
AutoJenkins Documentation

Release 1.1.1

Carles Barrobés

Aug 17, 2017

Contents

1	Introduction to AutoJenkins	3
1.1	AutoJenkins as API	3
1.2	AutoJenkins from the Command Line	4
1.3	More Info	6
2	autojenkins.jobs	7
3	Indices and tables	9
	Python Module Index	11

AutoJenkins is a tool to automate or remote-control Jenkins. You can e.g. create and delete build jobs, trigger builds, read latest build results, etc.

Contents:

Introduction to AutoJenkins

AutoJenkins was written to handle automation (remote control) of Jenkins tasks. Includes a class `autojenkins.Jenkins` that you can use to drive Jenkins.

Things you can do with it:

- Copy a job (e.g. from a template job)
- Delete a job
- Obtain the `config.xml` file for that job
- Trigger building of a job
- Obtain latest execution results
- ...

AutoJenkins may be used as an API or as a command-line tool.

AutoJenkins as API

Sample use:

```
from autojenkins import Jenkins

j = Jenkins('http://jenkins.pe.local')
# You can add verify_ssl_cert=False argument to the instantiation
# so that all requests will ignore SSL certificate errors, should you need that
# i.e. j = Jenkins('https://my.secure.jenkins', verify_ssl_cert=False)

# trigger a manual build and check results
j.build('warehouse-screens-us544_login')
j.last_result('warehouse-screens-us544_login')

# get only the result string (one of 'SUCCESS', 'UNSTABLE', 'FAILURE'):
j.last_result('warehouse-screens-us544_login')['result']
```

```
# get the configuration file for a job:
j.get_config_xml('template')

# Create a new job from a job named 'template', replacing variables
j.create_copy('my-new-job', 'template',
             repo='mbf-warehouse-screens',
             branch='us544_login',
             package='warehouse_screens')

# build
j.build('my-new-job')

# check result and delete if successful:
result = j.last_result('my-new-job')['result']
if result == 'SUCCESS':
    j.delete('my-new-job')
```

AutoJenkins from the Command Line

Available commands:

- `ajk-list` - list all jobs in a server
- `ajk-create` - create a job
- `ajk-build` - start building a job
- `ajk-delete` - delete a job

`ajk-list`

List all jobs in a Jenkins server. Each line in the output represents a job, and is colored according to the job's last build state:

- Blue: success
- Yellow: unstable
- Red: failure
- Gray: not built

A `*` symbol next to a job name indicates that the job is being built right now.

If instead of colored output, you prefer a string stating the status of the build, use the `--no-color` option. This is useful if you e.g. want to pipe the output into a `grep` command that filters jobs depending on status.

```
$ ajk-list -h

Usage: ajk-list host

Run autojenkins to list all jobs.

Options:
  -h, --help          show this help message and exit
  -u USER, --user=USER
                      username
```

```
-p PASSWORD, --password=PASSWORD
    password or token
-n, --no-color    do not use colored output
-r, --raw        only output job names, so that output is more usable for
                piping to other processes
```

ajk-create

Create a job from a template job, replacing variables that use the django/jinja2 syntax `{{ variable }}`.

Usage help:

```
$ ajk-create -h

Usage: ajk-create host [jobname] [options]

Run autojenkins to create a job.

Options:
  -h, --help            show this help message and exit
  -u USER, --user=USER  username
  -p PASSWORD, --password=PASSWORD
                        password or token
  -D VAR=VALUE          substitution variables to be used in the template
  -t TEMPLATE, --template=TEMPLATE
                        the template job to copy from
  -b, --build           start a build right after creation
```

Sample command:

```
$ ajk-create http://my.server my-job -t template -Dbranch=my-branch
```

ajk-build

Trigger the building of a job, optionally waiting for it to complete.

When used with the `--wait` flag, it will exit with a status different than 0 if the build is not successful. This makes it useful when including this in shell scripts that depend on a successful build to take further action.

Usage help:

```
$ ajk-build -h

Usage: ajk-build host [jobname] [options]

Run autojenkins to build a job.

Options:
  -h, --help            show this help message and exit
  -u USER, --user=USER  username
  -p PASSWORD, --password=PASSWORD
                        password or token
  -w, --wait           wait until the build completes
```

ajk-delete

Delete a job from a Jenkins server.

Usage help:

```
$ ajk-delete -h

Usage: ajk-delete host [jobname]... [options]

Run autojenkins to delete one or more jobs.

Options:
  -h, --help            show this help message and exit
  -u USER, --user=USER  username
  -p PASSWORD, --password=PASSWORD
                        password or token
```

More Info

Sources can be found in Github at <https://github.com/txels/autojenkins>

class autojenkins.jobs.**Jenkins** (*base_url, auth=None, verify_ssl_cert=True, proxies={}*)

Main class to interact with a Jenkins server.

all_jobs (*include_colorless=False*)

Get a list of tuples with (name, color) of all jobs in the server.

Color is blue, yellow or red depending on build results (SUCCESS, UNSTABLE or FAILED).

build (*jobname, params=None, wait=False, grace=10*)

Trigger Jenkins to build a job.

Parameters

- **params** – If params are provided, use the “buildWithParameters” endpoint
- **wait** – If True, wait until job completes building before returning

build_console (*jobname, build_number=None*)

Get the console output for the build of a job.

If no build number is specified, defaults to the most recent build.

build_info (*jobname, build_number=None*)

Get information for a build of a job.

If no build number is specified, defaults to the most recent build.

console_text (*jobname, build_number='lastBuild'*)

Get console text output of last build.

copy (*jobname, copy_from='template'*)

Copy a job from another one (by default from one called template).

create (*jobname, config_file, **context*)

Create a job from a configuration file.

create_copy (*jobname, template_job, enable=True, _force=False, **context*)

Create a job from a template job.

delete (*jobname*)
Delete a job.

disable (*jobname*)
Trigger Jenkins to disable a job.

enable (*jobname*)
Trigger Jenkins to enable a job.

get_config_xml (*jobname*)
Get the `config.xml` file that contains the job definition.

is_building (*jobname*)
Check if a job is building

job_info (*jobname*)
Get all information for a job as a Python object (dicts & lists).

job_url (*jobname*)
Get the human-browseable URL for a job.

last_build_console (*jobname*)
Get the console output for the last build of a job.

last_build_info (*jobname*)
Get information for last build of a job.

last_build_report (*jobname*)
Get full report of last build.

last_result (*jobname*)
Obtain results from last execution.

last_success (*jobname*)
Return information about the last successful build.

set_config_xml (*jobname*, *config*)
Replace the `config.xml` of an existing job.

transfer (*jobname*, *to_server*)
Copy a job to another server.

wait_for_build (*jobname*, *poll_interval=3*)
Wait until job has finished building

Sources can be found at <https://github.com/txels/autojenkins>

CHAPTER 3

Indices and tables

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

a

`autojenkins.jobs`, 7

A

all_jobs() (autojenkins.jobs.Jenkins method), 7
autojenkins.jobs (module), 7

B

build() (autojenkins.jobs.Jenkins method), 7
build_console() (autojenkins.jobs.Jenkins method), 7
build_info() (autojenkins.jobs.Jenkins method), 7

C

console_text() (autojenkins.jobs.Jenkins method), 7
copy() (autojenkins.jobs.Jenkins method), 7
create() (autojenkins.jobs.Jenkins method), 7
create_copy() (autojenkins.jobs.Jenkins method), 7

D

delete() (autojenkins.jobs.Jenkins method), 7
disable() (autojenkins.jobs.Jenkins method), 8

E

enable() (autojenkins.jobs.Jenkins method), 8

G

get_config_xml() (autojenkins.jobs.Jenkins method), 8

I

is_building() (autojenkins.jobs.Jenkins method), 8

J

Jenkins (class in autojenkins.jobs), 7
job_info() (autojenkins.jobs.Jenkins method), 8
job_url() (autojenkins.jobs.Jenkins method), 8

L

last_build_console() (autojenkins.jobs.Jenkins method), 8
last_build_info() (autojenkins.jobs.Jenkins method), 8
last_build_report() (autojenkins.jobs.Jenkins method), 8
last_result() (autojenkins.jobs.Jenkins method), 8

last_success() (autojenkins.jobs.Jenkins method), 8

S

set_config_xml() (autojenkins.jobs.Jenkins method), 8

T

transfer() (autojenkins.jobs.Jenkins method), 8

W

wait_for_build() (autojenkins.jobs.Jenkins method), 8