
Read the Docs Template Documentation

Release 0.1 alpha

Read the Docs

March 02, 2017

1	Software Documentation Template	3
1.1	Installation	3
1.2	Write you documentation	3
1.3	Editor	4
1.4	Deploy on ReadTheDocs	4
1.5	Contribute	4
1.6	Support and bugs	4
1.7	License	4
2	Project Introduction	5
3	Requisites	7
4	Architecture	9
5	UML Diagrams	11
5.1	Sequence diagrams	11
5.2	Use Case Diagrams	11
6	Code Organisation	13
7	Sustainability	15
8	Requisites	17
9	Developer Manual	19
10	User Manual	21
11	Tutorial	23
12	Appendices	25
12.1	Glossary	25
13	List of contributors	27
14	List of contributors	29
15	History page	31

16 Installation	33
17 Indices and tables	35

Contents:

Software Documentation Template

This is a template to generate software documentation from textual files written in Markdown and/or RestructuredText. The template is compliant with Sphinx and therefore the documentation can be generated in different formats as convenient: HTML, PDF, ePub, Latex, etc.

It has been extended from the the ReadTheDocs template and therefore your can be hosted out-of-the-box on the [ReadTheDocs](<https://readthedocs.org/>) website.

Diagrams written using PlantUML or Graphviz are rendered automatically by [Sphinx](sphinx-doc.org)

![ReadTheDocs Status](<https://readthedocs.org/projects/documentation-template/badge/?version=latest>)

Installation

```
# Create a python virtual environment virtualenv-3.5 venv-py35
```

```
# Load environment source venv-py35/bin/activate
```

```
# Install all dependencies pip install -r requirements.txt
```

```
# To unload environment deactivate
```

Diagrams are generated by Sphinx automatically if they are coded in GraphViz or PlantUML. ReadTheDocs integrates support for both format in their docker images that will go live in production in 2017. PlantUML is included for your convenience as [git-lfs](<https://git-lfs.github.com/>) in ``utils/plantuml.jar``.

Write you documentation

A base structure of a documentation is in the ``doc`` folder and is structured to document a software development project. It is organised in sub folders and a hierarchy of RestructuredText files including or linking other ResT file or markdown.

The document files can be just in ResT format but we haven't found a reasonable support for editing them with a live preview as instead is for Markdown. On the other hand Markdown doesn't support file inclusion that is a substantial limitation.

Therefore we use Markdown for all the documentation file except for the files containing the table of contents or including other docs.

Editor

- PyCharm
- Eclipse

Pycharm is a multi-platform and open source editor with two plugins helping our editing: Markdown split editor with one section for writing code and one for live preview; the GfmBrowser supports page reloading upon change so any time you recompile the documentation you see the final ReadTheDocs pages.

Eclipse has been identified as a multi-platform and open source editor with support for plantUML. Plugin supporting Markdown and ResT editing with live preview should be available but at the time of writing their installation was unsuccessful on Eclipse neon 2.0.

Generate your documentation in any of your preferred formats (html, latex, pdf, epub)

```
cd docs
```

```
# Generate Html documentation make html
```

```
# Open documentation with a Web browser firefox docs/_build/html/index.html
```

Deploy on ReadTheDocs

ReadTheDocs hosts for you version of your documentation that is rebuild automatically everytime you commit a change. Go to <https://readthedocs.org/> and setup your account. If your documentation is not a public git repository, use the [ReadTheDocs manual import](<https://readthedocs.org/dashboard/import/manual/>)

Contribute

Contributions are welcome. Open an Issue in the git repository and/or a git pull request.

Support and bugs

If you are having issues, please let open an issue on this documentation git repository or email contributors.

License

The project is licensed under the BSD license.

Project Introduction

1. Mission and Values
2. Layman summary
3. Destination users and summary of use cases
4. Functionalities
5. Timeplan, milestones and Gantt chart
6. Future directions
7. Funding Bodies
8. Web presence, social media and contacts
9. References

Requisites

Here must be annotated:

- Contractual features
- Environment requirements to allow the software to work
- API and interactions with other software components
- Standard compliancy
- Quality assurance requirements
- Deliverables and timeline

Architecture

Describe here the project architecture in all his components

- data model
- business logic
- Interface with users and other systems
- TODO

UML Diagrams

Sequence diagrams

Here below an example of Sequence diagram. Please refer to [PlatUML Sequence Diagram](<http://plantuml.com/sequence-diagram>) for further options.

Use Case Diagrams

Here below an example of Sequence diagram. Please refer to [PlatUML Sequence Diagram](<http://plantuml.com/sequence-diagram>) for further options.

- Use Cases
- Sequence Diagrams
- Class Diagram

Code Organisation

Here must be annotated:

- Logical Architecture of the project
- Installation procedure
- Modularity
- Design pattern used
- file structure, copyright and license
- coding workflow and git
- testing procedures and quality assurance
- Coding style and conventions

Sustainability

Here must be annotated:

- Growth analysis
- maintainability
- distribution ana packing
- Bug, issue solving and support
- Brand and copyright protection
- Funding Plan and financial model
- ... TODO

Requisites

Here must be annotated:

- Contractual features
- Environment requirements to allow the software to work
- API and interactions with other software components
- Standard compliancy
- Quality assurance requirements
- Deliverables and timeline

Developer Manual

1. Developer Community
2. Code organisation
3. Development Workflow
4. Coding Standards * Syntax convention * Golden Rules and Street Fighters * Test Driven Development
5. Continuous Integration
6. Quality Assurance and Testing * [PEP8 Coding style checker](#)
7. Documentation * in code documentation * ReadTheCode or public documentation
8. Bug report
9. Licensing
10. Suggested Readings

User Manual

1. Use case for end users
2. Functionalities available
3. API or interface description
4. How to use provided functionalities by scenario

Tutorial

1. Installation and test
2. Run a provided demo, example or functional test
3. Extend existing example

Appendices

1. Glossary
2. User agreement documents
3. Deliverable refs
4. Extra documents

Glossary

lemma 1 definition

lemma 2 definition.

List of contributors

Daniele Tartarini, d.tartarini@sheffield.ac.uk

Contributors of the base version of ReadTheDocs.org template are listed on their website.

List of contributors

Daniele Tartarini, d.tartarini@sheffield.ac.uk

Contributors of the base version of ReadTheDocs.org template are listed on their website.

History page

- Project history
- Contribution intensity
- Contributors timeline

Installation

Install the package with pip:

```
$ pip install read-the-docs-template
```

Indices and tables

- `genindex`
- `modindex`
- `search`

L

lemma 1, [25](#)

lemma 2, [25](#)