

Batch API Comparison

Stefano Russo
Gareth Hughes

Sources:

DIRAC readthedocs:

<https://dirac.readthedocs.io/en/latest/CodeDocumentation/Interfaces/API/Dirac.html>

Rosetta draft paper: http://sarusso.github.io/Rosetta_draft.pdf (See section 5.2. Computing)

Batch API Comparison

Sources:

DIRAC:

[deleteJob\(jobID\)](#)
[getJobAttributes\(jobID, printOutput=False\)](#)
[getJobInputData\(jobID\)](#)
[getJobParameters\(jobID, printOutput=False\)](#)
[getJobStatus\(jobID\)](#)
[getJobSummary\(jobID, outputFile=None, printOutput=False\)](#)
[pingService\(system, service, printOutput=False, url=None\)](#)
[rescheduleJob\(jobID\)](#)
[submitJob\(job, mode='wms'\)](#)
[setCPUTime\(timeInSecs\)](#)
[setExecutable\(executable, arguments="", logFile="", modulesList=None, parameters=None, paramValues=None\)](#)
[setExecutionEnv\(environmentDict\)](#)
[setInputData\(lfns\)](#)
[setName\(jobName\)](#)
[setNumberOfProcessors\(numberOfProcessors=None, minNumberOfProcessors=None, maxNumberOfProcessors=None\)](#)
[setOutputData\(lfns, outputSE=None, outputPath=""\)](#)
[addToWorkflow \(bool\)](#)
[setPlatform\(platform\)](#)

Rosetta:

DIRAC

```
#####  
from DIRAC.Interfaces.API.Dirac import Dirac  
#####
```

Everything is based on the jobID which can be an int, string or list. The jobID is set when the job is submitted.

jobID = {int, string, list}

submitJob(job, mode='wms')

Submit jobs to DIRAC (by default to the Workload Management System). These can be either:

{'OK': True, 'Value': '12345'}

Returns OK or ERROR

deleteJob(jobID)

Delete (set status=DELETED) to job or list of jobs from the WMS. If running, these jobs will be first killed

{'OK': True, 'Value': [12345]}

Returns OK or ERROR

getJobAttributes(jobID, printOutput=False)

Return DIRAC attributes associated with the given job. Each job will have certain attributes that affect the journey through the workload management system

{'AccountedFlag': 'False', 'ApplicationNumStatus': '0',

'ApplicationStatus': 'Job Finished Successfully',

'CPUTime': '0.0', 'DIRACSetup': 'LHCb-Production'}

Returns OK or ERROR

getJobInputData(jobID)

Retrieve the input data requirement of any job existing in the workload management system.

{'OK': True, 'Value': {1405:

['LFN:/lhcb/production/DC06/phys-v2-lumi5/00001680/DST/0000/00001680_00000490_5.dst']}}

Returns OK or ERROR

getJobParameters(jobID, printOutput=False)

Return DIRAC parameters associated with the given job. DIRAC keeps track of several job

parameters which are kept in the job monitoring service, see example below. Selected parameters also printed to screen.

```
{'OK': True, 'Value': {'JobPath':  
'JobPath,JobSanity,JobPolicy,InputData,JobScheduling,TaskQueue',  
'JobSanityCheck': 'Job: 768 JDL: OK, InputData: 2 LFNs OK, '  
Returns OK or ERROR
```

getJobStatus(jobID)

Monitor the status of DIRAC Jobs.

```
{79241: {'Status': 'Done',  
        'MinorStatus': 'Execution Complete',  
        'ApplicationStatus': 'some app status'  
        'Site': 'LCG.CERN.ch'}}
```

Returns OK or ERROR

getJobSummary(jobID, outputFile=None, printOutput=False)

Output similar to the web page can be printed to the screen or stored as a file or just returned as a dictionary for further usage.

```
{'OK': True, 'Value': {959209: {'Status': 'Staging', 'LastUpdateTime': '2008-12-08 16:43:18',  
'MinorStatus': '28 / 30', 'Site': 'Unknown', 'HeartBeatTime': 'None', 'ApplicationStatus': 'unknown',  
'JobGroup': '00003403', 'Owner': 'joel', 'SubmissionTime': '2008-12-08 16:41:38'}}}
```

Returns OK or ERROR

pingService(system, service, printOutput=False, url=None)

The ping function will attempt to return standard information from a system service if this is available. If the ping() command is unsuccessful it could indicate a period of service unavailability.

```
{'OK': True, 'Value': 'Job ping result'}
```

Returns OK or ERROR

rescheduleJob(jobID)

Reschedule a job or list of jobs in the WMS. This operation is the same as resubmitting the same job as new. The rescheduling operation may be performed to a configurable maximum number of times but the owner of a job can also reset this counter and reschedule jobs again by hand.

```
{'OK': True, 'Value': [12345]}
```

Returns OK or ERROR

```
#####  
from DIRAC.Interfaces.API.Job import Job  
#####
```

setCPUTime(timeInSecs)

setExecutable(executable, arguments="", logFile="", modulesList=None,
parameters=None, paramValues=None)

setExecutionEnv(environmentDict)

Optionally specify a dictionary of key, value pairs to be set before the job executes e.g.
{'MYVAR':3}

setInputData(lfns)

Specify input data by Logical File Name (LFN).

setName(jobName)

A name for the job can be specified if desired. This will appear in the JobName field of the monitoring webpage. If nothing is specified a default value will appear.

setNumberOfProcessors(numberOfProcessors=None, minNumberOfProcessors=None,
maxNumberOfProcessors=None)

Returns OK or ERROR

setOutputData(lfns, outputSE=None, outputPath="")

For specifying output data to be registered in Grid storage. If a list of OutputSEs are specified the job wrapper will try each in turn until successful. If the OutputPath is specified this will appear only after / <VO> / user / <initial> / <username> directory.

The output data can be LFNs or local file names. If they are LFNs they should be pre-pended by "LFN:", otherwise they will be interpreted as local files to be found. If local files are specified, then specifying the outputPath may become necessary, because if it's not specified then it will be constructed starting from the user name.

`addToWorkflow (bool)`

flag to add parameter to the workflow on the fly, if str, then use as the workflow parameter

Returns OK or ERROR

`setPlatform(platform)`

Developer function: sets the target platform, e.g. `Linux_x86_64_glibc-2.17`. This platform is in the form of what it is returned by the `dirac-platform` script (or `dirac-architecture` if your extension provides it)

```
#####
```

```
Basic Example Script
```

```
#####
```

```
# setup DIRAC
```

```
from DIRAC.Core.Utilities.DIRACScript import DIRACScript as Script
Script.parseCommandLine(ignoreErrors=False)
```

```
from DIRAC.Interfaces.API.Job import Job
from DIRAC.Interfaces.API.Dirac import Dirac
```

```
dirac = Dirac()
j = Job()
```

```
j.setCPUTime(500)
j.setExecutable('/bin/echo hello')
j.setExecutable('/bin/hostname')
j.setExecutable('/bin/echo hello again')
j.setName('API')
```

```
jobID = dirac.submitJob(j)
print('Submission Result: ', jobID)
```

```
status = getJobStatus(jobID)
```

```
print('Status: ', status)
```

```
#####
```

Rosetta: