yarn-api-client Documentation

Release 0.2.4

Iskandarov Eduard

Nov 22, 2021

Contents

Contents:

CHAPTER 1

Base Response class

class yarn_api_client.base.**Response**(*response*)

Basic container for response dictionary

Parameters **response** (*requests.Response*) – Response for call via requests lib

CHAPTER 2

ResourceManager API's.

class yarn_api_client.resource_manager.**ResourceManager**(*service_endpoints=None*, *timeout=30*, *auth=None*, *verify=True*)

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 *service_endpoint* argument is *None* client will try to extract it from Hadoop configuration files. If both *address* and *alt_address* are provided, the address corresponding to the ACTIVE HA Resource Manager will be used.

Parameters

- **service_endpoints** (*List[str]*) – List of ResourceManager HTTP(S) addresses
- **timeout** (*int*) – API connection timeout in seconds
- **auth** (*AuthBase*) – Auth to use for requests configurations
- **verify** (*boolean*) – Either a boolean, in which case it controls whether we verify the server's TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`

application_activities(*application_id*, *max_time=None*, *request_priorities=None*, *allocation_request_ids=None*, *group_by=None*, *limit=None*, *actions=None*, *summarize=None*)

Application activities RESTful API is available if you are using capacity scheduler and can fetch useful scheduling info for a specified application, the response has a hierarchical layout with following fields:

- AppActivities - AppActivities are root element of application activities within basic

information. * Allocations - Allocations are allocation attempts at app level queried from the cache. * Requests - Requests are shown as children of allocation, each of them contains request name, request priority, allocation request id, allocation state and optional children. * Nodes - Nodes are shown as children of request, each of them contains node id, allocation state, optional name which should appear after allocating or reserving a container on the node, and optional diagnostic which should appear if failed to allocate or reserve a container on the node. For aggregated nodes grouped by allocation state and diagnostic, each of them contains allocation state, aggregated node IDs and optional diagnostic.

Parameters **maxTime** (*int*) – the max duration in seconds from now on for recording application

activities. If not specified, this will default to 3 (seconds). :param List[int] requestPriorities: the priorities of request, used to filter application activities :param List[int] allocationRequestIds: the allocation request IDs of request, used to filter application activities :param str groupBy: the aggregation type of application activities, currently only support "diagnostic" with which user can query aggregated activities grouped by allocation state and diagnostic. :param str limit: the limit of application activities which can reduce the cost for both server and client side. :param List[str] actions: the required actions of app activities including "refresh" and "get" :param boolean summarize: whether app activities in multiple scheduling processes need to be summarized, specified as boolean, it's useful when multi-node placement disabled, because only one node can be considered in a single scheduling process, enabling this can give us a summary with diagnostics on all nodes. :returns: API response object with JSON data :rtype: *yarn_api_client.base.Response*

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_attempt_container_info(*application_id*, *attempt_id*, *container_id*)

With the application attempts API, you can obtain an information about container related to an application attempt.

Parameters

- **application_id** (*str*) – The application id
- **attempt_id** (*str*) – The attempt id
- **container_id** (*str*) – The container id

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_application_attempt_containers(*application_id*, *attempt_id*)

With the application attempts API, you can obtain an information about container related to an application attempt.

Parameters

- **application_id** (*str*) – The application id
- **attempt_id** (*str*) – The attempt id

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_application_attempt_info(*application_id*, *attempt_id*)

With the application attempts API, you can obtain an extended info about an application attempt.

Parameters

- **application_id** (*str*) – The application id
- **attempt_id** (*str*) – The attempt 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_kill(*application_id*)

(This feature is currently in the alpha stage and may change in the future)

With the application kill API, you can kill an application that is not in FINISHED or FAILED state.

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

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_application_state(*application_id*)

(This feature is currently in the alpha stage and may change in the future)

With the application state API, you can obtain the current state of an application.

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

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_application_statistics(*states=None*, *application_types=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 only works in Hadoop > 2.0.0

Parameters

- **states** (*List[str]*) – states of the applications. If states is not provided, the API will enumerate all application states and return the counts of them.
- **application_types** (*List[str]*) – types of the applications, specified as a comma-separated list. If application_types 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_application_timeout(*application_id*, *timeout_type*)

The Cluster Application Timeout resource contains information about timeout.

Parameters

- **application_id** (*str*) – The application id
- **timeout_type** (*str*) – Timeout type. Valid values are the members of the ApplicationTimeoutType enum. LIFETIME is currently the only valid value. .

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_application_timeouts(*application_id*)

Cluster Application Timeouts API can be used to get all configured timeouts of an application. When you run a GET operation on this resource, a collection of timeout objects is returned. Each timeout object is composed of a timeout type, expiry-time and remaining time in seconds.

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

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_applications(*state=None*, *states=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*, *application_types=None*, *application_tags=None*, *name=None*, *de_selects=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 [deprecated]
- **states** (*List[str]*) – applications matching the given application states
- **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
- **application_types** (*List[str]*) – applications matching the given application types, specified as a comma-separated list
- **application_tags** (*List[str]*) – applications matching any of the given application tags, specified as a comma-separated list
- **name** (*str*) – name of the application
- **de_selects** (*List[str]*) – a generic fields which will be skipped in the result

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_cancel_delegation_token(*delegation_token*)

(This feature is currently in the alpha stage and may change in the future)

API to cancel delegation token.

All delegation token requests must be carried out on a Kerberos authenticated connection(using SPNEGO). Carrying out operations on a non-kerberos connection will result in a FORBIDDEN response.

Parameters **delegation_token** (*str*) – Delegation token

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_change_application_priority(*application_id*, *priority*)

(This feature is currently in the alpha stage and may change in the future)

Update priority of a running or accepted app using a PUT request specifying the target priority.

To perform the PUT operation, authentication has to be setup for the RM web services. In addition, you must be authorized to move the app. Currently you can only move the app if you're using the Capacity scheduler or the Fair scheduler.

Please note that in order to move an app, you must have an authentication filter setup for the HTTP interface. The functionality requires that a username is set in the HttpServletRequest. If no filter is setup, the response will be an "UNAUTHORIZED" response.

Parameters

- **application_id** (*str*) – The application id
- **priority** (*int*) – application priority

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_change_application_queue(*application_id*, *queue*)

(This feature is currently in the alpha stage and may change in the future)

Move a running app to another queue using a PUT request specifying the target queue.

To perform the PUT operation, authentication has to be setup for the RM web services. In addition, you must be authorized to move the app. Currently you can only move the app if you're using the Capacity scheduler or the Fair scheduler.

Please note that in order to move an app, you must have an authentication filter setup for the HTTP interface. The functionality requires that a username is set in the HttpServletRequest. If no filter is setup, the response will be an "UNAUTHORIZED" response.

Parameters

- **application_id** (*str*) – The application id
- **queue** (*str*) – queue name

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_container_signal(*container_id*, *command*)

With the Container Signal API, you can send a signal to a specified container with one of the following commands: OUTPUT_THREAD_DUMP, GRACEFUL_SHUTDOWN and FORCEFUL_SHUTDOWN.

Parameters

- **container_id** (*str*) – container id

- **command** (*str*) – signal command

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_delete_reservation(*reservation_id*)

The Cluster Reservation API Update can be used to update existing reservations.Update of a Reservation works similarly to submit described above, but the user submits the reservation-id of an existing reservation to be updated. The semantics is a try-and-swap, successful operation will modify the existing reservation based on the requested update parameter, while a failed execution will leave the existing reservation unchanged.

Parameters **reservation_id** (*str*) – The id of the reservation to be deleted (the system automatically looks up the right queue from this)

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_get_application_priority(*application_id*)

(This feature is currently in the alpha stage and may change in the future)

With the application priority API, you can query the priority of a submitted app

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

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_get_application_queue(*application_id*)

(This feature is currently in the alpha stage and may change in the future)

With the application queue API, you can query the queue of a submitted app

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

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

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_modify_scheduler_conf_mutation(*data*)

(This feature is currently in the alpha stage and may change in the future)

API to modify the scheduler configuration

For data body definition refer to: (https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Scheduler_Configuration_Mutation_API)

Parameters **data** (*dict*) – sched-conf dictionary object

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_new_application()

(This feature is currently in the alpha stage and may change in the future)

With the New Application API, you can obtain an application-id which can then be used as part of the Cluster Submit Applications API to submit applications. The response also includes the maximum resource capabilities available on the cluster.

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_new_delegation_token(*renewer*)

(This feature is currently in the alpha stage and may change in the future)

API to create delegation token.

All delegation token requests must be carried out on a Kerberos authenticated connection(using SPNEGO). Carrying out operations on a non-kerberos connection will result in a FORBIDDEN response. In case of renewing a token, only the renewer specified when creating the token can renew the token. Other users(including the owner) are forbidden from renewing tokens.

Parameters **renewer** (*str*) – The user who is allowed to renew the delegation token

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_new_reservation()

(This feature is currently in the alpha stage and may change in the future)

Use the New Reservation API, to obtain a reservation-id which can then be used as part of the Cluster Reservation API Submit to submit reservations.

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_node_container_memory()

This endpoint allows clients to gather info on the maximum memory that can be allocated per container in the cluster. :returns: integer specifying the maximum memory that can be allocated in a container in the cluster

cluster_node_update_resource(*node_id*, *data*)

Update the total resources in a node.

For data body definition refer to: (https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Cluster_Node_Update_Resource_API)

Parameters **data** (*dict*) – resourceOption details

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_nodes(*states=None*)

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

Parameters **states** (*List[str]*) – the states of the node, specified as a comma-separated list valid values are: NEW, RUNNING, UNHEALTHY, DECOMMISSIONING, DECOMMISSIONED, LOST, REBOOTED, SHUTDOWN

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

Raises **yarn_api_client.errors.IllegalArgumentError** – if *healthy* incorrect

cluster_queue_partition(*candidate_queue*, *cluster_node_label*)

A queue can be divided into multiple partitions having different node labels. Given the candidate queue and parition node label, this extracts the partition we are interested in.

Parameters

- **candidate_queue** (*dict*) – queue dictionary
- **cluster_node_label** (*str*) – case sensitive node label name

Returns partition, None if not Found.

Return type dict

cluster_renew_delegation_token(*delegation_token*)

(This feature is currently in the alpha stage and may change in the future)

API to renew delegation token.

All delegation token requests must be carried out on a Kerberos authenticated connection(using SPNEGO). Carrying out operations on a non-kerberos connection will result in a FORBIDDEN response. In case of renewing a token, only the renewer specified when creating the token can renew the token. Other users(including the owner) are forbidden from renewing tokens.

Parameters **delegation_token** (*str*) – Delegation token

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_reservations(*queue=None*, *reservation_id=None*, *start_time=None*, *end_time=None*, *include_resource_allocations=None*)

The Cluster Reservation API can be used to list reservations. When listing reservations the user must specify the constraints in terms of a queue, reservation-id, start time or end time. The user must also specify whether or not to include the full resource allocations of the reservations being listed. The resulting page returns a response containing information related to the reservation such as the acceptance time, the user, the resource allocations, the reservation-id, as well as the reservation definition.

Parameters

- **queue** (*str*) – the queue name containing the reservations to be listed. if not set, this value will default to “default”
- **reservation_id** (*str*) – the reservation-id of the reservation which will be listed. If this parameter is present, start-time and end-time will be ignored.
- **start_time** (*str*) – reservations that end after this start-time will be listed. If unspecified or invalid, this will default to 0.
- **end_time** (*str*) – reservations that start after this end-time will be listed. If unspecified or invalid, this will default to Long.MaxValue.

- **include_resource_allocations** (*str*) – true or false. If true, the resource allocations of the reservation will be included in the response. If false, no resource allocations will be included in the response. This will default to false.

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

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*

cluster_scheduler_conf_mutation()

(This feature is currently in the alpha stage and may change in the future)

API to retrieve the scheduler's configuration that is currently loaded into scheduler's context.

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_scheduler_queue(*yarn_queue_name*)

Given a queue name, this function tries to locate the given queue in the object returned by scheduler endpoint.

The queue can be present inside a multilevel structure. This solution tries to locate the queue using breadth-first-search algorithm.

Parameters **yarn_queue_name** (*str*) – case sensitive queue name

Returns queue, None if not found

Return type dict

cluster_scheduler_queue_availability(*candidate_partition*, *availability_threshold*)

Checks whether the requested memory satisfies the available space of the queue This solution takes into consideration the node label concept in cluster. Following node labelling, the resources can be available in various partition. Given the partition data it tells you if the used capacity of this partition is spilling the threshold specified.

Parameters

- **candidate_parition** (*str*) – node label partition (case sensitive)
- **availability_threshold** (*float*) – value can range between 0 - 100 .

Returns Boolean

cluster_submit_application(*data*)

(This feature is currently in the alpha stage and may change in the future)

With the New Application API, you can obtain an application-id which can then be used as part of the Cluster Submit Applications API to submit applications. The response also includes the maximum resource capabilities available on the cluster.

For data body definition refer to: (https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Cluster_Applications_API.28Submit_Application.29)

Parameters **data** (*dict*) – Application details

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_submit_reservation(*data*)

The Cluster Reservation API can be used to submit reservations. When submitting a reservation the user specifies the constraints in terms of resources, and time that is required. The resulting response is successful if the reservation can be made. If a reservation-id is used to submit a reservation multiple times, the request will succeed if the reservation definition is the same, but only one reservation will be created. If the reservation definition is different, the server will respond with an error response. When the reservation is made, the user can use the reservation-id used to submit the reservation to get access to the resources by specifying it as part of Cluster Submit Applications API.

For data body definition refer to: (https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Cluster_Reservation_API_Submit)

Parameters **data** (*dict*) – Reservation details

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_update_application_timeout(*application_id*, *timeout_type*, *expiry_time*)

Update timeout of an application for given timeout type.

Parameters

- **application_id** (*str*) – The application id
- **timeout_type** (*str*) – Timeout type. Valid values are the members of the ApplicationTimeoutType enum. LIFETIME is currently the only valid value.
- **expiry_time** (*str*) – Time at which the application will expire in ISO8601 yyyy-MM-dd'T'HH:mm:ss.SSSZ format.

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

cluster_update_reservation(*data*)

The Cluster Reservation API Update can be used to update existing reservations.Update of a Reservation works similarly to submit described above, but the user submits the reservation-id of an existing reservation to be updated. The semantics is a try-and-swap, successful operation will modify the existing reservation based on the requested update parameter, while a failed execution will leave the existing reservation unchanged.

For data body definition refer to: (https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/ResourceManagerRest.html#Cluster_Reservation_API_Update)

Parameters **data** (*dict*) – Reservation details

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

get_active_endpoint()

The active address, port tuple to which this instance is associated. :return: str service_endpoint: Service endpoint URL corresponding to the active address of RM

scheduler_activities(*node_id=None*, *group_by=None*)

The scheduler activities RESTful API is available if you are using capacity scheduler and can fetch scheduler activities info recorded in a scheduling cycle.

The API returns a message that includes important scheduling activities info which has a hierarchical layout with following fields:

- Activities - Activities is the root object of scheduler activities.
- Allocations - Allocations are allocation attempts based on partition or reservation.
- Hierarchical Queues - Hierarchical Queues where the scheduler have been tried to allocate

containers to, each of them contains queue name, allocation state, optional diagnostic and optional children. * Applications - Applications are shown as children of leaf queue, each of them contains the basic info about the application. * Requests - Requests are shown as children of application, each of them contains the basic info about the request. * Nodes - Nodes are shown as children of request, each of them contains node id, allocation state, optional name which should appear after allocating or reserving a container on the node, and optional diagnostic which should present if failed to allocate or reserve a container on this node. For aggregated nodes grouped by allocation state and diagnostic, each of them contains allocation state, aggregated node IDs and optional diagnostic.

Parameters **`node_id`** (*`str`*) – specified node ID, if not specified, the scheduler will record the

scheduling activities info for the next scheduling cycle on all nodes. :param str group_by: aggregation type of application activities, currently only support “diagnostic” with which user can query aggregated activities grouped by allocation state and diagnostic :returns: API response object with JSON data :rtype: *`yarn_api_client.base.Response`*

CHAPTER 3

NodeManager API's.

class yarn_api_client.node_manager.**NodeManager**(*service_endpoint=None*, *timeout=30*, *auth=None*, *verify=True*)

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

If *service_endpoint* argument is *None* client will try to extract it from Hadoop configuration files.

Parameters

- **service_endpoint** (*str*) – NodeManager HTTP(S) address
- **timeout** (*int*) – API connection timeout in seconds
- **auth** (*AuthBase*) – Auth to use for requests
- **verify** (*boolean*) – Either a boolean, in which case it controls whether we verify the server's TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`

auxiliary_services()

With the auxiliary services API, you can obtain a collection of resources, each of which represents an auxiliary service. When you run a GET operation on this resource, you obtain a collection of auxiliary service information objects.

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

auxiliary_services_update(*data*)

A YARN admin can use a PUT operation to update the auxiliary services running on the NodeManager. The body of the request should be of the same format as an auxiliary services manifest file.

Parameters **data** (*dict*) – auxiliary services manifest file

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

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*

CHAPTER 4

MapReduce Application Master API's.

class yarn_api_client.application_master.**ApplicationMaster**(*service_endpoint=None*, *timeout=30*, *auth=None*, *verify=True*)

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

- **service_endpoint** (*str*) – ApplicationMaster HTTP(S) address
- **timeout** (*int*) – API connection timeout in seconds
- **auth** (*AuthBase*) – Auth to use for requests
- **verify** (*boolean*) – Either a boolean, in which case it controls whether we verify the server's TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`

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 (*application_id*, *job_id*)
With the job attempts API, you can obtain a collection of resources that represent the job attempts.

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_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_attempt_state (*application_id*, *job_id*, *task_id*, *attempt_id*)

With the task attempt state API, you can query the state of a submitted 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_attempt_state_kill (*application_id*, *job_id*, *task_id*, *attempt_id*)

Kill specific attempt using task attempt state API.

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*

CHAPTER 5

History Server API's.

class yarn_api_client.history_server.**HistoryServer**(*service_endpoint=None*, *timeout=30*, *auth=None*, *verify=True*)

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 *service_endpoint* argument is *None* client will try to extract it from Hadoop configuration files.

Parameters

- **service_endpoint** (*str*) – HistoryServer HTTP(S) address
- **timeout** (*int*) – API connection timeout in seconds
- **auth** (*AuthBase*) – Auth to use for requests
- **verify** (*boolean*) – Either a boolean, in which case it controls whether we verify the server's TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`

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.

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

Returns API response object with JSON data

Return type *yarn_api_client.base.Response*

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*, *job_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 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index