

Sphinx Documentation Guidelines

Introduction:

Sphinx is a tool that makes it easy to create intelligent and beautiful documentation. This was originally created for documenting Python Projects but C/C++ Projects can also be documented as well. Sphinx generates documents from **reStructuredText** document sources(.rst files). We will focus on python documentation here.

Installation:

To install run the following command:

```
$ pip install Sphinx
```

Sphinx comes with 4 scripts namely **sphinx-quickstart**, **sphinx-autogen**, **sphinx-apidoc**, **sphinx-build** when installed.

Documentation Guidelines to be followed while coding in Python:

Please follow [this](#) for python coding standards.

Below is a sample python module depicting all aspects of code-documentation which can be done while coding with python.

aws_metadata.py

```
#!/usr/bin/env python
'''
This module contains a class which provides API(s) to access AWS instace
metadata. This module should be used only in AWS Instances to work
properly.

Required Module(s): urllib2

'''
from urllib2 import Request, urlopen, URLLError

class AWSMetaData(object):
    '''
    This class provides APIs to retrieve latest metadata from the running
    aws instance

    '''
    def __init__(self):
        '''
        This is the init function of the class AWSMetaData.

        :returns: None
        :rtype: None

        '''
        self.__meta_url = "http://169.254.169.254/latest/meta-data/%s"
        self.__supported_types = ["instance-id"]

    def retrieve_instance_metadata(self, metadata_type):
        '''
```

This method retrieves aws instace metadata.

```
:param metadata_type: should contain type of metadata to retrieve.  
:type metadata_type: string  
:returns: metadata value  
:rtype: string
```

```
'''  
if metadata_type not in self.__supported_types:  
    raise Exception("Invalid Instance Metadata type")  
req = Request(self.__meta_url % (metadata_type))  
try:  
    response = urlopen(req)  
except URLError:  
    raise Exception("Platform not supported: AWS instace needed")  
return response.read()
```

```
def get_aws_instance_id(self):  
    '''  
    This function returns aws instance id from instance metadata.  
  
    :returns: aws instance id  
    :rtype: string  
  
    '''  
    return self.retrieve_instance_metadata("instance-id")
```

```
if __name__ == "__main__":  
    meta_obj = AWSMetaData()
```

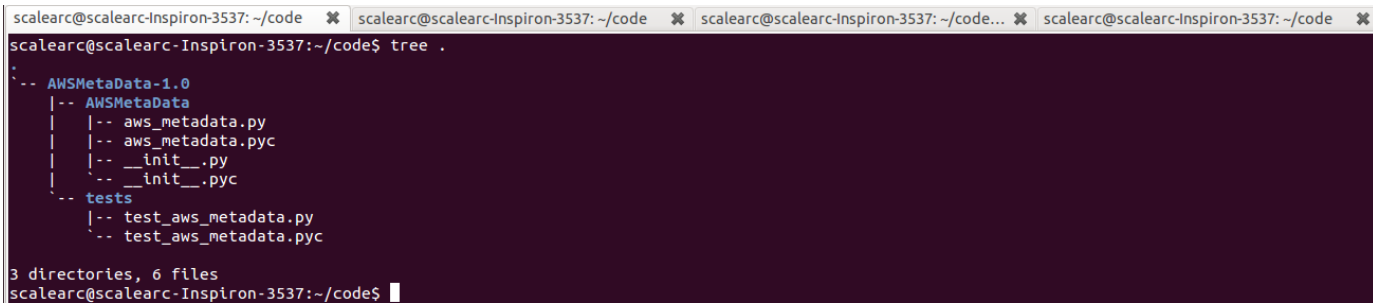
```
print "AWS Instance-id: %s" % (meta_obj.get_aws_instance_id())
```

Step by step guide:

Step- 1 (Setting up documentation environment):

The root directory of Sphinx is a collection of **reStructuredText** document sources is called the **source** directory . This directory also contains the Sphinx configuration file **conf.py**, where all configuration parameters of sphinx documentation like reading sources, formatting and building documents are defined.

Below is the directory structure of my path:

A terminal window screenshot showing the output of the 'tree .' command in a directory. The terminal has a dark purple background. The output shows a directory structure for 'AWSMetaData-1.0' containing subdirectories 'AWSMetaData' and 'tests'. 'AWSMetaData' contains 'aws_metadata.py', 'aws_metadata.pyc', '__init__.py', and '__init__.pyc'. 'tests' contains 'test_aws_metadata.py' and 'test_aws_metadata.pyc'. At the bottom, it says '3 directories, 6 files'.

```
scalearc@scalearc-Inspiron-3537: ~/code
scalearc@scalearc-Inspiron-3537:~/code$ tree .
.
|-- AWSMetaData-1.0
|   |-- AWSMetaData
|   |   |-- aws_metadata.py
|   |   |-- aws_metadata.pyc
|   |   |-- __init__.py
|   |   |-- __init__.pyc
|   |-- tests
|   |   |-- test_aws_metadata.py
|   |   |-- test_aws_metadata.pyc
|
3 directories, 6 files
scalearc@scalearc-Inspiron-3537:~/code$
```

To start a new environment:

```
$ sphinx-quickstart
```

This will start asking questions for different parameters. Make sure to say **yes** to the following:

1. Separate source and build directories (y/n)
2. autodoc: automatically insert docstrings from modules (y/n)
3. coverage: checks for documentation coverage (y/n)
4. viewcode: include links to the source code of documented Python objects (y/n)
5. Create Makefile? (y/n)

```
scalearc@scalearc-Inspiron-3537: ~/code  ScaleArc AWS Instance  scalearc@scalearc-Inspiron-3537: ~/code  scalearc@scalearc-Inspiron-3537: ~/code...

scalearc@scalearc-Inspiron-3537:~/code$ sphinx-quickstart
Welcome to the Sphinx 1.4.5 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Enter the root path for documentation.
> root path for the documentation [.] :
You have two options for placing the build directory for Sphinx output.
Either, you use a directory ".build" within the root path, or you separate
"source" and "build" directories within the root path.
> separate source and build directories (y/n) [n] : y
Inside the root directory, two more directories will be created: ".templates"
for custom HTML templates and ".static" for custom stylesheets and other static
files. You can enter another prefix (such as ".") to replace the underscore.
> name prefix for templates and static dir [...] :
The project name will occur in several places in the built documentation.
> Project name: AWSMetadata
> Author name(s) : Shubhrajyoti
Sphinx has the notion of a "version" and a "release" for the
software. Each version can have multiple releases. For example, for
python the version is something like 2.5 or 3.0, while the release is
something like 2.5.1 or 3.0.1. If you don't need this dual structure,
just set both to the same value.
> Project version: 1.0
> Project release: [1.0] :
If the documents are to be written in a language other than English,
you can select a language here by its language code. Sphinx will then
translate text that it generates into that language.
For a list of supported codes, see
http://sphinx-doc.org/config.html#confval-language.
> Project language: en
The file name suffix for source files. Commonly, this is either ".txt"
or ".rst". Only files with this suffix are considered documents.
> Source file suffix [.rst] :
One document is special in that it is considered the top node of the
"documents tree", that is, it is the root of the hierarchical structure
of the documents. Normally, this is "index", but if your "index"
document is a custom template, you can also set this to another filename.
Name of your master document (file) [index] :
Sphinx can also add configuration for your output:
Do you want to use the web builder? (y/n) [n] :
Please indicate if you want to use one of the following Sphinx extensions:
> autodoc: automatically insert docstrings from modules (y/n) [n] : y
> doctest: automatically test code fragments in doctest blocks (y/n) [n] :
> intersphinx: link between Sphinx documentation of different projects (y/n) [n] :
> todo: write "todo" entries that can be shown or hidden on build (y/n) [n] :
> coverage: checks for documentation coverage (y/n) [n] : y
> imgmath: include math, rendered as PNG or SVG images (y/n) [n] :
> mathjax: include math, rendered as the browser's MathJax (y/n) [n] :
> (i18n): internationalization of content based on config values (y/n) [n] :
> viewcode: include links to the source code of documented Python objects (y/n) [n] : y
> alabaster: create a new file to publish the document on alabaster (y/n) [n] :
A Makefile and a Windows command file can be generated for you so that you
only have to run e.g. "make html" instead of invoking sphinx-build
directly.
> Create Makefile? (y/n) [y] : y
> Create Windows command file? (y/n) [n] : n
Creating file ./source/conf.py.
Creating file ./source/index.rst.
Creating file ./Makefile.
Finished: An initial directory structure has been created.
You should now populate your master file ./source/index.rst and create other documentation
```

After sphinx-quickstart following is the directory structure;

Step- 2 (Generating reStructuredText document (rst) files from the source code directory):

```
$ sphinx-apidoc -e -f -d 10 -o source/ <Path to source code directory>
```

```
scalearc@scalearc-Inspiron-3537: ~/code  scalearc@scalearc-Inspiron-3537: ~/code

scalearc@scalearc-Inspiron-3537:~/code$ sphinx-apidoc -e -f -d 10 -o source/ AWSMetaData-1.0/
Creating file source/AWSMetaData.aws_metadata.rst.
Creating file source/AWSMetaData.rst.
Creating file source/modules.rst.
scalearc@scalearc-Inspiron-3537:~/code$ tree .
.
|-- AWSMetaData-1.0
|   |-- AWSMetaData
|   |   |-- aws_metadata.py
|   |   |-- aws_metadata.pyc
|   |   |-- __init__.py
|   |   |-- __init__.pyc
|   |-- tests
|   |   |-- test_aws_metadata.py
|   |   |-- test_aws_metadata.pyc
|-- build
|-- Makefile
-- source
    |-- AWSMetaData.aws_metadata.rst
    |-- AWSMetaData.rst
    |-- conf.py
    |-- index.rst
    |-- modules.rst
    |-- _static
    -- _templates

7 directories, 12 files
scalearc@scalearc-Inspiron-3537:~/code$
```

Step- 3 (Link generated rst file to index.rst):

Below is the original index.rst file.

source/index.rst

```
.. AWSMetaData documentation master file, created by
sphinx-quickstart on Tue Aug 23 19:29:32 2016.
You can adapt this file completely to your liking, but it should at
least
    contain the root `toctree` directive.
Welcome to AWSMetaData's documentation!
=====
Contents:
.. toctree::
    :maxdepth: 2

Indices and tables
=====
* :ref:`genindex`
* :ref:`modindex`
* :ref:`search`
```

To link <package_name>.rst here AWSMetaData.rst; insert an entry with proper indentation in the "Contents" block. Below is the index.rst after linking. here.

source/index.rst

```
.. AWSMetaData documentation master file, created by
sphinx-quickstart on Tue Aug 23 19:29:32 2016.
You can adapt this file completely to your liking, but it should at
least
    contain the root `toctree` directive.
Welcome to AWSMetaData's documentation!
=====
Contents:
.. toctree::
    :maxdepth: 2

    AWSMetaData.rst

Indices and tables
=====
* :ref:`genindex`
* :ref:`modindex`
* :ref:`search`
```

If you are generating multiple .rst then the modules.rst can be put in the index.rst instead of individual module names.

The Welcome page can also be customised by changing the contents of source/index.rst.

Step- 4 (Setting system library path to pick our source code path):

This step is needed to because Sphinx needs to import all the source codes and then generate the document from them. So we need to set library path in source/conf.py by un-commenting

1. import sys
2. sys.path.insert(0, "<path to source code directory>")

By default Sphinx generates the documents using

html_theme = 'alabaster'

pygments_style = 'sphinx'

If you want to create documentation which is themed like <https://docs.python.org/2.7/> modify both the values to

html_theme = 'default'

pygments_style = 'default'

source/conf.py

```
# -*- coding: utf-8 -*-
#
# AWSMetaData documentation build configuration file, created by
# sphinx-quickstart on Tue Aug 23 19:29:32 2016.
#
# This file is execfile()d with the current directory set to its
# containing dir.
#
# Note that not all possible configuration values are present in this
# autogenerated file.
#
# All configuration values have a default; values that are commented out
# serve to show the default.
# If extensions (or modules to document with autodoc) are in another
# directory,
# add these directories to sys.path here. If the directory is relative to
# the
# documentation root, use os.path.abspath to make it absolute, like shown
# here.
#
# import os
import sys
sys.path.insert(0, '/home/scalearc/code/AWSMetaData-1.0/')
# -- General configuration -----
# If your documentation needs a minimal Sphinx version, state it here.
#
# needs_sphinx = '1.0'
# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.coverage',
    'sphinx.ext.viewcode',
]
# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']
# The suffix(es) of source filenames.
# You can specify multiple suffix as a list of string:
#
# source_suffix = ['.rst', '.md']
```

```

source_suffix = '.rst'
# The encoding of source files.
#
# source_encoding = 'utf-8-sig'
# The master toctree document.
master_doc = 'index'
# General information about the project.
project = u'AWSMetaData'
copyright = u'2016, Shubhrajyoti'
author = u'Shubhrajyoti'
# The version info for the project you're documenting, acts as replacement
for
# |version| and |release|, also used in various other places throughout the
# built documents.
#
# The short X.Y version.
version = u'1.0'
# The full version, including alpha/beta/rc tags.
release = u'1.0'
# The language for content autogenerated by Sphinx. Refer to documentation
# for a list of supported languages.
#
# This is also used if you do content translation via gettext catalogs.
# Usually you set "language" from the command line for these cases.
language = None
# There are two options for replacing |today|: either, you set today to
some
# non-false value, then it is used:
#
# today = ''
#
# Else, today_fmt is used as the format for a strftime call.
#
# today_fmt = '%B %d, %Y'
# List of patterns, relative to source directory, that match files and
# directories to ignore when looking for source files.
# This patterns also effect to html_static_path and html_extra_path
exclude_patterns = []
# The reST default role (used for this markup: `text`) to use for all
# documents.
#
# default_role = None
# If true, '()' will be appended to :func: etc. cross-reference text.
#
# add_function_parentheses = True
# If true, the current module name will be prepended to all description
# unit titles (such as .. function:).
#
# add_module_names = True
# If true, sectionauthor and moduleauthor directives will be shown in the
# output. They are ignored by default.
#
# show_authors = False

```

```

# The name of the Pygments (syntax highlighting) style to use.
pygments_style = 'default'
# A list of ignored prefixes for module index sorting.
# modindex_common_prefix = []
# If true, keep warnings as "system message" paragraphs in the built
documents.
# keep_warnings = False
# If true, `todo` and `todoList` produce output, else they produce nothing.
todo_include_todos = False
# -- Options for HTML output -----
# The theme to use for HTML and HTML Help pages.  See the documentation for
# a list of builtin themes.
#
html_theme = 'default'
# Theme options are theme-specific and customize the look and feel of a
theme
# further.  For a list of options available for each theme, see the
# documentation.
#
# html_theme_options = {}
# Add any paths that contain custom themes here, relative to this
directory.
# html_theme_path = []
# The name for this set of Sphinx documents.
# "<project> v<release> documentation" by default.
#
# html_title = u'AWSMetaData v1.0'
# A shorter title for the navigation bar.  Default is the same as
html_title.
#
# html_short_title = None
# The name of an image file (relative to this directory) to place at the
top
# of the sidebar.
#
# html_logo = None
# The name of an image file (relative to this directory) to use as a
favicon of
# the docs.  This file should be a Windows icon file (.ico) being 16x16 or
32x32
# pixels large.
#
# html_favicon = None
# Add any paths that contain custom static files (such as style sheets)
here,
# relative to this directory. They are copied after the builtin static
files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']
# Add any extra paths that contain custom files (such as robots.txt or
# .htaccess) here, relative to this directory. These files are copied
# directly to the root of the documentation.
#

```



```

# html_extra_path = []
# If not None, a 'Last updated on:' timestamp is inserted at every page
# bottom, using the given strftime format.
# The empty string is equivalent to '%b %d, %Y'.
#
# html_last_updated_fmt = None
# If true, SmartyPants will be used to convert quotes and dashes to
# typographically correct entities.
#
# html_use_smartypants = True
# Custom sidebar templates, maps document names to template names.
#
# html_sidebars = {}
# Additional templates that should be rendered to pages, maps page names to
# template names.
#
# html_additional_pages = {}
# If false, no module index is generated.
#
# html_domain_indices = True
# If false, no index is generated.
#
# html_use_index = True
# If true, the index is split into individual pages for each letter.
#
# html_split_index = False
# If true, links to the reST sources are added to the pages.
#
# html_show_sourcelink = True
# If true, "Created using Sphinx" is shown in the HTML footer. Default is
True.
#
# html_show_sphinx = True
# If true, "(C) Copyright ..." is shown in the HTML footer. Default is
True.
#
# html_show_copyright = True
# If true, an OpenSearch description file will be output, and all pages
will
# contain a <link> tag referring to it. The value of this option must be
the
# base URL from which the finished HTML is served.
#
# html_use_opensearch = ''
# This is the file name suffix for HTML files (e.g. ".xhtml").
# html_file_suffix = None
# Language to be used for generating the HTML full-text search index.
# Sphinx supports the following languages:
#   'da', 'de', 'en', 'es', 'fi', 'fr', 'hu', 'it', 'ja'
#   'nl', 'no', 'pt', 'ro', 'ru', 'sv', 'tr', 'zh'
#
# html_search_language = 'en'
# A dictionary with options for the search language support, empty by

```

```

default.
# 'ja' uses this config value.
# 'zh' user can custom change `jieba` dictionary path.
#
# html_search_options = {'type': 'default'}
# The name of a javascript file (relative to the configuration directory)
that
# implements a search results scorer. If empty, the default will be used.
#
# html_search_scorer = 'scorer.js'
# Output file base name for HTML help builder.
htmlhelp_basename = 'AWSMetaDatadoc'
# -- Options for LaTeX output -----
latex_elements = {
    # The paper size ('letterpaper' or 'a4paper').
    #
    # 'papersize': 'letterpaper',
    # The font size ('10pt', '11pt' or '12pt').
    #
    # 'pointsize': '10pt',
    # Additional stuff for the LaTeX preamble.
    #
    # 'preamble': '',
    # Latex figure (float) alignment
    #
    # 'figure_align': 'htbp',
}
# Grouping the document tree into LaTeX files. List of tuples
# (source start file, target name, title,
#  author, documentclass [howto, manual, or own class]).
latex_documents = [
    (master_doc, 'AWSMetaData.tex', u'AWSMetaData Documentation',
     u'Shubhrajyoti', 'manual'),
]
# The name of an image file (relative to this directory) to place at the
top of
# the title page.
#
# latex_logo = None
# For "manual" documents, if this is true, then toplevel headings are
parts,
# not chapters.
#
# latex_use_parts = False
# If true, show page references after internal links.
#
# latex_show_pagerefs = False
# If true, show URL addresses after external links.
#
# latex_show_urls = False
# Documents to append as an appendix to all manuals.
#
# latex_appendices = []

```

```

# It false, will not define \strong, \code,      itleref, \crossref ... but
only
# \sphinxstrong, ..., \sphinxtitleref, ... To help avoid clash with user
added
# packages.
#
# latex_keep_old_macro_names = True
# If false, no module index is generated.
#
# latex_domain_indices = True
# -- Options for manual page output -----
# One entry per manual page. List of tuples
# (source start file, name, description, authors, manual section).
man_pages = [
    (master_doc, 'awsmetadata', u'AWSMetaData Documentation',
     [author], 1)
]
# If true, show URL addresses after external links.
#
# man_show_urls = False
# -- Options for Texinfo output -----
# Grouping the document tree into Texinfo files. List of tuples
# (source start file, target name, title, author,
#  dir menu entry, description, category)
texinfo_documents = [
    (master_doc, 'AWSMetaData', u'AWSMetaData Documentation',
     author, 'AWSMetaData', 'One line description of project.',
     'Miscellaneous'),
]
# Documents to append as an appendix to all manuals.
#
# texinfo_appendices = []
# If false, no module index is generated.
#
# texinfo_domain_indices = True
# How to display URL addresses: 'footnote', 'no', or 'inline'.
#
# texinfo_show_urls = 'footnote'
# If true, do not generate a @detailmenu in the "Top" node's menu.
#

```

```
# texinfo_no_detailmenu = False
```

Step- 5 (Building the final documentation):

To build the documentation in html format run

```
$ make html
```

If No Makefile is generated while sphinx-quickstart, then run

```
$ sphinx-build -b html source/ build/
```

Note: There are many formats like html, pickle, json, epub, latex, latexpdf, xml, text etc. supported by Sphinx which can be passed to above the make command or sphinx-build -b option.

```
scalearc@scalearc-Inspiron-35... ✖ ScaleArc AWS Instance ✖ scalearc@scalearc-Inspiron-35... ✖ scalearc@scalearc-Inspiron-35... ✖ scalearc@scalearc-Inspiron-35... ✖
scalearc@scalearc-Inspiron-3537:~/code$ make html
sphinx-build -b html -d build/doctrees source build/html
Running Sphinx v1.4.5
making output directory...
loading pickled environment... not yet created
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 3 source files that are out of date
updating environment: 3 added, 0 changed, 0 removed
reading sources... [100%] modules
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] modules
generating indices... genindex py-modindex
highlighting module code... [100%] AWSMetaData.aws_metadata
writing additional pages... search
copying static files... done
copying extra files... done
dumping search index in English (code: en) ... done
dumping object inventory... done
build succeeded.

Build finished. The HTML pages are in build/html.
scalearc@scalearc-Inspiron-3537:~/code$
```

Note: You could find ImportError while building html which must be fixed by setting proper sys.path in source/conf.py to get proper documentation.

Start httpd server and point your browser to build/html to view the documentation.

Note: To start a temporary web server with the current path as Document Root; enter in the command line

```
$ python -m SimpleHTTPServer 8000
```

This will start a web server with your LAN IP at port 8000.

The documentation of the package AWSMetaData is below.

Google... Mail-S... Edit-S... sphinx... sphinx... using S... Docum... python... Correc... Ubunt... Scripts... Welco... x +

localhost:8000/build/html/ sphinx apidoc

AWSMetaData 1.0 documentation » next | modules | index

Welcome to AWSMetaData's documentation!

Table Of Contents

- Welcome to AWSMetaData's documentation! Indices and tables

Next topic

AWSMetaData package

This Page

Show Source

Quick search

Go

Contents:

- AWSMetaData package
 - Submodules
 - Module contents

Indices and tables

- Index
- Module Index
- Search Page

AWSMetaData 1.0 documentation » next | modules | index

© Copyright 2016, Shubhrajyoti. Created using [Sphinx](#) 1.4.5.

Google... Mail-S... Edit-S... sphinx... sphinx... using S... Docum... python... Correc... Ubunt... Scripts... AWSM... x +

localhost:8000/build/html/AWSMetaData.aws_metadata.html sphinx apidoc

AWSMetaData 1.0 documentation » AWSMetaData package » previous | modules | index

AWSMetaData.aws_metadata module

Previous topic

AWSMetaData package

This Page

Show Source

Quick search

Go

This module contains class which provide methods to access AWS Instance metadata. This module should be used only in AWS Instances to work properly.

Required Module(s): urllib2

class `AWSMetaData.aws_metadata.AWSMetaData` [\[source\]](#)

Bases: `object`

This class retrieves latest metadata from the running aws Instance

get_aws_instance_id() [\[source\]](#)

This function returns aws Instance id from instance metadata.

Returns: aws Instance Id
Return type: string

retrieve_instance_metadata(metadata_type) [\[source\]](#)

This method retrieves aws Instance metadata.

Parameters: `metadata_type` (*string*) – should contain type of metadata to retrieve.
Returns: metadata value
Return type: string

AWSMetaData 1.0 documentation » AWSMetaData package » previous | modules | index

© Copyright 2016, Shubhrajyoti. Created using [Sphinx](#) 1.4.5.

To add more modules into this document tree repeat step - 2 to step - 5.

Click [here](#) for detailed documentation of Sphinx.

