
job_progress Documentation

Release 0.0.1

Charles-Axel Dein

February 02, 2015

1	Quickstart	3
1.1	Installation	3
1.2	Setup	3
1.3	Storing progress	3
1.4	Getting progress	4
1.5	Setting state	4
1.6	Example session	4
1.7	Example Integration With Celery	5
2	Foreword	7
3	Indices and tables	9

Contents:

1.1 Installation

1.2 Setup

Here's how to setup *job_progress*:

```
# mypackage/jobprogress.py
from __future__ import absolute_import

from job_progress import JobProgress, Session
from job_progress.backends.redis import RedisBackend

TEST_CONFIG = {
    'backend_url': 'redis://localhost:6379/0',
    'heartbeat_enabled': True,
}
session = Session(backend=RedisBackend(TEST_CONFIG))
JobProgress.set_session(session)
```

1.3 Storing progress

You can then use it from other files:

```
from job_progress import states

from mypackage.job_progress import JobProgress

def toast_bread(toasts):
    """Toast some bread."""
    # You can store any meta data on the job, and provide a job identifier.
    job = JobProgress(id_='toaster_job_1'
                      data={'created_at': '2014-12-12'},
                      amount=len(toasts), # Amount of work
                      state=states.STARTED)

    for toast in toasts:
        try:
```

```
        # Toast bread, then
        job.add_one_success()
    except:
        job.add_failure()
```

1.4 Getting progress

If you have a specific job id, you can get the job directly:

```
from mypackage.job_progress import session

job = session.get('toaster_job_1')
job.get_progress()
# Return something like:
# {'FAILURE': 1, 'PENDING': 8, 'SUCCESS': 1}
```

Jobs are also indexed:

```
from job_progress import states

from mypackage.job_progress import session

jobs = session.query(state=states.STARTED)
```

1.5 Setting state

You can set the state of a specific job:

```
from job_progress import states, session

job = session.get('31bd07a2-5174-4cbd-a575-d68382518f20')
job.state = states.SUCCESS
```

1.6 Example session

```
>>> from job_progress.job_progress import JobProgress
>>> from job_progress.backends.redis import RedisBackend
>>> from job_progress import states, session
>>> settings = {"url": "redis://localhost:6379/0"}
>>> JobProgress.backend = RedisBackend(settings)
>>> JobProgress.session = session.Session()
>>> # 10 is the amount of work
>>> job = JobProgress({"created_at": "2013-12-12"}, amount=10, state=states.STARTED)
>>> job.add_one_success()
>>> job.add_one_failure()
>>> job.get_progress()
{'FAILURE': 1, 'PENDING': 8, 'SUCCESS': 1}
>>> del job
>>> # Jobs are indexed
>>> jobs = JobProgress.query(state=states.STARTED)
>>> jobs[0].get_progress()
{'FAILURE': 1, 'PENDING': 8, 'SUCCESS': 1}
```


1.7 Example Integration With Celery

Foreword

The main idea behind this library is to be able to store a job's metadata and progress into a backend, so as to be able to retrieve them.

Indices and tables

- *genindex*
- *modindex*
- *search*