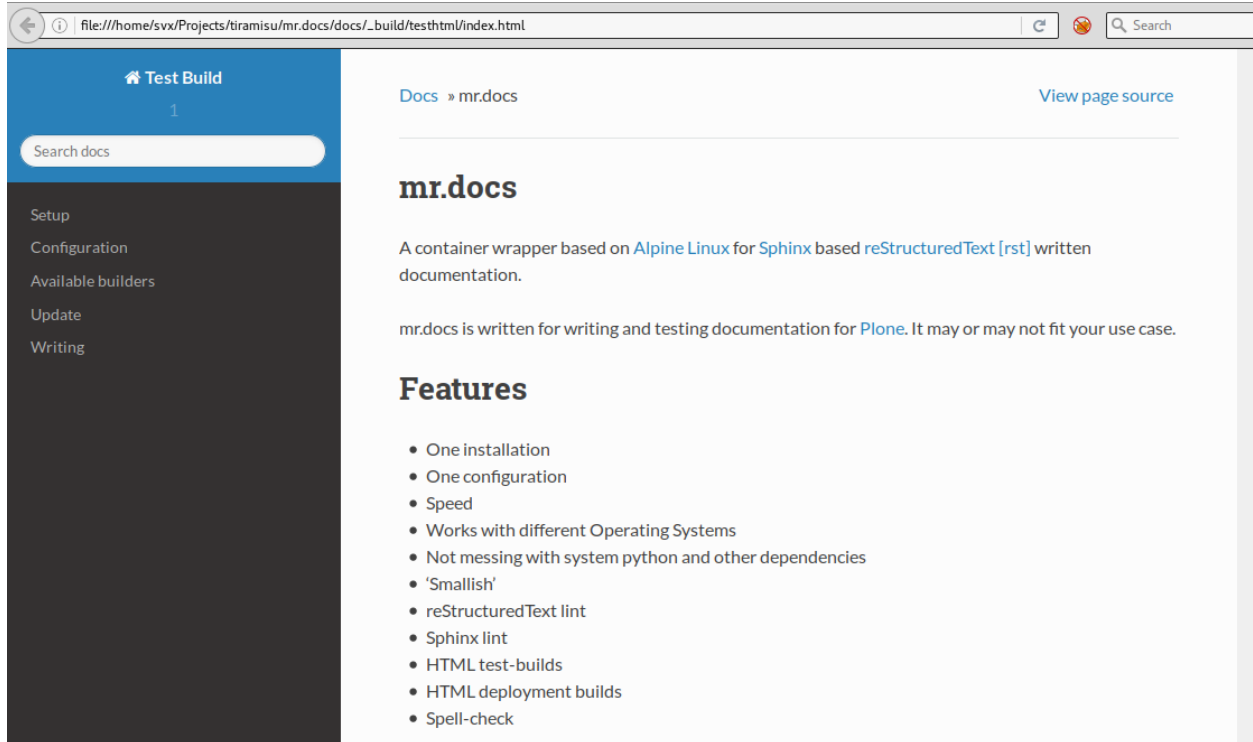
A terminal window with a dark background and light-colored text. The prompt is 'svx at beaker in ~'. The user enters '\$ cd Projects/tiramisu/mr.docs/'. The prompt changes to 'svx at beaker in ~/.../tiramisu/mr.docs on 2.1 [!?\$]'. The user enters '\$ ls'. The output is a directory listing: 'CHANGES.md', 'docs', 'NOTES', 'setup', 'tests', 'conf', 'Makefile', 'README.md', 'spelling_wordlist.txt', 'VERSION', 'Dockerfile', 'mrdocs', 'scripts', 'templates'. The user enters '\$' and the prompt returns to 'svx at beaker in ~/.../tiramisu/mr.docs on 2.1 [!?\$]'.

```
svx at beaker in ~
$ cd Projects/tiramisu/mr.docs/

svx at beaker in ~/.../tiramisu/mr.docs on 2.1 [!?$]
$ ls
CHANGES.md  docs      NOTES     setup      tests
conf         Makefile  README.md spelling_wordlist.txt VERSION
Dockerfile   mrdocs    scripts   templates
```

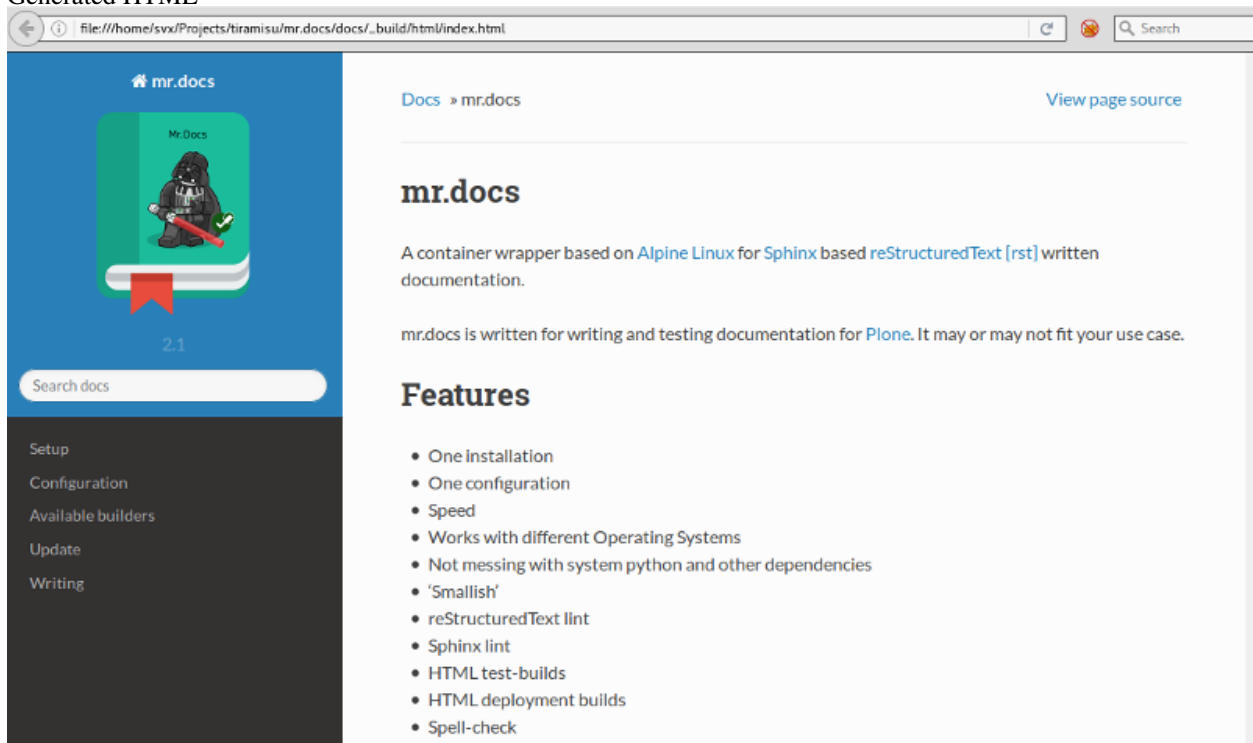
Running a test build:

Generated HTML



Running a deployment build:

Generated HTML



Usage

Commands

Running A Test Build

```
docker run --rm -v "${PWD}/docs":/build/docs:rw -u "$(id -u)": "$(id -g)" quay.io/tiramisu/mr.docs test
```

Running Spell-Check

```
docker run --rm -v "${PWD}/docs":/build/docs:rw -u "$(id -u)": "$(id -g)" quay.io/tiramisu/mr.docs spell
```

Read more about [spell-check](#).

Using Configuration Template

```
docker run --rm -v "${PWD}/docs":/build/docs:rw -u "$(id -u)": "$(id -g)" quay.io/tiramisu/mr.docs config
```

Read more about a custom base setup.

Building For Deployment

```
docker run --rm -v "${PWD}/docs":/build/docs:rw -u "$(id -u)": "$(id -g)" quay.io/tiramisu/mr.docs build
```

Read more about a deployment setup.

Using A Script

Note: This is on your own risk, before you download and use it, please review it first! Never use scripts without review !!

To make the usage of mr.docs easier and faster, there is a bash script, which also adds certain logics such as directory and file checking. **Use this on your own risk !**

You can find the script on [GitHub](#).

```
curl -o /usr/local/bin https://raw.githubusercontent.com/tiramisusolutions/mr.docs/2.1/mrdocs
```

This will download the script called *mr.docs* into */usr/local/bin*.

Feel free to adjust the location according to your setup/needs.

Script Usage:

```
Usage: mrdocs [option]
-help, -h  => show this help
-config, -c => create example config as starting point
-html      => builds html version ready for deploy
-testhtml  => builds test version, for local developing
-spell, -s => runs a spell-check against the docs

For more, please read the docs at https://mrdocs.readthedocs.io/en/latest/
```

Example

Running a test build:

```
mrdocs -testbuild
```

Configuration

New in version 2.1.

Configuration

mr.docs uses a set of pre-configured add-ons and settings, which are most useful for testing [Plone](#) documentation.

You can find it on [GitHub](#).

Use Your Own Configuration File

Note: There is no warranty that mr.docs will work with your own custom configuration, since mr.docs is build with a pre-configured set of add-ons.

If you have already a own configuration file [*conf.py*] you can try use it. Copy it into your *docs* directory and it may work, it depends on your configuration. If your own setup, is using a add-on which is not installed within mr.docs this will not work.

Create Base Configuration You can use mr.docs to create a base configuration.

This is a good start, if you want to test building your project documentation, with your logo, company name and so on.

Running

```
docker run --rm -v "${PWD}/docs":/build/docs:rw -u "$(id -u)": "$(id -g)" quay.io/tiramisu/mr.docs con
```

will copy a base template into your */docs* directory.

You can now open the file *conf.py* in */docs* with the editor of your choice and start to adjust settings.

For example you may want to change *ExampleProjectName* with the real name of your project. You can find more information about possible configuration settings in the [Sphinx documentation](#).

Spell-check mr.docs uses a default word-list for spell checking, this list written for [Plone](#) and contains terms which are typical known in Plone but not by spell-checker of dictionaries.

If you want to add missing, new or your own needed terms or words to the spell-check you can add them on top of the document.

```
.. spelling::
    word1
    word2
    ...
```

Changelog

2.1

- Removal of *.mrdocsfunc*

- We added the functionality to copy a base template of a Sphinx configuration
- Added test builds for html [testhtml]
- Added html builds for deployment [html]
- Update Docker image
- Update docs
- Improved spell-check

2.0

- With the release of mr.docs version 2.0 we renamed the function file from *.dockerfunc* to *.mrdocsfunc*.
- Slight modifications to *.mrdocsfunc*, the easiest way for updating from version 1.0 to 2.0 is to remove *.dockerfunc* and download the new one.