

---

# **AutoJenkins Documentation**

*Release 1.1.0*

**Carles Barrobés**

June 14, 2016



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



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.

### 1.1 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')
```

## 1.2 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

### 1.2.1 `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
```



## 1.2.2 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
```

## 1.2.3 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
```

## 1.2.4 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
```

### 1.3 More Info

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

---

**autojenkins.jobs**

---

**class** autojenkins.jobs.**Jenkins** (*base\_url, auth=None, verify\_ssl\_cert=True, proxies={}*)

Main class to interact with a Jenkins server.

**all\_jobs** ()

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>

---

## 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), 7

**E**

enable() (autojenkins.jobs.Jenkins method), 7

**G**

get\_config\_xml() (autojenkins.jobs.Jenkins method), 7

**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