jenkins-job-builder-addons Documentation
Release 1.0.5

Mike Buzzetti

August 13, 2015

Contents

Contents:

jenkins-job-builder-addons

A set of addons for jenkins job builder

- Free software: Apache license
- Documentation: https://jenkins-job-builder-addons.readthedocs.org

1.1 Features

- Supports job folders
- Supports Build Pipeline View
- Supports Delivery Pipeline View

1.2 Install

- As of August 12th 2015 this requires two special branches. One for python-jenkins and one for jenkins-job-builder

1. Get python-jenkins

- git clone git://git.openstack.org/stackforge/python-jenkins
- cd python-jenkins
- git fetch https://review.openstack.org/stackforge/python-jenkins refs/changes/85/180185/19 && git checkout FETCH_HEAD
- python setup.py install

2. Get jenkins-job-builder

- git clone git://git.openstack.org/openstack-infra/jenkins-job-builder
- cd jenksin-job-builder
- git fetch https://review.openstack.org/openstack-infra/jenkins-job-builder refs/changes/07/134307/2 && git checkout FETCH_HEAD
- python setup.py install

3. Get jenkins-job-builder-addons

- git clone git@github.com:jimbydamonk/jenkins-job-builder-addons.git

- cd jenkins-job-builder-addons

- python setup.py install

Installation

At the command line:

```
$ easy_install jenkins-job-builder-addons
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv jenkins-job-builder-addons
$ pip install jenkins-job-builder-addons
```

Usage

To use jenkins-job-builder-addons in a project:

```python
import jenkins_jobs_addons
```

3.1 Folders

The Folder plugin handles creating CloudBeesFolder Jenkins Jobs. You man specify `folder` in the `project-type` attribute of the Job definition.

Requires the Jenkins CloudBees Folder Plugin.

> **arg str primary-view** Name of the default view to show for this folder.
>
> **arg list health-metrics** A list of metrics to use as a health check. Must be one of the following:
>
> > • **worst-child-health-metric**

Job example:

```yaml
name: folder_test
project-type: folder
primary-view: "Test Primary View"
health-metrics:
  - worst-child-health-metric
```

class `jenkins_jobs_addons.folders.`**`Folder`**(*registry*)

> Bases: `jenkins_jobs.modules.base.Base`
>
> Class built off Base
>
> **`root_xml`**(*data*)
>
> > Called after data is parsed. Returns xml representing a job :arg dict data: the YAML data structure
>
> **`sequence = 0`**

3.2 Views

Views show job status.

Component: views

> **Macro** views
>
> **Entry Point** jenkins_jobs.views

class `jenkins_jobs_addons.views.`**`Views`**`(`*registry*`)`

> Bases: `jenkins_jobs.modules.base.Base`
>
> **`component_list_type`** = 'views'
>
> **`component_type`** = 'view'
>
> **`gen_xml`** (*parser*, *xml_parent*, *data*)
>
> **`sequence`** = 20

`jenkins_jobs_addons.views.`**`all_view`**`(`*parser*, *xml_parent*, *data*`)`

> All view
>
> > **Parameters**
> >
> > - **`filter-executors`** (*bool*) – only those build executors will be shown that could execute the jobs in this view.
> >
> > - **`filter-queue`** (*bool*) – only jobs in this view will be shown in the queue.
> >
> > - **`folder`** (*bool*) – Wether or not this view is in a folder.
>
> Example:
>
> ```
> views:
>   - all:
>       filter-executors: false
>       filter-queue: false
> ```

`jenkins_jobs_addons.views.`**`build_pipeline_view`**`(`*parser*, *xml_parent*, *data*`)`

> Build Pipeline View requires the Jenkins Build Pipeline Plugin.
>
> > **Parameters**
> >
> > - **`filter-executors`** (*bool*) – only those build executors will be shown that could execute the jobs in this view.
> >
> > - **`filter-queue`** (*bool*) – only jobs in this view will be shown in the queue.
> >
> > - **`folder`** (*bool*) – Wether or not this view is in a folder.
> >
> > - **`name`** (*str*) – The name of this view.
> >
> > - **`first-job`** (*str*) – Select the initial or parent Job in the build pipeline view.
> >
> > - **`display-number-of-builds`** (*int*) – Select the number of build pipelines to display in the view.
> >
> > - **`build-view-title`** (*str*) – The title of this view.
> >
> > - **`console-output-link-style`** (*str*) – One the following:
> >
> >   - **This Window**
> >
> >   - **New Window**
> >
> >   - **Light Box** (default)
> >
> > - **`trigger-only-latest-job`** (*bool*) – Select this option to restrict the display of a Trigger button to only the most recent successful build pipelines. This option will also limit retries to just unsuccessful builds of the most recent build pipelines.

- **True: Only the most recent successful builds displayed on the** view will have a manual trigger button for the next build in the pipeline.

- **False: All successful builds displayed on the view will have a** manual trigger button for the next build in the pipeline.

- **always-allow-manual-trigger** (*bool*) – Select this option if you want to be able to execute again a successful pipeline step. If the build is parameterized, this will re-execute the step using the same parameter values that were used when it was previously executed.

- **start-with-parameters** (*bool*) – Select this option if you want to show the pipeline definition header in the pipeline view. If this option is not selected, then a pipeline that has never been run will not show any details about its jobs and appear like a blank form. Job details will only appear after the pipeline has been run at least once.

- **show-pipeline-parameters-in-header** (*bool*) – Select this option if you want to display the parameters used to run the latest successful job in the pipeline's project headers.

- **show-pipeline-parameters** (*bool*) – Select this option if you want to display the parameters used to run the first job in each pipeline's revision box.

- **refresh-frequency** (*bool*) – Frequency at which the Build Pipeline Plugin updates the build cards in seconds

- **css-url** (*str*) – Link to override style sheet

Example:

```
views:
  - build_pipeline:
      filter-executors: false
      filter-queue: false
      folder: true
      first-job: testjob
      name: testbuild
      build-view-title: "Test Build Pipeline"
      display-number-of-builds: 10
      console-output-link-style: "Light Box"
      csss-url: ""
      trigger-only-latest-job: false
      always-allow-manual-trigger: false
      show-pipeline-parameters: false
      show-pipeline-parameters-in-header: false
      start-with-parameters: false
      refresh-frequency: 3
      show-pipeline-definition-in-headers: false
```

jenkins_jobs_addons.views.**delivery_pipeline_view**(*parser*, *xml_parent*, *data*)
    Delivery Pipeline View requires the Jenkins Delivery Pipeline Plugin.

        Parameters

- **filter-executors** (*bool*) – only those build executors will be shown that could execute the jobs in this view.

- **filter-queue** (*bool*) – only jobs in this view will be shown in the queue.

- **folder** (*bool*) – Wether or not this view is in a folder.

- **name** (*str*) – The name of this view.

- **components** (*dict*) – The components (jobs) for this pipeline:

> > – **name (str): Name of the pipeline, usually the name of the** component or product.
> >
> > – **first-job (str): First job in the pipeline. Usually the** build/compile job. The build number/build display name will be used as the version in later tasks or stages. If using folders, it should be a full path to the job.

- **number-of-pipelines** (*int*) – Number of pipelines instances shown for each pipeline.

- **show-aggregated-pipeline** (*bool*) – Show an aggregated view where each stage shows the latest version being executed.

- **number-of-columns** (*int*) – Number of columns used for showing pipelines. Useful for multiple components in the view to show them beside each others.

- **sorting** (*int*) – How to sort the pipeline in the view. Only applicable for several pipelines. Can be sorted by latest activity or by name.

- **update-interval** (*int*) – How often will the view be updated in seconds.

- **allow-pipeline-start** (*bool*) – Start a new pipeline build.

- **allow-manual-triggers** (*bool*) – If a task is manual (Build other projects (manual step) from Build Pipeline Plugin, show a button.

- **allow-rebuild** (*bool*) – Rebuild a task.

- **show-avatars** (*str*) – Show avatars pictures instead of names of the people involved in a pipeline instance. Use the Avatar Plugin or the Gravatar Plugin. or similar to set avatar picture for contributors.

- **show-changes** (*bool*) – Show SCM change log for the first job in the pipeline. If Repository browser is configured, link to change will be created to the repository browser.

- **show-description** (*bool*) – Show build description connected to a task.

- **show-promotions** (*bool*) – Show promotions from the *Promoted Builds Plugin. <https://wiki.jenkins-ci.org/display/JENKINS/Promoted+Builds+Plugin>_*

- **show-total-buildtime** (*bool*) – Show total build time of a pipeline. If there are multiple routes in a pipeline, total build time is calculated as the sum of the build times in the longest route.

- **css-url** (*str*) – Possibility to override CSS for the normal view. Enter the full url to the custom CSS.

- **fullscreen-css-url** (*str*) – Possibility to override CSS for the fullscreen view. Enter the full url to the custom CSS.

- **regexp-first-jobs** (*list*) – Find jenkins job matching regular expression. ^build-(.+?)-project

Example:

```
views:
  - delivery_pipeline:
      filter-executors: false
      filter-queue: false
      folder: true
      components:
        - name: Build
          first-job: testjob
      name: testbuild
      build-view-title: "Test Build Pipeline"
      number-of-pipelines: 3
```

```
        show-aggregated-pipeline: false
        number-of-columns: 1
        sorting: none
        show-avatars: false
        update-interval: 1
        show-changes: false
        allow-manual-triggers: false
        show-total-buildtime: false
        allow-rebuild: false
        allow-pipeline-start: false
        show-description: false
        show-promotions: false
```

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/jimbydamonk/jenkins-job-builder-addonss/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" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.1.4 Write Documentation

jenkins-job-builder-addonscould always use more documentation, whether as part of the official jenkins-job-builder-addonsdocs, 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/jimbydamonk/jenkins-job-builder-addonss/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 *jenkins-job-builder-addonss* for local development.

1. Fork the *jenkins-job-builder-addonss* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/jenkins-job-builder-addonss.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 jenkins-job-builder-addonss
$ cd jenkins-job-builder-addonss/
$ 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 jenkins-job-builder-addonss tests
$ python setup.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 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/jimbydamonk/jenkins-job-builder-addonss/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 tests.test_jenkins-job-builder-addonss
```

Credits

5.1 Development Lead

- Mike Buzzetti <mike.buzzetti@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

0.1.0 (2015-01-11)

- First release on PyPI.

Indices and tables

- genindex
- modindex
- search

j
jenkins_jobs_addons.folders, 7
jenkins_jobs_addons.views, 7

A

all_view() (in module jenkins_jobs_addons.views), 8

B

build_pipeline_view() (in module jenkins_jobs_addons.views), 8

C

component_list_type (jenkins_jobs_addons.views.Views attribute), 8

component_type (jenkins_jobs_addons.views.Views attribute), 8

D

delivery_pipeline_view() (in module jenkins_jobs_addons.views), 9

F

Folder (class in jenkins_jobs_addons.folders), 7

G

gen_xml() (jenkins_jobs_addons.views.Views method), 8

J

jenkins_jobs_addons.folders (module), 7
jenkins_jobs_addons.views (module), 7

R

root_xml() (jenkins_jobs_addons.folders.Folder method), 7

S

sequence (jenkins_jobs_addons.folders.Folder attribute), 7

sequence (jenkins_jobs_addons.views.Views attribute), 8

V

Views (class in jenkins_jobs_addons.views), 8