github-docs-index Documentation

Release 0.1.2

Jason Antman

Apr 22, 2018

Contents

build passing Generate a single-page index of documentation hosted in one or more GitHub organizations on github.com and/or one or more GitHub Enterprise instances.

This package is intended for organizations that host their documentation alongside code on GitHub (including GitHub Enterprise) and need a convenient single-page index to help people find things. It's a small, opinionated, and purpose-specific tool, originally written so that my team could have a master index of our documentation (spread across github.com, two GitHub Enterprises, Confluence, and another intranet solution) without having to remember to add every new repository.

Full documentation is available on ReadTheDocs: http://github-docs-index.readthedocs.io/en/latest/

CHAPTER 1

Features

- Outputs docutils-ready reStructuredText for processing or conversion to a variety of formats.
- Supports a manually-curated "quick links" section at the top of the page, which can include arbitrary non-GitHub URLs
- Iterates repositories in any number of Organizations (or Users) on github.com and/or any number of GitHub Enterprise instances. Allows blacklisting or whitelisting repositories per-org/user and blacklisting organizations.
- Output sorted alphabetically and by last commit/update date.
- Configurable to show only repositories with GitHub Pages, a Repository URL, repositories with a README file longer than a specified length, repositories with a description, or all repositories.
- Option to ignore forks.
- Python API which allows injecting additional links to the chronological and alphabetical lists.
- GitHub tokens taken from environment variables.
- Configurable title, subtitle/header and footer; subtitle and footer can be overridden by environment variables.

CHAPTER 2

Requirements

- Python 2.7 or 3.4+ (currently tested with 2.7, 3.4, 3.5, 3.6)
- VirtualEnv and `pip` (recommended installation method; your OS/distribution should have packages for these)

CHAPTER 3

Installation

It's recommended that you install into a virtual environment (virtualenv / venv). See the virtualenv usage documentation for information on how to create a venv.

```
pip install github-docs-index
```

CHAPTER 4

Configuration

Configuration is provided by a YAML file; see *Example Configuration* for a detailed example. The YAML file must have a mapping/hash/dict at the top level. Keys are as follows:

- **title**, string, the title of the index rst document
- **footer**, *optional*, string, a footer line to include at the end of the index rst document. This configuration option is overridden by the `GITHUB_DOCS_FOOTER` environment variable if set.
- **subtitle**, *optional*, string, a subtitle/header line to include below the title of the index rst document. This configuration option is overridden by the `GITHUB_DOCS_SUBTITLE` environment variable if set.
- **githubs**, array/list of mappings/dicts specifying the github instances to query. Each item in the array has the following structure:
  - **token_env_var**, string, name of the environment variable that contains the Personal Access Token for this github instance
  - **url**, *optional*, string, URL to this GitHub instance for GitHub Enterprise instances. If not specified, defaults to `https://api.github.com`
  - **orgs**, *optional*, array of string organization names to scan repositories in. If neither this option nor "orgs" is specified, default to scanning all repos of orgs that the authenticated user belongs to.
  - **users**, *optional*, array of string user names to scan repositories in. If neither this option nor "orgs" is specified, default to scanning all repos of orgs that the authenticated user belongs to.
  - **whitelist_repos**, *optional*, array of string repository slugs (full names) in "owner_name/repo_name" format. If specified, only these repos will be included regardless of other configuration for this GitHub instance.
  - **blacklist_repos**, *optional*, array of string repository slugs (full names) in "owner_name/repo_name" format to never include in the output documentation index.
  - **blacklist_orgs**, *optional*, array of string organization names to ignore when scanning repos.
- **ignore_forks**, *optional*, boolean, default False. If True, do not include any repos that are forks in the listing.
- **quick_links**, *optional*, array/list of mappings/dicts specifying manually-curated links to add to the "Quick Links" section at the top of the document. Each item in the array has the following structure:

  - **title** string, the title/text of the link
  - **url** string, the URL to link to
  - **description**, *optional*, string description to output after the link to the repo

- **repo_criteria**, array/list of strings or mappings determining which repos to include in the listing and what URL to set for them. For each repo, these are evaluated in order and the first match wins; if none match, the repo is not added to the list. Possible options are:
  - `homepage`, string: if present, use the Repository Homepage URL (at the top of the repo page, next to the description) as the link. Only matches repos with a Homepage set.
  - `github_pages`, string: if present, use the repo's GitHub Pages URL as the link. Only matches repos with GitHub Pages enabled.
  - `readme: N`, mapping/dict where "readme" is a string and N is an integer: match repos with a readme file of size N or greater, and link to the repo's main HTML URL (github web UI URL)
  - `description`, string: match any repo with a description set, and link to the repo's main HTML URL (github web UI URL)
  - `all` match any/all repos, and link to the repo's main HTML URL (github web UI URL)

CHAPTER 5

CLI Usage

Usage via the command line is straightforward for common use cases. The reStructuredText output is printed to STDOUT, and can be redirected to a file. For example, assuming you've already installed the package as shown above, and using `example_config.yaml` as an example:

```
# these next three environment variable names are specified in example_config.yaml
export GITHUB_TOKEN=yourToken
export GHE_TOKEN=anotherToken
export OTHER_GHE_TOKEN=yetAnotherToken
github-docs-index config.yaml > index.rst
```

This rst file can be converted to the format of your choice with any tool that understands reStructuredText input. For example, it can be converted to HTML using `rst2html.py` from the docutils package (`pip install docutils`):

```
rst2html.py --report=4 index.rst > index.html
```

CHAPTER 6

Setting Subtitle and Footer

The optional subtitle (line below the title) and footer (line at the bottom of the document) can also be specified as environment variables. For example:

```
export GITHUB_DOCS_SUBTITLE="This document was automatically generated at $(date)"
export GITHUB_DOCS_FOOTER="This document was generated by Jenkins: ${BUILD_URL}"
github-docs-index config.yaml > index.rst
```

CHAPTER 7

Example Output

You can see an example of the actual HTML output for my own github user in the source tree at *Example Output*.

CHAPTER 8

Python Usage

github-docs-index can also be imported and used in other Python code. This can be especially useful for doing something with the raw rst output; here is an example that replicates the functionality of the above CLI examples in a single Python script:

```python
#!/usr/bin/env python

# for generating the rst
from github_docs_index.config import Config
from github_docs_index.index_generator import GithubDocsIndexGenerator

# for docutils rst -> HTML
from docutils import core
from docutils.writers.html4css1 import Writer, HTMLTranslator


# this replicates "github-docs-index config.yaml" at the command line
g = GithubDocsIndexGenerator(Config('config.yaml'))
rst_string = g.generate_index()

# the code below here replicates "rst2html.py --report=4 index.rst > index.html"


class HTMLFragmentTranslator(HTMLTranslator):

    def __init__(self, document):
        HTMLTranslator.__init__(self, document)
        self.head_prefix = ['', '', '', '', '']
        self.body_prefix = []
        self.body_suffix = []
        self.stylesheet = []

    def astext(self):
        return ''.join(self.body)
```

```
html_fragment_writer = Writer()
html_fragment_writer.translator_class = HTMLFragmentTranslator

with open('index.html', 'wb') as fh:
    fh.write(core.publish_string(rst_string, writer=html_fragment_writer))
print('Output written to: index.html')
```

8.1 Adding Documentation From Other Sources

It's also possible via the Python API to include aribtrary documents from sources other than GitHub in the index; they will be sorted into the chronological and alphabetical lists along with the GitHub repositories. This can be helpful if you have other sources of documentation such as an Intranet or Wiki that you can programmatically query. The only requirement is that each document has a URL, title, date (generally a created/modified/updated date) and optional short description. The *github_docs_index.index_generator.GithubDocsIndexGenerator.generate_index* method takes an optional `additional_links` argument which is a list of instances of a subclass of *github_docs_index.index_link.IndexLink*. So long as the instances implement the three properties of `IndexLink`, they will be included in the documentation index. Here is a short, contrived example based on the code above which includes two other documents with hard-coded dates, titles and URLs; the `generate_additional_links()` function could be switched out for one which queries your alternate documentation stores and returns similar output.

```
#!/usr/bin/env python3

from datetime import datetime, timezone
from github_docs_index.config import Config
from github_docs_index.index_generator import GithubDocsIndexGenerator
from github_docs_index.index_link import IndexLink


class StaticLink(IndexLink):
    """This class implements the three property methods in IndexLink"""

    def __init__(self, title, url, sort_datetime, description=''):
        self._title = title
        self._url = url
        self._sort_datetime = sort_datetime
        self._description = description

    @property
    def sort_datetime(self):
        return self._sort_datetime

    @property
    def sort_name(self):
        return self._title.lower()

    @property
    def rst_line(self):
        r = '`%s <%s>`_' % (self._title, self._url)
        if self._description is not None and self._description.strip() != '':
            r += ' - ' + self._description
        return r


def generate_additional_links():
    return [
        StaticLink(
            'Some Document', 'http://example.com/someDocument',
            datetime(2017, 6, 3, 12, 34, 41, tzinfo=timezone.utc),
            description='this is a document'
        ),
        StaticLink(
            'Other Document', 'http://example.com/otherDocument',
            datetime(2018, 8, 12, 19, 24, 53, tzinfo=timezone.utc),
            description='this is another document'
        )
    ]


# this replicates "github-docs-index config.yaml" at the command line
g = GithubDocsIndexGenerator(Config('config.yaml'))
rst_string = g.generate_index(additional_links=generate_additional_links())

with open('index.rst', 'w') as fh:
    fh.write(rst_string)
```

CHAPTER 9

Bugs and Feature Requests

Bug reports and feature requests are happily accepted via the GitHub Issue Tracker. Pull requests are welcome. Issues that don't have an accompanying pull request will be worked on as my time and priority allows.

CHAPTER 10

Development

To install for development:

1. Fork the github-docs-index repository on GitHub
2. Create a new branch off of master in your fork.

```
$ virtualenv github-docs-index
$ cd github-docs-index && source bin/activate
$ pip install -e git+git@github.com:YOURNAME/github-docs-index.git@BRANCHNAME
↪#egg=github-docs-index
$ cd src/github-docs-index
```

The git clone you’re now in will probably be checked out to a specific commit, so you may want to `git checkout BRANCHNAME`.

10.1 Guidelines

- pep8 compliant with some exceptions (see pytest.ini)
- 100% test coverage with pytest (with valid tests)

10.2 Testing

Testing is done via pytest, driven by tox.

- testing is as simple as:
  - `pip install tox`
  - `tox`
- If you want to pass additional arguments to pytest, add them to the tox command line after “–”. i.e., for verbose pytext output on py27 tests: `tox -e py27 -- -v`

10.3 Release Checklist

1. Open an issue for the release; cut a branch off master for that issue.
2. Confirm that there are CHANGES.rst entries for all major changes.
3. Ensure that Travis tests passing in all environments.
4. Ensure that test coverage is no less than the last release (ideally, 100%).
5. Increment the version number in github-docs-index/version.py and add version and release date to CHANGES.rst, then push to GitHub.
6. Confirm that README.rst renders correctly on GitHub.
7. Upload package to testpypi:
   - Make sure your ~/.pypirc file is correct (a repo called `test` for https://test.pypi.org/)
   - `rm -Rf dist`
   - `python setup.py sdist bdist_wheel`
   - `twine upload -r test dist/*`
   - Check that the README renders at https://test.pypi.org/project/github-docs-index
8. Create a pull request for the release to be merged into master. Upon successful Travis build, merge it.
9. Tag the release in Git, push tag to GitHub:
   - tag the release. for now the message is quite simple: `git tag -s -a X.Y.Z -m 'X.Y.Z released YYYY-MM-DD'`
   - push the tag to GitHub: `git push origin X.Y.Z`

11. TravisCI will cut the release and upload to PyPI.

CHAPTER 11

Contents

11.1 Example Configuration

```
title: Example github-docs-index Output
footer: footer of the index document here; can be overridden via GITHUB_DOCS_FOOTER environment variable
subtitle: This is the subtitle I configured, which can be overridden via GITHUB_DOCS_SUBTITLE environment variable
githubs:
  # this first one is implicitly github.com
  - token_env_var: SOME_GITHUB_TOKEN
    # if the "users" and "orgs" keys are not specified or are empty lists,
    # we default to looking at all Organizations that the GITHUB_TOKEN user
    # is a member of
    blacklist_repos: [two/puppet, three/chef]
    blacklist_orgs: [badOrg]
  - token_env_var: GITHUB_TOKEN
    users: [jantman]
  - url: https://ghe.example.com
    token_env_var: GHE_TOKEN
    orgs: [one, two, three]
  - url: https://ghe.other.com
    token_env_var: OTHER_GHE_TOKEN
    orgs: [one, four]
    users: [someone]
    whitelist_repos: [one/a, four/b, someone/c, someone/d]
ignore_forks: true
quick_links:
  - title: My Blog
    url: http://blog.jasonantman.com/
  - title: My Resume
    url: http://resume.jasonantman.com/
    description: source for this is at https://github.com/jantman/resume
  - title: Our Intranet
```

```
    url: https://intranet.example.com
  - title: Our ticketing system
    description: no idea why this is linked in a docs index
    url: https://tickets.example.com
  - title: Big Project
    description: some big project
    url: https://project.example.com
  - title: Important Repo
    url: https://ghe.other.com/four/reponame
# this is the order used to determine whether to show a repository in the list or not.
# if a repo doesn't meet any of these, it isn't shown. Otherwise, we use these in the␣
↪order
# specified to determine if we show a repo, and if so, what we show about it
repo_criteria:
  - homepage # if present, use the Repository Homepage URL (at the top of the repo␣
↪page, next to the description) as the link
  - github_pages # otherwise, if present, use the repo's GitHub Pages URL as the link
  - readme: 100 # for any repo with a README >= 100 bytes, link to the repo itself
  - description # for any repo with a description set, link to the repo itself
  - all # link to ALL repos, even if they don't have any of the above
```

11.2 Example github-docs-index Output

This is the subtitle I configured, which can be overridden via GITHUB_DOCS_SUBTITLE environment variable

< *Chronological Index* | *Alphabetical Index* >

11.2.1 Quick Links

- My Blog
- My Resume - source for this is at https://github.com/jantman/resume

< *top.* | *Quick Links* | *Alphabetical Index* >

11.2.2 Chronological Index (most recently updated first)

- github-docs-index - Generate a single-page index of documentation hosted in one or more GitHub organizations on github.com and/or one or more GitHub Enterprise instances.
- aws-docs-history - Historical versions of AWS documentation
- awslimitchecker - A script and python module to check your AWS service limits and usage via boto.
- blog - my blog
- python-package-skeleton - Skeleton for a python package, complete with tox/Travis-CI, Landscape.io, codecov, etc.
- biweeklybudget - Responsive Flask/SQLAlchemy personal finance app, specifically for biweekly budgeting.
- xfinity-usage - Python/selenium script to get Xfinity bandwidth usage
- misc-scripts - A collection of my standalone scripts to small/quick for their own repos. All kinds of useful stuff.
- arch-pkgbuilds - My PKGBUILDs for Arch Linux

- pizero-gpslog - Raspberry Pi Zero gpsd logger with status LEDs
- jiveapi - Simple and limited Python client for Jive collaboration software ReST API v3, along with utilities for massaging HTML to display better on Jive.
- vagrant-r10k - Vagrant middleware plugin to retrieve puppet modules using r10k.
- repostatus.org - A standard to easily communicate to humans and machines the development/support and usability status of software repositories/projects.
- vault-aws-creds - Python helper to export Vault-provided temporary AWS creds into the environment
- pypi-download-stats - Calculate detailed download stats and generate HTML and badges for PyPI packages
- nagios-scripts - A collection of my Nagios check scripts, and related stuff.
- serverspec-extended-types - A set of extended types for ServerSpec 2.x
- versionfinder - Python package to find the version of another package, whether installed via pip, setuptools or git
- scantool.net - ScanTool.net's GPLv2 cross-platform OBDII software, "scantool.net"
- archautorepo - Automatic Arch Linux package rebuilds and repository creation.
- pydnstest - Python tool for testing DNS changes (add, remove, rename, change records) against a staging DNS server, verifying the same changes against production, or confirming that a record returns the same result in both environments.
- SwitchSNMP - A collection of PHP classes and scripts to communicate with and control network devices via SNMP.
- rspec-matcher-num-times - A rspec matcher for include or match a specified number of times
- rspec-matcher-hash-item - Rspec matchers for dealing with hashes and their content
- crondigest - A cron job wrapper to send batched digest email and notify of missed jobs
- franz-recipe-pushover - meetfranz.com (Franz 5) recipe for Pushover
- jantman.github.io - My personal homepage
- ecsjobs - A scheduled job wrapper for ECS, focused on email reporting and adding docker exec and local command abilities.
- puppet-archlinux-macbookretina - Puppet module and accompanying documentation to install/setup Arch linux on a MacBook Pro Retina 11,4
- webhook2lambda2sqs - Generate code and manage infrastructure for receiving webhooks with AWS API Gateway and pushing to SQS via Lambda.
- tf-vol-attachment-cycle - Proof-of-concept for Terraform issue 16237
- tf-consul-lock-reproduce - code to reproduce the terraform "consul lock was lost" error
- resume - My resume. see http://resume.jasonantman.com
- puppet-facter-facts - a collection of facts for user with Facter and Puppet
- puppet-archlinux-workstation - Puppet module for configuring various aspects of an Arch Linux workstation/desktop.
- workstation-bootstrap - My r10k/puppet based workstation bootstrapping and configuration
- php-nagios-xml - This is a PHP script that parses the Nagios status.dat file into an array, and then outputs that array as XML. There is also a PHP module written in C to do the same task, and based on the same code.
- piface-webhooks - Python script/daemon to read RPi PiFace inputs and run a webhook when they change

- RPyMostat-sensor - The temperature sensor component of RPyMostat
- python-obd - Some OBD code for Python. Old and crappy, but putting it here in case I need it again.
- nodemeister - NodeMeister is an External Node Classifier (ENC) for Puppet, written in Python using the Django framework. It supports hierarchical groups, exclusions and overrides, and aims to provide a fully-functional ENC for those more comfortable with Python.
- s3sfe - Sync a list of files to S3, using server-side encryption with customer-provided keys
- multibindadmin - A web-based (PHP) tool for managing BIND DNS zone files, with builtin understanding of split-horizon DNS with NAT (i.e. a name can have an internal address and an external address).
- RPyMostat - A python-based intelligent home thermostat, targeted at the RaspberryPi.
- gw2copilot - A Python-based GuildWars2 helper tool
- autosimulationcraft - A python script to run SimulationCraft reports for World of Warcraft characters when their gear/stats/level/etc. changes.
- pi2graphite - RaspberryPi-targeted app to send 1wire temperature & wifi stats to graphite.
- versionfinder-test-pkg - test package for https://github.com/jantman/versionfinder acceptance tests
- userscripts - Some of my UserScripts
- pymonitoringapi - Python module to interact with Nagios and Icinga over the web.
- git-clone-sync - Script to keep git clones in sync with origin and upstream
- PUP1244_test - tests for PUP-1244
- NetworkLunchbox - My Network Lunchbox project (originally I was going to name it palantir) aims to provide a ready-to-use image for Soekris net4801 (and possibly 4501) to make them usable as general-purpose network troubleshooting devices. Specifically, the intent is enable administrators of large, complex or geographically dispersed networks to be able to install a Soekris device with one interface on a network reachable from their workstation and use the other two interfaces as entry points into unreachable or problematic networks.
- BoardTest - Testing various GitHub kanban boards
- docker-debian-mongodb24 - Debian-based Docker image with MongoDB 2.4 server.
- RPyMostat-ui - Web UI for RPyMostat Project.
- RPyMostat-control - Relay control portion of the RPyMostat project.
- php-ems-tools - Web-based Patient Care Report and organization management software for volunteer Emergency Medical Services. Largely unusable, just an example.
- rpymostat-common - Common libraries shared by packages in the RPyMostat project.
- specfiles - A collection of my RPM spec files
- rpmbuild-vagrant-boxes - A collection of Vagrantfiles for boxes to build RPMs
- reviewboard-scripts - Collection of scripts that use the ReviewBoard python client (RBClient) to perform various tasks with ReviewBoard.
- puppet-nodemeister - A puppet module to install, configure and run the NodeMeister ENC.
- kvmdashclient - Python client script for the kvmdash application.
- kvmdash - kvmdash is a simple Python daemon and web app to collect information about libvirt-controlled qemu/kvm guest VMs running on standalone hosts, ans present the information on a single web page (with a simple API). It will also include a companion Puppet module for installation, and Facter facts that make use of the data.

- grumble - What do you want to grumble about today?
- gitlab-scripts - Some scripts to help administering and migrating to GitLab
- ec2-utils-el7 - Amazon Linux ec2-utils for EL7
- boxen - boxen.github.com for my Mac
- CFDash - A simple dashboard for CloudFormation stack metrics from CloudWatch.
- pypuppetdb-daily-report - Daily run summary report for PuppetDB, written in Python using nedap's pypuppetdb module.
- rebuildbot - Rebuildbot re-runs builds of your inactive projects.
- networkmapper - Tools to map physical data network connetions.
- rackman - An old PHP project to track machines' locations in a rack, diagram them, and preview changes.
- Android-CycleSys - A really old, bad, incomplete implementation of the Cycle System task management for Android. Only staying around because I'm sentimental.
- boguskeys - SSH keys that I want github to allow, but I don't actually want to give access to my stuff.
- cloaked-hipster - GitHub suggested this name, so I just had to use it. I'll find something to put here eventually.
- helga-queue - A simple helga IRC bot plugin to let you manage a short queue (FIFO) of strings (eg. todo items).
- howtopuppet - howtopuppet.com - a community FAQ site for Puppet and related technologies
- ideas-for-projects - Various ideas I've had for different projects, which I haven't started yet.
- kbdtest - wxPython tool for testing keyboards on Linux/Mac/Unix operating systems
- PHPsa - This is really the skeleton of a defunct project from a few years ago. In retrospect, PHP was almost certainly the wrong language to use for this (compared to pretty much any modern web language, but especially because of its scoping issues). I'm leaving it here only to pay homage to the idea of it, which still might have some usefulness.
- puppet-python-test - Test of a puppet-python module
- pytest-batches - pytest plugin to run marked tests in batches, and skip the rest if a batch fails
- tuxtruck - Aborted attempt at a Python/Linux-based CarPC platform. Some of the OBD and GPS code might be useful, and there's Python code to read the accelerometer on a SunSPOT.
- TuxTruck-wxPython - TuxTruck-wxPython was a project to creat a wxPython/Linux-based in-vehicle PC platform. It currently has an audio player and not much else.
- tuxostat - abandoned linux/python thermostat project
- xmlfinal - My final project from my Building Data-Driven Websites undergrad course. Here for nostalgia.

< *top.* | *Quick Links* | *Chronological Index* >

11.2.3 Alphabetical Index

- Android-CycleSys - A really old, bad, incomplete implementation of the Cycle System task management for Android. Only staying around because I'm sentimental.
- arch-pkgbuilds - My PKGBUILDs for Arch Linux
- archautorepo - Automatic Arch Linux package rebuilds and repository creation.
- autosimulationcraft - A python script to run SimulationCraft reports for World of Warcraft characters when their gear/stats/level/etc. changes.

- aws-docs-history - Historical versions of AWS documentation
- awslimitchecker - A script and python module to check your AWS service limits and usage via boto.
- biweeklybudget - Responsive Flask/SQLAlchemy personal finance app, specifically for biweekly budgeting.
- blog - my blog
- BoardTest - Testing various GitHub kanban boards
- boguskeys - SSH keys that I want github to allow, but I don't actually want to give access to my stuff.
- boxen - boxen.github.com for my Mac
- CFDash - A simple dashboard for CloudFormation stack metrics from CloudWatch.
- cloaked-hipster - GitHub suggested this name, so I just had to use it. I'll find something to put here eventually.
- crondigest - A cron job wrapper to send batched digest email and notify of missed jobs
- docker-debian-mongodb24 - Debian-based Docker image with MongoDB 2.4 server.
- ec2-utils-el7 - Amazon Linux ec2-utils for EL7
- ecsjobs - A scheduled job wrapper for ECS, focused on email reporting and adding docker exec and local command abilities.
- franz-recipe-pushover - meetfranz.com (Franz 5) recipe for Pushover
- git-clone-sync - Script to keep git clones in sync with origin and upstream
- github-docs-index - Generate a single-page index of documentation hosted in one or more GitHub organizations on github.com and/or one or more GitHub Enterprise instances.
- gitlab-scripts - Some scripts to help administering and migrating to GitLab
- grumble - What do you want to grumble about today?
- gw2copilot - A Python-based GuildWars2 helper tool
- helga-queue - A simple helga IRC bot plugin to let you manage a short queue (FIFO) of strings (eg. todo items).
- howtopuppet - howtopuppet.com - a community FAQ site for Puppet and related technologies
- ideas-for-projects - Various ideas I've had for different projects, which I haven't started yet.
- jantman.github.io - My personal homepage
- jiveapi - Simple and limited Python client for Jive collaboration software ReST API v3, along with utilities for massaging HTML to display better on Jive.
- kbdtest - wxPython tool for testing keyboards on Linux/Mac/Unix operating systems
- kvmdash - kvmdash is a simple Python daemon and web app to collect information about libvirt-controlled qemu/kvm guest VMs running on standalone hosts, ans present the information on a single web page (with a simple API). It will also include a companion Puppet module for installation, and Facter facts that make use of the data.
- kvmdashclient - Python client script for the kvmdash application.
- misc-scripts - A collection of my standalone scripts to small/quick for their own repos. All kinds of useful stuff.
- multibindadmin - A web-based (PHP) tool for managing BIND DNS zone files, with builtin understanding of split-horizon DNS with NAT (i.e. a name can have an internal address and an external address).
- nagios-scripts - A collection of my Nagios check scripts, and related stuff.

- NetworkLunchbox - My Network Lunchbox project (originally I was going to name it palantir) aims to provide a ready-to-use image for Soekris net4801 (and possibly 4501) to make them usable as general-purpose network troubleshooting devices. Specifically, the intent is enable administrators of large, complex or geographically dispersed networks to be able to install a Soekris device with one interface on a network reachable from their workstation and use the other two interfaces as entry points into unreachable or problematic networks.
- networkmapper - Tools to map physical data network connetions.
- nodemeister - NodeMeister is an External Node Classifier (ENC) for Puppet, written in Python using the Django framework. It supports hierarchical groups, exclusions and overrides, and aims to provide a fully-functional ENC for those more comfortable with Python.
- php-ems-tools - Web-based Patient Care Report and organization management software for volunteer Emergency Medical Services. Largely unusable, just an example.
- php-nagios-xml - This is a PHP script that parses the Nagios status.dat file into an array, and then outputs that array as XML. There is also a PHP module written in C to do the same task, and based on the same code.
- PHPsa - This is really the skeleton of a defunct project from a few years ago. In retrospect, PHP was almost certainly the wrong language to use for this (compared to pretty much any modern web language, but especially because of its scoping issues). I'm leaving it here only to pay homage to the idea of it, which still might have some usefulness.
- pi2graphite - RaspberryPi-targeted app to send 1wire temperature & wifi stats to graphite.
- piface-webhooks - Python script/daemon to read RPi PiFace inputs and run a webhook when they change
- pizero-gpslog - Raspberry Pi Zero gpsd logger with status LEDs
- PUP1244_test - tests for PUP-1244
- puppet-archlinux-macbookretina - Puppet module and accompanying documentation to install/setup Arch linux on a MacBook Pro Retina 11,4
- puppet-archlinux-workstation - Puppet module for configuring various aspects of an Arch Linux workstation/desktop.
- puppet-facter-facts - a collection of facts for user with Facter and Puppet
- puppet-nodemeister - A puppet module to install, configure and run the NodeMeister ENC.
- puppet-python-test - Test of a puppet-python module
- pydnstest - Python tool for testing DNS changes (add, remove, rename, change records) against a staging DNS server, verifying the same changes against production, or confirming that a record returns the same result in both environments.
- pymonitoringapi - Python module to interact with Nagios and Icinga over the web.
- pypi-download-stats - Calculate detailed download stats and generate HTML and badges for PyPI packages
- pypuppetdb-daily-report - Daily run summary report for PuppetDB, written in Python using nedap's pypuppetdb module.
- pytest-batches - pytest plugin to run marked tests in batches, and skip the rest if a batch fails
- python-obd - Some OBD code for Python. Old and crappy, but putting it here in case I need it again.
- python-package-skeleton - Skeleton for a python package, complete with tox/Travis-CI, Landscape.io, codecov, etc.
- rackman - An old PHP project to track machines' locations in a rack, diagram them, and preview changes.
- rebuildbot - Rebuildbot re-runs builds of your inactive projects.

- repostatus.org - A standard to easily communicate to humans and machines the development/support and usability status of software repositories/projects.
- resume - My resume. see http://resume.jasonantman.com
- reviewboard-scripts - Collection of scripts that use the ReviewBoard python client (RBClient) to perform various tasks with ReviewBoard.
- rpmbuild-vagrant-boxes - A collection of Vagrantfiles for boxes to build RPMs
- RPyMostat - A python-based intelligent home thermostat, targeted at the RaspberryPi.
- rpymostat-common - Common libraries shared by packages in the RPyMostat project.
- RPyMostat-control - Relay control portion of the RPyMostat project.
- RPyMostat-sensor - The temperature sensor component of RPyMostat
- RPyMostat-ui - Web UI for RPyMostat Project.
- rspec-matcher-hash-item - Rspec matchers for dealing with hashes and their content
- rspec-matcher-num-times - A rspec matcher for include or match a specified number of times
- s3sfe - Sync a list of files to S3, using server-side encryption with customer-provided keys
- scantool.net - ScanTool.net's GPLv2 cross-platform OBDII software, "scantool.net"
- serverspec-extended-types - A set of extended types for ServerSpec 2.x
- specfiles - A collection of my RPM spec files
- SwitchSNMP - A collection of PHP classes and scripts to communicate with and control network devices via SNMP.
- tf-consul-lock-reproduce - code to reproduce the terraform "consul lock was lost" error
- tf-vol-attachment-cycle - Proof-of-concept for Terraform issue 16237
- tuxostat - abandoned linux/python thermostat project
- tuxtruck - Aborted attempt at a Python/Linux-based CarPC platform. Some of the OBD and GPS code might be useful, and there's Python code to read the accelerometer on a SunSPOT.
- TuxTruck-wxPython - TuxTruck-wxPython was a project to creat a wxPython/Linux-based in-vehicle PC platform. It currently has an audio player and not much else.
- userscripts - Some of my UserScripts
- vagrant-r10k - Vagrant middleware plugin to retrieve puppet modules using r10k.
- vault-aws-creds - Python helper to export Vault-provided temporary AWS creds into the environment
- versionfinder - Python package to find the version of another package, whether installed via pip, setuptools or git
- versionfinder-test-pkg - test package for https://github.com/jantman/versionfinder acceptance tests
- webhook2lambda2sqs - Generate code and manage infrastructure for receiving webhooks with AWS API Gateway and pushing to SQS via Lambda.
- workstation-bootstrap - My r10k/puppet based workstation bootstrapping and configuration
- xfinity-usage - Python/selenium script to get Xfinity bandwidth usage
- xmlfinal - My final project from my Building Data-Driven Websites undergrad course. Here for nostalgia.

< *top.* | *Quick Links* | *Chronological Index* | *Alphabetical Index* >

footer of the index document here; can be overridden via GITHUB_DOCS_FOOTER environment variable

11.3 github_docs_index

11.3.1 github_docs_index package

11.3.1.1 Submodules

github_docs_index.config module

github_docs_index.github_instance module

github_docs_index.index_document module

class `github_docs_index.index_document.`**`IndexDocument`**(*config*)

Bases: `object`

Class to represent the actual index document

`add_indexlinks`(*links*)

`doc_template = '\n.. _top:\n\n{title}\n\n< `Chronological Index <#chrono>`_ | `Alphabe`

`generate_rst`()

github_docs_index.index_generator module

class `github_docs_index.index_generator.`**`GithubDocsIndexGenerator`**(*config*)

Bases: `object`

`generate_index`(*additional_links=[]*)

Main entry point to query GitHub, retrieve repository information, generate the index document and return rST.

Parameters **`additional_links`** (`list` of *`IndexLink`*) – Optional list of additional *`IndexLink`* instances to include in the documentation index.

Returns generated rST index document

github_docs_index.index_link module

class `github_docs_index.index_link.`**`IndexLink`**

Bases: `object`

Base class to represent documents that will be linked in the index page.

`rst_line`

Return the rsStructuredText link/line for this link; this will be used as an item in the document list. This should be a hyperlink to the document URL with the appropriate title, optionally followed by a hyphen and the (~1 sentence) description if available. i.e.:

title

or:

title - Description goes here.

Returns rst link to document, with optional description

Return type str

sort_datetime

Return a datetime.datetime instance to be used when sorting documents by creation/update time.

Returns creation/modification time of the document for chronological sorting

Return type datetime.datetime

sort_name

Return a lower-case string to be used as the document title/name when sorting documents alphabetically.

Returns document name/title used for sorting documents alphabetically. Should be lower-case.

Return type str

github_docs_index.quick_link module

class github_docs_index.quick_link.**QuickLink**(*title*, *url*, *description=None*)

Bases: object

as_dict

rst_line

github_docs_index.repo_link module

class github_docs_index.repo_link.**RepoLink**(*owner_name*, *repo_name*, *url*, *description=None*, *last_commit_datetime=None*)

Bases: *github_docs_index.index_link.IndexLink*

Represents one repository that should be linked to in the docs index.

as_dict

description

full_name

lower_case_full_name

rst_line

set_description(*desc*)

set_last_commit_datetime(*dt*)

sort_datetime

sort_name

url

github_docs_index.runner module

github_docs_index.version module

11.4 Changelog

11.4.1 0.1.2 (2018-04-22)

- Fix bug where only public Organization membership was detected for a user.

11.4.2 0.1.1 (2018-04-21)

- Fix package description in setup.py

11.4.3 0.1.0 (2018-04-21)

- Initial release

CHAPTER 12

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index