yarn-api-client Documentation

Release 0.2.4

Iskandarov Eduard

Contents

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.

If *address* argument is *None* client will try to extract *address* and *port* from Hadoop configuration files.

> **Parameters**
>
> - **`address`** (`str`) – ResourceManager HTTP address
>
> - **`port`** (`int`) – ResourceManager HTTP port
>
> - **`timeout`** (`int`) – API connection timeout in seconds

`cluster_application`(*application_id*)

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

> **Parameters** **`application_id`** (`str`) – The application id
>
> **Returns** API response object with JSON data
>
> **Return type** `yarn_api_client.base.Response`

`cluster_application_attempts`(*application_id*)

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

> **Parameters** **`application_id`** (`str`) – The application id
>
> **Returns** API response object with JSON data
>
> **Return type** `yarn_api_client.base.Response`

`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

> **Parameters**
>
> - **state_list** (`list`) – states of the applications, specified as a comma-separated list. If states is not provided, the API will enumerate all application states and return the counts of them.
> - **application_type_list** (`list`) – types of the applications, specified as a comma-separated list. If applicationTypes is not provided, the API will count the applications of any application type. In this case, the response shows * to indicate any application type. Note that we only support at most one applicationType temporarily. Otherwise, users will expect an BadRequestException.
>
> **Returns** API response object with JSON data
>
> **Return type** yarn_api_client.base.Response

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.
>
> **Parameters**
>
> - **state** (`str`) – state of the application
> - **final_status** (`str`) – the final status of the application - reported by the application itself
> - **user** (`str`) – user name
> - **queue** (`str`) – queue name
> - **limit** (`str`) – total number of app objects to be returned
> - **started_time_begin** (`str`) – applications with start time beginning with this time, specified in ms since epoch
> - **started_time_end** (`str`) – applications with start time ending with this time, specified in ms since epoch
> - **finished_time_begin** (`str`) – applications with finish time beginning with this time, specified in ms since epoch
> - **finished_time_end** (`str`) – applications with finish time ending with this time, specified in ms since epoch
>
> **Returns** API response object with JSON data
>
> **Return type** yarn_api_client.base.Response
>
> **Raises** **yarn_api_client.errors.IllegalArgumentError** – if *state* or *final_status* incorrect

cluster_information()

> The cluster information resource provides overall information about the cluster.
>
> **Returns** API response object with JSON data
>
> **Return type** yarn_api_client.base.Response

cluster_metrics()

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

> > **Returns** API response object with JSON data
>
> > **Return type** yarn_api_client.base.Response

cluster_node(*node_id*)
> A node resource contains information about a node in the cluster.
>
> > **Parameters node_id**(*str*) – The node id
>
> > **Returns** API response object with JSON data
>
> > **Return type** yarn_api_client.base.Response

cluster_nodes(*state=None*, *healthy=None*)
> With the Nodes API, you can obtain a collection of resources, each of which represents a node.
>
> > **Returns** API response object with JSON data
>
> > **Return type** yarn_api_client.base.Response
>
> > **Raises yarn_api_client.errors.IllegalArgumentError** – if *healthy* incorrect

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.
>
> > **Returns** API response object with JSON data
>
> > **Return type** yarn_api_client.base.Response

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.

   > **Parameters**
   >
   > - **`address`** (`str`) – NodeManager HTTP address
   >
   > - **`port`** (`int`) – NodeManager HTTP port
   >
   > - **`timeout`** (`int`) – API connection timeout in seconds

   `node_application`(*application_id*)
      An application resource contains information about a particular application that was run or is running on this NodeManager.

      > **Parameters** **`application_id`** (`str`) – The application id
      >
      > **Returns** API response object with JSON data
      >
      > **Return type** `yarn_api_client.base.Response`

   `node_applications`(*state=None*, *user=None*)
      With the Applications API, you can obtain a collection of resources, each of which represents an application.

      > **Parameters**
      >
      > - **`state`** (`str`) – application state
      >
      > - **`user`** (`str`) – user name
      >
      > **Returns** API response object with JSON data
      >
      > **Return type** `yarn_api_client.base.Response`
      >
      > **Raises** **`yarn_api_client.errors.IllegalArgumentError`** – if *state* incorrect

node_container(*container_id*)
    A container resource contains information about a particular container that is running on this NodeManager.

> **Parameters** **container_id**(*str*) – The container id
>
> **Returns** API response object with JSON data
>
> **Return type** yarn_api_client.base.Response

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

> **Returns** API response object with JSON data
>
> **Return type** yarn_api_client.base.Response

node_information()
    The node information resource provides overall information about that particular node.

> **Returns** API response object with JSON data
>
> **Return type** yarn_api_client.base.Response

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.

If *address* argument is *None* client will try to extract *address* and *port* from Hadoop configuration files.

> **Parameters**
>> • **address** (*str*) – Proxy HTTP address
>>
>> • **port** (*int*) – Proxy HTTP port
>>
>> • **timeout** (*int*) – API connection timeout in seconds

`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.

> **Returns** API response object with JSON data
>
> **Return type** `yarn_api_client.base.Response`

`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.

> **Parameters**
>> • **application_id** (*str*) – The application id
>>
>> • **job_id** (*str*) – The job id
>
> **Returns** API response object with JSON data
>
> **Return type** `yarn_api_client.base.Response`

`job_attempts`(*job_id*)

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

Parameters **job_id** (*str*) – The job id

Returns API response object with JSON data

Return type `yarn_api_client.base.Response`

job_conf (*application_id*, *job_id*)
A job configuration resource contains information about the job configuration for this job.

Parameters

- **application_id** (*str*) – The application id
- **job_id** (*str*) – The job id

Returns API response object with JSON data

Return type `yarn_api_client.base.Response`

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.

Parameters

- **application_id** (*str*) – The application id
- **job_id** (*str*) – The job id

Returns API response object with JSON data

Return type `yarn_api_client.base.Response`

job_task (*application_id*, *job_id*, *task_id*)
A Task resource contains information about a particular task within a job.

Parameters

- **application_id** (*str*) – The application id
- **job_id** (*str*) – The job id
- **task_id** (*str*) – The task id

Returns API response object with JSON data

Return type `yarn_api_client.base.Response`

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.

Parameters

- **application_id** (*str*) – The application id
- **job_id** (*str*) – The job id

Returns API response object with JSON data

Return type `yarn_api_client.base.Response`

jobs (*application_id*)
The jobs resource provides a list of the jobs running on this application master.

Parameters **application_id** (*str*) – The application id

Returns API response object with JSON data

Return type `yarn_api_client.base.Response`

---

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

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

>   **Parameters**

>   - **application_id** (*str*) – The application id
>   - **job_id** (*str*) – The job id
>   - **task_id** (*str*) – The task id
>   - **attempt_id** (*str*) – The attempt id

>   **Returns** API response object with JSON data

>   **Return type** yarn_api_client.base.Response

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.

>   **Parameters**

>   - **application_id** (*str*) – The application id
>   - **job_id** (*str*) – The job id
>   - **task_id** (*str*) – The task id
>   - **attempt_id** (*str*) – The attempt id

>   **Returns** API response object with JSON data

>   **Return type** yarn_api_client.base.Response

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.

>   **Parameters**

>   - **application_id** (*str*) – The application id
>   - **job_id** (*str*) – The job id
>   - **task_id** (*str*) – The task id

>   **Returns** API response object with JSON data

>   **Return type** yarn_api_client.base.Response

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.

>   **Parameters**

>   - **application_id** (*str*) – The application id
>   - **job_id** (*str*) – The job id
>   - **task_id** (*str*) – The task id

>   **Returns** API response object with JSON data

>   **Return type** yarn_api_client.base.Response

History Server API's.

class yarn_api_client.history_server.**HistoryServer**(*address=None*, *port=19888*, *time-out=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.

If *address* argument is *None* client will try to extract *address* and *port* from Hadoop configuration files.

> **Parameters**
> - **address** (*str*) – HistoryServer HTTP address
> - **port** (*int*) – HistoryServer HTTP port
> - **timeout** (*int*) – API connection timeout in seconds

application_information()

> The history server information resource provides overall information about the history server.
>
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

job(*job_id*)

> A Job resource contains information about a particular job identified by jobid.
>
> > **Parameters** **job_id**(*str*) – The job id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

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.
>
> > **Parameters** **job_id**(*str*) – The job id
> >
> > **Returns** API response object with JSON data

> **Return type** yarn_api_client.base.Response

job_counters(*job_id*)

> With the job counters API, you can object a collection of resources that represent al the counters for that job.
>
> > **Parameters** **job_id** (*str*) – The job id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

job_task(*job_id*, *task_id*)

> A Task resource contains information about a particular task within a job.
>
> > **Parameters**
> >
> > - **job_id** (*str*) – The job id
> > - **task_id** (*str*) – The task id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

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

> With the tasks API, you can obtain a collection of resources that represent a task within a job.
>
> > **Parameters**
> >
> > - **job_id** (*str*) – The job id
> > - **type** (*str*) – type of task, valid values are m or r. m for map task or r for reduce task
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

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.
>
> > **Parameters**
> >
> > - **user** (*str*) – user name
> > - **state** (*str*) – the job state
> > - **queue** (*str*) – queue name
> > - **limit** (*str*) – total number of app objects to be returned
> > - **started_time_begin** (*str*) – jobs with start time beginning with this time, specified in ms since epoch
> > - **started_time_end** (*str*) – jobs with start time ending with this time, specified in ms since epoch
> > - **finished_time_begin** (*str*) – jobs with finish time beginning with this time, specified in ms since epoch
> > - **finished_time_end** (*str*) – jobs with finish time ending with this time, specified in ms since epoch
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

> **Raises** **yarn_api_client.errors.IllegalArgumentError** – if *state* incorrect

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

> A Task Attempt resource contains information about a particular task attempt within a job.
>
> > **Parameters**
> >
> > - **job_id** (*str*) – The job id
> > - **task_id** (*str*) – The task id
> > - **attempt_id** (*str*) – The attempt id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

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.
>
> > **Parameters**
> >
> > - **job_id** (*str*) – The job id
> > - **task_id** (*str*) – The task id
> > - **attempt_id** (*str*) – The attempt id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

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.
>
> > **Parameters**
> >
> > - **job_id** (*str*) – The job id
> > - **task_id** (*str*) – The task id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

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.
>
> > **Parameters**
> >
> > - **job_id** (*str*) – The job id
> > - **task_id** (*str*) – The task id
> >
> > **Returns** API response object with JSON data
> >
> > **Return type** yarn_api_client.base.Response

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

y

Index

R

T

Y