

Basic python job submission I. Single command

To begin we will go through a simple job submission. In this example, we will submit a job that runs a single command (in this case, the command "hostname") on one worker. This is the simplest form of job submission.

```
#!/usr/bin/env python3
# Below are required imports for the script to run
import os, sys
# The next few lines attempt to import the Qube API. If the path to the qb module
# is not in $PATH or $PYTHONPATH, we will attempt to find it by looking in known
# locations
try:
    import qb
except ImportError:
    if os.environ.get("QBDIR"):
        qbdir_api = os.path.join(os.environ.get("QBDIR"), "api", "python")
        for api_path in (qbdir_api,
                         "/Applications/pfx/qube/api/python/",
                         "/usr/local/pfx/qube/api/python/",
                         "C:\\Program Files\\pfx\\qube\\api\\python",
                         "C:\\Program Files (x86)\\pfx\\qube\\api\\python"):
            if api_path not in sys.path and os.path.exists(api_path):
                sys.path.insert(0, api_path)
                try:
                    import qb
                except:
                    continue
                break
    # this should throw an exception if we've exhausted all other possibilities
    import qb

# Below is the main function to run in this script
def main():

    # Below creates an empty dictionary to be filled by the following lines of code
    job = {}

    # Below defines the name of the Qube! job. This is the name that will be
    # displayed in the GUI and through the command line tools
    job['name'] = 'python test job'

    # Below defines how many Instances/subjobs the job is to spawn. Because we
    # will be running only a single command, there is no need to request more than 1.

    job['cpus'] = 1

    # Below defines the internal Qube! jobtype to be used to execute the job.
    # 'cmdline' tells Qube that on the backend, we will execute a single command line
    # command. This will be the same as opening a terminal/command prompt and typing
    # out a command.
    job['prototype'] = 'cmdline'

    # The below parameters are explained further in the "Job submission with job
    # package explained" page
    package = {}
    package['cmdline'] = 'hostname'
    job['package'] = package
```

```
# Below creates an empty list filled by the following lines of code.
listOfJobsToSubmit = []

# Below evaluates the jobs to be submitted and adds the to the above list
listOfJobsToSubmit.append(job)

# Below calls the list of jobs to be submitted and then prints the job IDs for
each
# While it is not strictly necessary that one submits a list of jobs, it is a good
# habit to start, so we will only submit lists of jobs. It is, however, perfectly
# acceptable to qb.submit(job)
listOfSubmittedJobs = qb.submit(listOfJobsToSubmit)
for job in listOfSubmittedJobs:
    print(job['id'])
# Below runs the "main" function
if __name__ == "__main__":
    main()
```

```
sys.exit(0)
```

This example and others like it can be found in:

- Windows - C:\Program Files\pfx\qube\examples
- macOS - /Application/pfx/qube/examples
- Linux - /usr/local/pfx/qube/examples

Continue to [Basic python job submission II. Frames](#)