Jenkins
User Conference

Orchestrating Your Jenkins Pipelines With Python and Jenkins API

London, 23 & 24 June 2015

whoami

Pradeepto Bhattacharya / @pradeepto

Clogeny
An MSys Company

Jenkins User Conference

Agenda

#jenkinsconf

- Why do we need this library?
- Installation
- Library Modules
- Code Examples
- Documentations
- Q & A

Jenkins User Conference

#jenkinsconf

#jenkinsconf

Jenkins User Conference

The Motivation

Jenkins User Conference

Why?

#jenkinsconf

- Sometimes you want more control over how you want run your jobs.
- It is true that you have many plugins to so many things like - copy artifacts, run child jobs etc. No doubt about it.
- But then there are times you have to develop systems or CI pipelines where you need more fine grained control.
- Thanks to the developers of Jenkins, there is an excellent REST API.
- Open source being open source, some good dudes have written a Python library encapsulating the REST API
- This talk is about this library and what can we do with it.

Jenkins
User Conference

#jenkinsconf

#jenkinsconf

Jenkins
User Conference

Installation

Jenkins User Conference

Installation

#jenkinsconf

- pip install jenkinsapi
- easy_install jenkinsapi
- sudo apt-get install python-jenkinsapi

- Note : There are other similar libraries. I found this one best for all my use cases. It definitely was the most exhaustive.

Jenkins User Conference

#jenkinsconf

#jenkinsconf

Jenkins User Conference

Modules and Code

Jenkins
User Conference

Main modules

#jenkinsconf

- Jenkins : The Jenkins instance
- Job(s) : Represents Jenkins jobs.
- User API : Consists of helpful high-level functions
- Build : This module encapsulates the single run of a Jenkins job
- Artifact : Artifacts are created by Jenkins build. This module encapsulates that.
- Node : Encapsulates Jenkins Node

Connect to Jenkins

#jenkinsconf

```
from jenkinsapi.jenkins import Jenkins

def get_jenkins_instance():
    return Jenkins('http://10.10.20.100:8080')
```

Jenkins User Conference

Jobs

#jenkinsconf

```
>>> jenkins = get_jenkins_instance()
>>> jenkins.keys()
>>> ['Beefy Job', 'Simple Job']
>>> jobs = jenkins.get_jobs()
>>> for job in jobs:
...        print job
...
('Beefy Job', <jenkinsapi.job.Job Beefy Job>)
('Simple Job', <jenkinsapi.job.Job Simple Job>)
```

Jobs

```
>>> job = jenkins.get_job('Simple Job')
>>> job.get_description()
'A very simple job'
>>> job.is_enabled()
True
>>> job.is_running()
False
>>> job.get_last_stable_buildnumber()
29
```

Jobs

#jenkinsconf

```
>>> job = jenkins.get_job('Beefy Job')
>>> job.get_last_failed_buildnumber()
3
>>> job.get_last_completed_build()
<jenkinsapi.build.Build Beefy Job #53>
>>> job.invoke()
<jenkinsapi.invocation.Invocation object at 0x7f0f75733050>
>>> job.is_running()
True
```

Jobs

#jenkinsconf

```
>>> job.get_scm_type()
'git'
>>> job.get_scm_branch()
['*/master']
>>> job.get_scm_url()
['http://github.com/jenkinsci/mesos-plugin']
```

Jobs

#jenkinsconf

```
>>> job.disable()
>>> job.is_enabled()
False
>>> job.enable()
>>> job.is_enabled()
True
```

Jenkins User Conference

Jobs

#jenkinsconf

- 'get_first_build', 'get_first_buildnumber', 'get_jenkins_obj', 'get_last_build', 'get_last_buildnumber', 'get_last_completed_build', 'get_last_completed_buildnumber', 'get_last_failed_buildnumber', 'get_last_good_build', 'get_last_good_buildnumber', 'get_last_stable_build', 'get_last_stable_buildnumber', 'get_next_build_number', 'get_params', 'get_params_list', 'get_revision_dict', 'get_scm_branch', 'get_scm_type', 'get_scm_url', 'get_upstream_job_names', 'get_upstream_jobs', 'has_queued_build', 'invoke', 'is_enabled', 'is_queued', 'is_queued_or_running', 'is_running'

User API

#jenkinsconf

```
from jenkinsapi.api import *
>>> get_latest_build('http://10.10.20.100:8080','Simple Job')
<jenkinsapi.build.Build Simple Job #29>
>>> get_latest_complete_build('http://10.10.20.100:8080','Beefy Job')
<jenkinsapi.build.Build Beefy Job #53>
```

Build

#jenkinsconf

```
>>> build = job.get_build(54)
>>> build.is_running()
True
>>> build.get_revision()
'b0c30d15b4c0ac55d003db7533c00e889f74891e'
>>> build.name
'Beefy Job #57'
>>> build.get_status()
'SUCCESS'
```

Jenkins User Conference

Build

#jenkinsconf

```
>>> build.get_console()
'Started by user anonymous\nBuilding on master in workspace /var/lib/jenkins/workspace/Beefy Job\n > git rev-parse --is-inside-work-tree # timeout=10\nFetching changes from the remote Git repository\n > git config remote.origin.url http://github.com/jenkinsci/mesos-plugin # timeout=10\nFetching upstream changes from http://github.com/jenkinsci/mesos-plugin\n > git --version # timeout=10\n > git -c core.askpass=true fetch --tags --progress http://github.com/jenkinsci/mesos-plugin +refs/heads/*:refs/remotes/origin/*\n > git rev-parse refs/remotes/origin/master^{commit} # timeout=10\n > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10\nChecking out Revision b0c30d15b4c0ac55d003db7533c00e889f74891e (refs/remotes/origin/master)\n > git config core.sparsecheckout # timeout=10\n > git checkout -f b0c30d15b4c0ac55d003db7533c00e889f74891e\n > git rev-list b0c30d15b4c0ac55d003db7533c00e889f74891e # timeout=10\n[Beefy Job] $ /bin/sh -xe /tmp/hudson1394675615697321597.sh\n+ set -x\n+ echo This job takes a lot of time and consumes a lot of resources.\nThis job takes a lot of time and consumes a lot of resources.\n+ which python\n/usr/bin/python\n+ python --version\nPython 2.7.6\n+ sleep 180\n+ echo Done building beefy job.\nDone building beefy job.\nFinished: SUCCESS\n'
```

Build

- 'block', 'block_until_complete', 'buildno', 'get_actions', 'get_artifact_dict', 'get_artifacts', 'get_console', 'get_data', 'get_downstream_builds', 'get_downstream_job_names', 'get_downstream_jobs', 'get_duration', 'get_jenkins_obj', 'get_master_build', 'get_master_build_number', 'get_master_job', 'get_master_job_name', 'get_matrix_runs', 'get_number', 'get_result_url', 'get_resultset', 'get_revision', 'get_revision_branch', 'get_status', 'get_timestamp', 'get_upstream_build', 'get_upstream_build_number', 'get_upstream_job', 'get_upstream_job_name', 'has_resultset', 'is_good', 'is_running'

Artifacts

```
>>> job =  jenkins.get_job('Simple Job')
>>> build = job.get_build(33)
>>> build.get_artifact_dict()
{'artifact.txt': <jenkinsapi.artifact.Artifact http://10.10.20.100:8080/job/Simple%20Job/33/artifact/artifact.txt>}
>>> build.get_artifact_dict()['artifact.txt'].get_data()
'This is artifact\n'
```

Nodes

#jenkinsconf

```
>>> for node in jenkins.get_nodes().keys():
...               print node
...
master
Big
Small
>>> small.is_online()
True
```

Jenkins User Conference

Plugins

#jenkinsconf

```
>>> jenkins.get_plugins().keys()
['git-client', 'matrix-auth', 'mesos', 'maven-plugin', 'javadoc', 'external-monitor-job', 'ant', 'ssh-slaves', 'pam-auth', 'windows-slaves', 'git', 'scm-api', 'subversion', 'antisamy-markup-formatter', 'ldap', 'junit', 'mailer', 'credentials', 'translation', 'ssh-credentials', 'matrix-project', 'cvs', 'script-security']
```

Exceptions

- This library comes with bunch of built-in exceptions. I highly recommend you use them in your code and do stuff as you want in case you catch those whilst running your CI pipelines.

- JenkinsAPIException, UnknownJob, UnknownView, UnknownNode, UnknownPlugin, NoBuildData, NoResults ...

Other things you could ...

#jenkinsconf

- Create Jobs
- Create Views
- Monitor and query builds/jobs/nodes
- Search and manipulate artifacts

Jenkins User Conference

Documentation

#jenkinsconf

- Some (very few) of the modules are documented.
- There are a few examples in documentation.
- Great opportunity to contribute to this wonderful library.
- https://github.com/salimfadhley/jenkinsapi

Questions?

pradeeptob@gmail.com

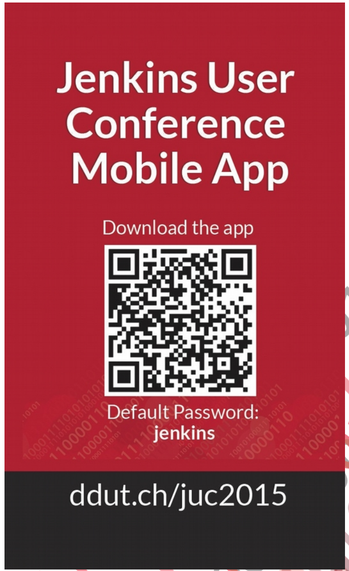
Agenda

CloudBees and organisers of JUC Europe 2015
Contributors of Jenkins, Mesos, Mesos plugin
My Family
Kalpak, Shikha, Lenin
Girija and Rupali (my partners in CI related crimes)

Please Share Your Feedback

#jenkinsconf

- Did you find this session valuable?
- Please share your thoughts in the
- Jenkins User Conference Mobile App.
- Find the session in the app and click
- on the feedback area.


Jenkins User Conference Mobile App

Download the app

Default Password: jenkins

ddut.ch/juc2015