

```

1 #!/usr/bin/env python2.6
2 '''
3 This is a very simple program to illustrate how to use the qjam framework. It
4 shows how to:
5
6 * create a pool of workers
7 * create a simple DataSet object out of a numpy matrix
8 * start a qjam master
9 * run a job on the worker pool and retrieve the result
10
11 All of the significant pieces are sufficiently commented to help you understand
12 how to write programs to use qjam.
13 '''
14 import logging
15 import os
16 import sys
17
18 import numpy
19
20 # Set up import path to qjam.
21 sys.path.append(os.path.join(os.path.dirname(sys.argv[0]), '..'))
22
23 from qjam.dataset import NumpyMatrixDataSet
24 from qjam.master.master import Master
25 from qjam.master.remote_worker import RemoteWorker
26
27 # sum_dataset is an example module that contains a function that is run on each
28 # piece of the dataset.
29 from examples import sum_dataset
30
31
32 def main():
33     _fmt = '%(asctime)s [%(levelname)s] %(name)s: %(message)s'
34     logging.basicConfig(level=logging.INFO, format=_fmt)
35
36     # Worker hostnames are the commandline options. Each hostname specified on
37     # the commandline will be bootstrapped and have a worker started on it.
38     if len(sys.argv) < 2:
39         print 'usage: %s <hostnames> [<hostname> ...]' % sys.argv[0]
40         sys.exit(1)
41     hostnames = sys.argv[1:]
42
43     # Start the worker processes on the specified hostnames.
44     workers = []
45     for i, hostname in enumerate(hostnames):
46         worker = RemoteWorker(hostname)
47         workers.append(worker)
48         print 'Started worker %d on %s' % (i, hostname)
49
50     # Create a numpy matrix.
51     matrix = numpy.matrix('1 2 3 4; 5 6 7 8; 9 10 11 12')
52
53     print 'Created matrix:'
54     print matrix
55     print
56
57     # Create a DataSet object from this matrix.
58     #
59     # DataSet objects are necessary so the framework can determine how to split
60     # up the data into smaller chunks and distribute the chunks to the workers.
61     dataset = NumpyMatrixDataSet(matrix)
62
63     # Sum the matrix!
64     master = Master(workers)
65     params = None # No parameters needed for this job.
66     result = master.run(sum_dataset, params, dataset)
67
68     # Print the result.
69     print 'Result is: %d' % result
70
71
72 if __name__ == '__main__':
73     main()

```