yarn-api-client Documentation

Release 0.2.3

Iskandarov Eduard

Sep 01, 2017

Contents

1 ResourceManager API’s. 3

2 NodeManager API’s. 5

3 MapReduce Application Master API’s. 7

4 History Server API’s. 9

5 Indices and tables 11

Python Module Index 13

Contents:

CHAPTER 1

ResourceManager API's.

class `yarn_api_client.resource_manager.`**`ResourceManager`**(*address=None*, *port=8088*, *timeout=30*)

The ResourceManager REST API's allow the user to get information about the cluster - status on the cluster, metrics on the cluster, scheduler information, information about nodes in the cluster, and information about applications on the cluster.

`cluster_application`(*application_id*)

An application resource contains information about a particular application that was submitted to a cluster.

`cluster_application_attempts`(*application_id*)

With the application attempts API, you can obtain a collection of resources that represent an application attempt.

`cluster_application_statistics`(*state_list=None*, *application_type_list=None*)

With the Application Statistics API, you can obtain a collection of triples, each of which contains the application type, the application state and the number of applications of this type and this state in ResourceManager context.

This method work in Hadoop > 2.0.0

`cluster_applications`(*state=None*, *final_status=None*, *user=None*, *queue=None*, *limit=None*, *started_time_begin=None*, *started_time_end=None*, *finished_time_begin=None*, *finished_time_end=None*)

With the Applications API, you can obtain a collection of resources, each of which represents an application.

`cluster_information`()

The cluster information resource provides overall information about the cluster.

`cluster_metrics`()

The cluster metrics resource provides some overall metrics about the cluster. More detailed metrics should be retrieved from the jmx interface.

`cluster_node`(*node_id*)

A node resource contains information about a node in the cluster.

cluster_nodes(*state=None*, *healthy=None*)

With the Nodes API, you can obtain a collection of resources, each of which represents a node.

cluster_scheduler()

A scheduler resource contains information about the current scheduler configured in a cluster. It currently supports both the Fifo and Capacity Scheduler. You will get different information depending on which scheduler is configured so be sure to look at the type information.

CHAPTER 2

NodeManager API's.

class `yarn_api_client.node_manager.`**`NodeManager`**(*address=None*, *port=8042*, *timeout=30*)

The NodeManager REST API's allow the user to get status on the node and information about applications and containers running on that node.

`node_application`(*application_id*)

An application resource contains information about a particular application that was run or is running on this NodeManager.

`node_applications`(*state=None*, *user=None*)

With the Applications API, you can obtain a collection of resources, each of which represents an application.

`node_container`(*container_id*)

A container resource contains information about a particular container that is running on this NodeManager.

`node_containers`()

With the containers API, you can obtain a collection of resources, each of which represents a container.

`node_information`()

The node information resource provides overall information about that particular node.

CHAPTER 3

MapReduce Application Master API's.

class `yarn_api_client.application_master.`**`ApplicationMaster`**(*address=None*, *port=8088*, *timeout=30*)

The MapReduce Application Master REST API's allow the user to get status on the running MapReduce application master. Currently this is the equivalent to a running MapReduce job. The information includes the jobs the app master is running and all the job particulars like tasks, counters, configuration, attempts, etc.

`application_information`(*application_id*)

The MapReduce application master information resource provides overall information about that mapreduce application master. This includes application id, time it was started, user, name, etc.

`job`(*application_id*, *job_id*)

A job resource contains information about a particular job that was started by this application master. Certain fields are only accessible if user has permissions - depends on acl settings.

`job_attempts`(*job_id*)

With the job attempts API, you can obtain a collection of resources that represent the job attempts.

`job_conf`(*application_id*, *job_id*)

A job configuration resource contains information about the job configuration for this job.

`job_counters`(*application_id*, *job_id*)

With the job counters API, you can object a collection of resources that represent all the counters for that job.

`job_task`(*application_id*, *job_id*, *task_id*)

A Task resource contains information about a particular task within a job.

`job_tasks`(*application_id*, *job_id*)

With the tasks API, you can obtain a collection of resources that represent all the tasks for a job.

`jobs`(*application_id*)

The jobs resource provides a list of the jobs running on this application master.

`task_attempt`(*application_id*, *job_id*, *task_id*, *attempt_id*)

A Task Attempt resource contains information about a particular task attempt within a job.

`task_attempt_counters`(*application_id*, *job_id*, *task_id*, *attempt_id*)
With the task attempt counters API, you can object a collection of resources that represent al the counters for that task attempt.

`task_attempts`(*application_id*, *job_id*, *task_id*)
With the task attempts API, you can obtain a collection of resources that represent a task attempt within a job.

`task_counters`(*application_id*, *job_id*, *task_id*)
With the task counters API, you can object a collection of resources that represent all the counters for that task.

CHAPTER 4

History Server API's.

class `yarn_api_client.history_server.`**`HistoryServer`**(*address=None*, *port=19888*, *timeout=30*)

The history server REST API's allow the user to get status on finished applications. Currently it only supports MapReduce and provides information on finished jobs.

`application_information`()

The history server information resource provides overall information about the history server.

`job`(*job_id*)

A Job resource contains information about a particular job identified by jobid.

`job_attempts`(*job_id*)

With the job attempts API, you can obtain a collection of resources that represent a job attempt.

`job_conf`(*job_id*)

A job configuration resource contains information about the job configuration for this job.

`job_counters`(*job_id*)

With the job counters API, you can object a collection of resources that represent al the counters for that job.

`job_task`(*job_id*, *task_id*)

A Task resource contains information about a particular task within a job.

`job_tasks`(*job_id*, *type=None*)

With the tasks API, you can obtain a collection of resources that represent a task within a job.

`jobs`(*state=None*, *user=None*, *queue=None*, *limit=None*, *started_time_begin=None*, *started_time_end=None*, *finished_time_begin=None*, *finished_time_end=None*)

The jobs resource provides a list of the MapReduce jobs that have finished. It does not currently return a full list of parameters.

`task_attempt`(*job_id*, *task_id*, *attempt_id*)

A Task Attempt resource contains information about a particular task attempt within a job.

`task_attempt_counters`(*job_id*, *task_id*, *attempt_id*)

With the task attempt counters API, you can object a collection of resources that represent al the counters for that task attempt.

task_attempts (*job_id*, *task_id*)

With the task attempts API, you can obtain a collection of resources that represent a task attempt within a job.

task_counters (*job_id*, *task_id*)

With the task counters API, you can object a collection of resources that represent all the counters for that task.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

y

yarn_api_client.application_master, 7
yarn_api_client.history_server, 9
yarn_api_client.node_manager, 5
yarn_api_client.resource_manager, 3

Index

A

application_information() (yarn_api_client.application_master.ApplicationMaster method), 7

application_information() (yarn_api_client.history_server.HistoryServer method), 9

ApplicationMaster (class in yarn_api_client.application_master), 7

C

cluster_application() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_application_attempts() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_application_statistics() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_applications() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_information() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_metrics() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_node() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_nodes() (yarn_api_client.resource_manager.ResourceManager method), 3

cluster_scheduler() (yarn_api_client.resource_manager.ResourceManager method), 4

H

HistoryServer (class in yarn_api_client.history_server), 9

J

job() (yarn_api_client.application_master.ApplicationMaster method), 7

job() (yarn_api_client.history_server.HistoryServer method), 9

job_attempts() (yarn_api_client.application_master.ApplicationMaster method), 7

job_attempts() (yarn_api_client.history_server.HistoryServer method), 9

job_conf() (yarn_api_client.application_master.ApplicationMaster method), 7

job_conf() (yarn_api_client.history_server.HistoryServer method), 9

job_counters() (yarn_api_client.application_master.ApplicationMaster method), 7

job_counters() (yarn_api_client.history_server.HistoryServer method), 9

job_task() (yarn_api_client.application_master.ApplicationMaster method), 7

job_task() (yarn_api_client.history_server.HistoryServer method), 9

job_tasks() (yarn_api_client.application_master.ApplicationMaster method), 7

job_tasks() (yarn_api_client.history_server.HistoryServer method), 9

jobs() (yarn_api_client.application_master.ApplicationMaster method), 7

jobs() (yarn_api_client.history_server.HistoryServer method), 9

N

node_application() (yarn_api_client.node_manager.NodeManager method), 5

node_applications() (yarn_api_client.node_manager.NodeManager method), 5

node_container() (yarn_api_client.node_manager.NodeManager method), 5

node_containers() (yarn_api_client.node_manager.NodeManager method), 5

node_information() (yarn_api_client.node_manager.NodeManager method), 5

NodeManager (class in yarn_api_client.node_manager), 5

R

ResourceManager (class in yarn_api_client.resource_manager), 3

T

task_attempt() (yarn_api_client.application_master.ApplicationMaster method), 7
task_attempt() (yarn_api_client.history_server.HistoryServer method), 9
task_attempt_counters() (yarn_api_client.application_master.ApplicationMaster method), 7
task_attempt_counters() (yarn_api_client.history_server.HistoryServer method), 9
task_attempts() (yarn_api_client.application_master.ApplicationMaster method), 8
task_attempts() (yarn_api_client.history_server.HistoryServer method), 10
task_counters() (yarn_api_client.application_master.ApplicationMaster method), 8
task_counters() (yarn_api_client.history_server.HistoryServer method), 10

Y

yarn_api_client.application_master (module), 7
yarn_api_client.history_server (module), 9
yarn_api_client.node_manager (module), 5
yarn_api_client.resource_manager (module), 3