unix-at Documentation

Release 0.2

Remi Rampin

Apr 27, 2023

Contents

Welcome to *unix-at*'s documentation!

This tiny Python library allows you to talk to the *at(1)* system, available on most UNIX machines, to schedule jobs to be run later.

Using *at(1)* can be much more light-weight than running a full-fledged job-processing system such as Celery if you are running very few jobs, however the performance will be much lower if you are running a considerate amount of tasks.

CHAPTER 1

Example

This simple example should be enough to get you started:

```
import unix_at

job = unix_at.submit_shell_job(['touch', '/some/file'])
unix_at.cancel_job(job)
job = unix_at.submit_python_job(os.mkdir, 'now + 1 hour', '/some/dir')
```

CHAPTER 2

Scheduling jobs

unix_at.**submit_shell_job**(*command*, *time*, *at='at'*)
Submits a shell command to be run later with *at(1)*.

Parameters

- **command** – A command as a single string or an iterable of words, similar to what is expected by the first argument to `subprocess.Popen`.
- **time** – Either a `datetime.datetime` object, or a string in the format accepted by *at(1)*, such as `"now + 1 minute"` or `"2m + 2 days"`.
- **at** – Overrides the location of the *at* binary (defaults to `'at'`).

Returns The *Job* object for the new job.

Note that *at(1)* usually restores the working directory and environment variables when it runs the job.

unix_at.**submit_python_job**(*func*, *time*, **args*, ***kwargs*)
Submits a Python function to be run later with *at(1)*.

The current interpreter will be used, unless *python* is set to a different executable.

Parameters

- **func** – Either a fully-qualified function name (e.g. `os.path.dirname`) or a function object (that will be pickled).
- **time** – Either a `datetime.datetime` object, or a string in the format accepted by *at(1)*, such as `"now + 1 minute"` or `"2m + 2 days"`.
- **at** – Overrides the location of the *at* binary (defaults to `'at'`).

Returns The *Job* object for the new job.

CHAPTER 3

Listing and manipulating jobs

unix_at.**list_jobs**(*at='at'*)
Lists all the jobs currently in the queue.

Parameters **at** – Overrides the location of the *at* binary (defaults to `'at'`).

Returns A *list* of *Job* objects.

unix_at.**get_script_for_job**(*job_name*, *at='at'*)
Gets the full shell script associated with a job.

Parameters

- **job_name** – Either the name of the job as a string, like it is shown by `at -l`, or a *Job* object.
- **at** – Overrides the location of the *at* binary (defaults to `'at'`).

Returns The script as *bytes*, or *None* if the job does not exist.

unix_at.**cancel_job**(*job_name*, *atrm=None*, *at=None*)
Cancels one or multiple jobs from their names or *Job* objects.

Parameters

- **job_name** – An iterable of either names of jobs as strings, like it is shown by `at -l`, or *Job* objects.
- **atrm** – Overrides the location of the *atrm* binary (defaults to `'atrm'`).
- **at** – Overrides the location of the *at* binary. If set, `at -r` will be called instead of `atrm`.

Returns *True* on success, *False* if some jobs were not found.

CHAPTER 4

Job objects

class unix_at.**Job**(*name*, *time*)

Represents a job, parsed from the output of *at(1)*.

name

Name of the job, shown by `at -l`

time

Time at which the job is to run, as a `datetime.datetime` object.

CHAPTER 5

Links

- GitLab repository
- Bug Tracker

Index