atlasapi Documentation

Release 0.14.1

Matthew G. Monteleone

Apr 24, 2023

CONTENTS:

1 atlasapi package **3**
- 1.1 atlasapi.atlas module . . . 3
- 1.2 atlasapi.atlas_types module . . . 5
- 1.3 atlasapi.alerts module . . . 5
- 1.4 atlasapi.clusters module . . . 5
- 1.5 atlasapi.events module . . . 10
- 1.6 atlasapi.measurements module . . . 11
- 1.7 atlasapi.whitelist module . . . 18
- 1.8 atlasapi.errors module . . . 18
- 1.9 atlasapi.network module . . . 22
- 1.10 atlasapi.settings module . . . 24
- 1.11 atlasapi.specs module . . . 27
- 1.12 atlasapi.maintenance_window module . . . 32
- 1.13 atlasapi.cloud_backup module . . . 33
- 1.14 atlasapi.lib module . . . 35
- 1.15 atlasapi.projects module . . . 37
- 1.16 atlasapi.organizations module . . . 38

2 Nested class for atlasapi.atlas::Atlas **39**
- 2.1 Atlas._Clusters . . . 39
- 2.2 Atlas._DatabaseUsers . . . 43
- 2.3 Atlas._Alerts . . . 44
- 2.4 Atlas._MaintenanceWindows . . . 46
- 2.5 Atlas._Hosts . . . 46
- 2.6 Atlas._Events . . . 49
- 2.7 Atlas._Whitelist . . . 50
- 2.8 Atlas._CloudBackups . . . 51
- 2.9 Atlas._Projects . . . 53
- 2.10 Atlas._Organizations . . . 54

3 atlascli - A Command line program for MongoDB Atlas **57**

4 Indices and tables **59**

Python Module Index **61**

Index **63**

Python Bindings for the Atlas Public API

CHAPTER

ONE

ATLASAPI PACKAGE

1.1 atlasapi.atlas module

Atlas module

Core module which provides access to MongoDB Atlas Cloud Provider APIs

class atlasapi.atlas.**AlertsGetAll**(*atlas*, *status*, *pageNum*, *itemsPerPage*)

Bases: *AtlasPagination*

Pagination for Alerts : Get All

fetch(*pageNum*, *itemsPerPage*)

Intermediate fetching

Parameters

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Users per Page

Returns
Response payload

Return type
dict

class atlasapi.atlas.**Atlas**(*user: str*, *password: str*, *group: ~typing.Optional[str] = None*, *auth_method: ~typing.Union[~requests.auth.HTTPBasicAuth, ~requests.auth.HTTPDigestAuth] = <class 'requests.auth.HTTPDigestAuth'>*)

Bases: object

Atlas constructor

Parameters

- **user** (*str*) – Atlas user
- **password** (*str*) – Atlas password
- **group** (*str*) – Atlas group
- **auth_method** (*Union[HTTPBasicAuth,HTTPDigestAuth]*) – Authentication method to use, defaults to digest, but you
- **Proxy.** (*can override to Basic if needed for use with a*) –

class atlasapi.atlas.**AtlasPagination**(*atlas*, *fetch*, *pageNum: int*, *itemsPerPage: int*)

Bases: object

Atlas Pagination Generic Implementation

Constructor

Parameters

- **atlas** (Atlas) – Atlas instance
- **fetch** (*function*) – The function "get_all" to call
- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Users per Page

class atlasapi.atlas.**CloudBackupSnapshotsGetAll**(*atlas*, *pageNum*, *itemsPerPage*)

Bases: *AtlasPagination*

Pagination for Database User : Get All

class atlasapi.atlas.**ClustersGetAll**(*atlas*, *pageNum*, *itemsPerPage*)

Bases: *AtlasPagination*

Pagination for Clusters : Get All

class atlasapi.atlas.**DatabaseUsersGetAll**(*atlas*, *pageNum*, *itemsPerPage*)

Bases: *AtlasPagination*

Pagination for Database User : Get All

class atlasapi.atlas.**EventsGetForProject**(*atlas: Atlas*, *since_datetime: datetime*, *pageNum: int*, *itemsPerPage: int*)

Bases: *AtlasPagination*

fetch(*pageNum*, *itemsPerPage*)

Intermediate fetching

Parameters

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Events per Page

Returns
Response payload

Return type
dict

class atlasapi.atlas.**EventsGetForProjectAndType**(*atlas: Atlas*, *event_type: AtlasEventTypes*, *since_datetime: datetime*, *pageNum: int*, *itemsPerPage: int*)

Bases: *AtlasPagination*

fetch(*pageNum*, *itemsPerPage*)

Intermediate fetching

Parameters

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Events per Page

Returns
Response payload

Return type
dict

class atlasapi.atlas.**HostsGetAll**(*atlas: Atlas*, *pageNum: int*, *itemsPerPage: int*)
Bases: *AtlasPagination*

Pagination for Processes : Get All

class atlasapi.atlas.**OrganizationProjectsGetAll**(*atlas: Atlas*, *org_id: str*, *pageNum*, *itemsPerPage*)
Bases: *AtlasPagination*

Pagination for Database User : Get All

fetch(*pageNum*, *itemsPerPage*)
Intermediate fetching

Parameters

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Events per Page

Returns
Response payload

Return type
dict

class atlasapi.atlas.**WhitelistGetAll**(*atlas*, *pageNum*, *itemsPerPage*)
Bases: *AtlasPagination*

Pagination for Database User : Get All

1.2 atlasapi.atlas_types module

1.3 atlasapi.alerts module

class atlasapi.alerts.**Alert**(*data_dict: dict*)
Bases: object

1.4 atlasapi.clusters module

Classes related to Atlas clusters.

Supports the creation and configuration of Atlas clusters of various types.

Enums are used in order to minimize invalid configuration values.

class atlasapi.clusters.**AdvancedOptions**(*failIndexKeyTooLong: Optional[bool] = None*, *javascriptEnabled: Optional[bool] = None*, *minimumEnabledTlsProtocol: Optional[TLSProtocols] = None*, *noTableScan: Optional[bool] = None*, *oplogSizeMB: Optional[int] = None*, *sampleSizeBIConnector: Optional[int] = None*, *sampleRefreshIntervalBIConnector: Optional[int] = None*)

Bases: object

Container for Atlas Cluster Advanced options

Parameters

- **failIndexKeyTooLong** – When true, documents can only be updated or inserted if, for all indexed fields on the target collection, the corresponding index entries do not exceed 1024 bytes.When false, mongod writes documents that breach the limit but does not index them.
- **javascriptEnabled** – When true, the cluster allows execution of operations that perform server-side executions of JavaScript. When false, the cluster disables execution of those operations.
- **minimumEnabledTlsProtocol** – The minimum Transport Layer Security (TLS) version the cluster accepts for incoming connections.
- **noTableScan** – When true, the cluster disables the execution of any query that requires a collection scan to return results. When false, the cluster allows the execution of those operations.
- **oplogSizeMB** – The custom oplog size of the cluster. A value of null indicates that the cluster uses the default oplog size calculated by Atlas.
- **sampleSizeBIConnector** – Number of documents per database to sample when gathering schema information.
- **sampleRefreshIntervalBIConnector** – Interval in seconds at which the mongosqld process re-samples data to create its relational schema.

property as_dict: dict

Returns a json-able dict of only non-null properties.

Returns

classmethod fill_from_dict(*data_dict: dict*)

Fills the advanced options object from an Atlas Dict

Parameters

data_dict (dict) – A dict as returned from Atlas

Returns

class atlasapi.clusters.**AtlasBasicReplicaSet**(*name: str*, *size:* InstanceSizeName = *InstanceSizeName.M10*, *disk_size: int = 10*, *provider:* ProviderName = *ProviderName.AWS*, *region: str = 'US_WEST_2'*, *version:* MongoDBMajorVersion = *MongoDBMajorVersion.v4_0*)

Bases: object

Helper object for the creation of a basic replica set with default options.

Only the cluster name is required.

Other parameters will default to An M10 cluster on AWS US_WEST_2 running 4.0 with a 10 GB disk.

Args:

name: The name given to the cluster/replica set. size: The InstanceSizeName of the cluster/replica set disk_size: Size of disks on all members provider: The Cloud provider region: The region in the cloud provider version: The MongoDB major version

as_dict()

class atlasapi.clusters.**ClusterConfig**(*backup_enabled: bool = False*, *cluster_type: ClusterType = ClusterType.REPLICASET*, *disk_size_gb: int = 32*, *name: Optional[str] = None*, *mongodb_major_version: MongoDBMajorVersion = MongoDBMajorVersion.v4_4*, *mongodb_version: Optional[str] = None*, *num_shards: int = 1*, *mongo_uri: Optional[str] = None*, *mongo_uri_updated: Optional[str] = None*, *mongo_uri_with_options: Optional[str] = None*, *paused: bool = False*, *pit_enabled: bool = False*, *replication_factor: Optional[int] = None*, *state_name: Optional[ClusterStates] = None*, *autoscaling: Optional[dict] = None*, *replication_specs: Optional[ReplicationSpecs] = None*, *srv_address: Optional[str] = None*, *providerSettings: Optional[ProviderSettings] = None*, *links: Optional[list] = None*, *id: Optional[str] = None*, *create_date: Optional[datetime] = None*)

Bases: object

Stores the Atlas Cluster Config, is sent back to the API for any reconfiguations.

https://docs.atlas.mongodb.com/reference/api/clusters-get-one/#http-response-elements

Args:

backup_enabled: cluster_type: disk_size_gb: name: mongodb_major_version: mongodb_version: num_shards: mongo_uri: mongo_uri_updated: mongo_uri_with_options: paused: pit_enabled: replication_factor: state_name: autoscaling: replication_specs: srv_address: providerSettings: links:

as_create_dict() → dict

Returns the config object in a format acceptable for the POST (create) endpoint.

Removes properties which are read-only.

TODO: Refactor to identify which properties are RO in the spec, and automatically loop through and remove.

Returns

dict: A dict containing a valid create object for the POST endpoint.

as_dict() → dict

as_modify_dict() → dict

Returns the config object in a format acceptable for the PATCH (modify) endpoint.

Removes properties which are read-only.

TODO: Refactor to identify which properties are RO in the spec, and automatically loop through and remove.

Returns

dict: A dict containing a valid create object for the POST endpoint.

classmethod fill_from_dict(*data_dict: dict*)

class atlasapi.clusters.**ClusterStates**(*value*)

Bases: Enum

The states of Atlas clusters. RO attribute.

CREATING = 'Creating'

DELETED = 'Deleted'

DELETING = 'Deleting'

IDLE = 'Idle'

REPAIRING = 'Repairing'

UNKNOWN = 'Unknown'

UPDATING = 'Updating'

class atlasapi.clusters.**InstanceSizeName**(*value*)

Bases: Enum

The Atlas instance sizes, which are equivalent across all providers.

NVME suggixed types provide loval NVME disk.

M0 = 'M0'

M10 = 'M10'

M100 = 'M100'

M140 = 'M140'

M2 = 'M2'

M20 = 'M20'

M200 = 'M200'

M200_NVME = 'M200_NVME'

M30 = 'M30'

M300 = 'M300'

M40 = 'M40'

M400 = 'M400'

M400_NVME = 'M400_NVME'

M40_NVME = 'M40 NVME'

M5 = 'M5'

M50 = 'M50'

M50_NVME = 'M50 NVME'

M60 = 'M60'

M60_NVME = 'M60 NVME'

M80 = 'M80'

M80_NVME = 'M80 NVME'

R200 = 'R200'

R300 = 'R300'

R40 = 'R40'

R400 = 'R400'

R50 = 'R50'

R60 = 'R60'

R700 = 'R700'

R80 = 'R80'

class atlasapi.clusters.**ProviderSettings**(*size: InstanceSizeName = InstanceSizeName.M10*, *provider: ProviderName = ProviderName.AWS*, *region: str = 'US_WEST_1'*, *autoScaling: Optional[dict] = None*, *diskIOPS: Optional[int] = None*, *encryptEBSVolume: bool = True*, *volumeType: VolumeTypes = VolumeTypes.STANDARD*)

Bases: object

as_dict() → dict

classmethod from_dict(*data_dict: dict*)

class atlasapi.clusters.**RegionConfig**(*electable_node_count: int = 3*, *priority: int = 7*, *read_only_node_count: int = 0*, *analytics_node_count: int = 0*)

Bases: object

Configuration object for each region.

Allows for the configuration of each region independently. Includes sane defaults.

Parameters

- **electable_node_count** (*int*) – Number of electable nodes.
- **priority** – Priority of the region in the replica set.
- **read_only_node_count** – Count of read_only nodes.
- **analytics_node_count** – Count of analytics nodes.

class atlasapi.clusters.**ReplicationSpecs**(*id: Optional[str] = '45f46aed-ca87-4928-9fea-d40af71092bc'*, *num_shards: Optional[int] = 1*, *zone_name: Optional[str] = None*, *regions_config: Optional[dict] = None*)

Bases: object

as_create_dict()

as_dict()

classmethod from_dict(*data_dict: dict*)

class atlasapi.clusters.**ShardedClusterConfig**(*backup_enabled: bool = False*, *cluster_type: ClusterType = ClusterType.REPLICASET*, *disk_size_gb: int = 32*, *name: Optional[str] = None*, *mongodb_major_version: MongoDBMajorVersion = MongoDBMajorVersion.v4_0*, *mongodb_version: Optional[str] = None*, *num_shards: int = 1*, *mongo_uri: Optional[str] = None*, *mongo_uri_updated: Optional[str] = None*, *mongo_uri_with_options: Optional[str] = None*, *paused: bool = False*, *pit_enabled: bool = False*, *replication_factor: Optional[int] = None*, *state_name: Optional[ClusterStates] = None*, *autoscaling: dict = {}*, *replication_specs: list = []*, *srv_address: Optional[str] = None*, *providerSettings: Optional[ProviderSettings] = None*, *links: Optional[list] = None*, *id: Optional[str] = None*)

Bases: *ClusterConfig*

as_dict() → dict

class atlasapi.clusters.**TLSProtocols**(*value*)

Bases: Enum

An enumeration.

TLS1_0 = 'TLS1_0'

TLS1_1 = 'TLS1_1'

TLS1_2 = 'TLS1_2'

TLS1_3 = 'TLS1_3'

class atlasapi.clusters.**VolumeTypes**(*value*)

Bases: Enum

The volume types available on atlas

PROVISIONED = 'Provisioned'

STANDARD = 'Standard'

1.5 atlasapi.events module

class atlasapi.events.**AtlasCPSEvent**(*value_dict: dict*)

Bases: _AtlasBaseEvent

class atlasapi.events.**AtlasClusterEvent**(*value_dict: dict*)

Bases: _AtlasBaseEvent

class atlasapi.events.**AtlasDataExplorerEvent**(*value_dict: dict*)

Bases: _AtlasUserBaseEvent

class atlasapi.events.**AtlasEvent**(*value_dict: dict*)

Bases: _AtlasBaseEvent

class atlasapi.events.**AtlasFeatureEvent**(*value_dict: dict*)

Bases: _AtlasUserBaseEvent

class atlasapi.events.**AtlasHostEvent**(*value_dict: dict*)

Bases: _AtlasBaseEvent

atlasapi.events.**atlas_event_factory**(*value_dict: dict*) → Union[*AtlasEvent*, *AtlasDataExplorerEvent*, *AtlasClusterEvent*, *AtlasHostEvent*, *AtlasFeatureEvent*, *AtlasCPSEvent*]

1.6 atlasapi.measurements module

class atlasapi.measurements.**AtlasMeasurement**(*name:* AtlasMeasurementTypes, *period:* AtlasPeriods, *granularity:* AtlasGranularities, *units: Optional[str] = None*, *measurements: Optional[List[*AtlasMeasurementValue*]] = None*)

Bases: object

A point in time container for an Atlas measurement.

For a certain period, granularity and measurementType holds a list fo measurementValues.

Parameters

- **name** (AtlasMeasurementTypes) – The name of the measurement type
- **units** (*Text*) – Descriptive text of units used.
- **period** (AtlasPeriods) – The period the measurement covers
- **granularity** (AtlasGranularities) – The granularity used for the measurement
- **measurements** (*List[*AtlasMeasurementValue*]*) – A list of the actual measurement values

property **as_dict**

Returns the measurement as a dict, including the computed properties.

Return type
dict

property **date_end**

The date of the last measurement

Returns
The date of the last measurement.

Return type
datetime

property **date_start**

The date of the first measurement.

Returns
The date of the first measurement.

Return type
datetime

property measurement_stats: *StatisticalValues*

Returns a statistical info for measurement data

property measurement_stats_friendly: *StatisticalValuesFriendly*

Returns statistical info for measurement data in friendly bytes format

property measurements: Iterable[*AtlasMeasurementValue*]

Getter for the measurements.

Returns
An iterator containing values objects.

Return type
Iterator[*AtlasMeasurementValue*]

measurements_as_tuples()

property measurements_count

The count of measurements

Returns
The count of measurements in the set

Return type
int

class atlasapi.measurements.**AtlasMeasurementTypes**

Bases: _GetAll

Helper class for all available atlas measurements.

All classes and embedded classes have a get_all class method that returns an iterator of all measurements and sub measurements.

class Asserts

Bases: _GetAll

msg = 'ASSERT_MSG'

regular = 'ASSERT_REGULAR'

user = 'ASSERT_USER'

warning = 'ASSERT_WARNING'

class CPU

Bases: _GetAll

class Process

Bases: _GetAll

children_kernel = 'PROCESS_CPU_CHILDREN_KERNEL'

children_user = 'PROCESS_CPU_CHILDREN_USER'

kernel = 'PROCESS_CPU_KERNEL'

user = 'PROCESS_CPU_USER'

class ProcessNormalized

Bases: _GetAll

children_kernel = 'PROCESS_NORMALIZED_CPU_CHILDREN_KERNEL'

children_user = 'PROCESS_NORMALIZED_CPU_CHILDREN_USER'

kernel = 'PROCESS_NORMALIZED_CPU_KERNEL'

user = 'PROCESS_NORMALIZED_CPU_USER'

class System

Bases: _GetAll

guest = 'SYSTEM_CPU_GUEST'

iowait = 'SYSTEM_CPU_IOWAIT'

irq = 'SYSTEM_CPU_IRQ'

kernel = 'SYSTEM_CPU_KERNEL'

nice = 'SYSTEM_CPU_NICE'

softirq = 'SYSTEM_CPU_SOFTIRQ'

steal = 'SYSTEM_CPU_STEAL'

user = 'SYSTEM_CPU_USER'

class SystemNormalized

Bases: _GetAll

guest = 'SYSTEM_NORMALIZED_CPU_GUEST'

iowait = 'SYSTEM_NORMALIZED_CPU_IOWAIT'

irq = 'SYSTEM_NORMALIZED_CPU_IRQ'

kernel = 'SYSTEM_NORMALIZED_CPU_KERNEL'

nice = 'SYSTEM_NORMALIZED_CPU_NICE'

softirq = 'SYSTEM_NORMALIZED_CPU_SOFTIRQ'

steal = 'SYSTEM_NORMALIZED_CPU_STEAL'

user = 'SYSTEM_NORMALIZED_CPU_USER'

class Cache

Bases: _GetAll

bytes_read = 'CACHE_BYTES_READ_INTO'

bytes_written = 'CACHE_BYTES_WRITTEN_FROM'

dirty = 'CACHE_DIRTY_BYTES'

used = 'CACHE_USED_BYTES'

class Cursors

Bases: _GetAll

open = 'CURSORS_TOTAL_OPEN'

timed_out = 'CURSORS_TOTAL_TIMED_OUT'

class Db

Bases: _GetAll

data_size = 'DB_DATA_SIZE_TOTAL'

storage = 'DB_STORAGE_TOTAL'

class Disk

Bases: _GetAll

class Free

Bases: _GetAll

percent_fee = 'DISK_PARTITION_SPACE_PERCENT_FREE'

percent_free_max = 'MAX_DISK_PARTITION_SPACE_PERCENT_FREE'

percent_used = 'DISK_PARTITION_SPACE_PERCENT_USED'

percent_used_max = 'MAX_DISK_PARTITION_SPACE_PERCENT_USED'

space_free = 'DISK_PARTITION_SPACE_FREE'

space_free_max = 'MAX_DISK_PARTITION_SPACE_FREE'

used = 'DISK_PARTITION_SPACE_USED'

used_max = 'MAX_DISK_PARTITION_SPACE_USED'

class IOPS

Bases: _GetAll

read = 'DISK_PARTITION_IOPS_READ'

read_max = 'MAX_DISK_PARTITION_IOPS_READ'

total = 'DISK_PARTITION_IOPS_TOTAL'

total_max = 'MAX_DISK_PARTITION_IOPS_TOTAL'

write = 'DISK_PARTITION_IOPS_WRITE'

write_max = 'MAX_DISK_PARTITION_IOPS_WRITE'

class Latency

Bases: _GetAll

read = 'DISK_PARTITION_LATENCY_READ'

read_max = 'MAX_DISK_PARTITION_LATENCY_READ'

write = 'DISK_PARTITION_LATENCY_WRITE'

write_max = 'MAX_DISK_PARTITION_LATENCY_WRITE'

class Util

Bases: _GetAll

util = 'DISK_PARTITION_UTILIZATION'

util_max = 'MAX_DISK_PARTITION_UTILIZATION'

class DocumentMetrics

Bases: _GetAll

deleted = 'DOCUMENT_METRICS_DELETED'

inserted = 'DOCUMENT_METRICS_INSERTED'

returned = 'DOCUMENT_METRICS_RETURNED'

updated = 'DOCUMENT_METRICS_UPDATED'

class ExtraInfo

Bases: _GetAll

page_faults = 'EXTRA_INFO_PAGE_FAULTS'

class GlobalLockCurrentQueue

Bases: _GetAll

readers = 'GLOBAL_LOCK_CURRENT_QUEUE_READERS'

total = 'GLOBAL_LOCK_CURRENT_QUEUE_TOTAL'

writers = 'GLOBAL_LOCK_CURRENT_QUEUE_WRITERS'

class Memory

Bases: _GetAll

mapped = 'MEMORY_MAPPED'

resident = 'MEMORY_RESIDENT'

virtual = 'MEMORY_VIRTUAL'

class Namespaces

Bases: _GetAll

Metrics regarding namespaces (databases) on each host.

As found in dbstats (https://www.mongodb.com/docs/manual/reference/command/dbStats/)

collection_count = 'DATABASE_COLLECTION_COUNT'

data_size = 'DATABASE_DATA_SIZE'

extent_count = 'DATABASE_EXTENT_COUNT'

index_count = 'DATABASE_INDEX_COUNT'

index_size = 'DATABASE_INDEX_SIZE'

object_count = 'DATABASE_OBJECT_COUNT'

object_size = 'DATABASE_AVERAGE_OBJECT_SIZE'

storage_size = 'DATABASE_STORAGE_SIZE'

view_count = 'DATABASE_VIEW_COUNT'

class Network

Bases: _GetAll

bytes_id = 'NETWORK_BYTES_IN'

bytes_in = 'NETWORK_BYTES_IN'

bytes_out = 'NETWORK_BYTES_OUT'

num_requests = 'NETWORK_NUM_REQUESTS'

class Opcounter

Bases: _GetAll

class Repl

Bases: _GetAll

cmd = 'OPCOUNTER_REPL_CMD'

delete = 'OPCOUNTER_REPL_DELETE'

insert = 'OPCOUNTER_REPL_INSERT'

update = 'OPCOUNTER_REPL_UPDATE'

cmd = 'OPCOUNTER_CMD'

delete = 'OPCOUNTER_DELETE'

getmore = 'OPCOUNTER_GETMORE'

insert = 'OPCOUNTER_INSERT'

query = 'OPCOUNTER_QUERY'

update = 'OPCOUNTER_UPDATE'

class Operations

Bases: _GetAll

class ExecutionTime

Bases: _GetAll

commands = 'OP_EXECUTION_TIME_COMMANDS'

reads = 'OP_EXECUTION_TIME_READS'

writes = 'OP_EXECUTION_TIME_WRITES'

scan_and_order = 'OPERATIONS_SCAN_AND_ORDER'

class Oplog

Bases: _GetAll

master_time = 'OPLOG_MASTER_TIME'

rate = 'OPLOG_RATE_GB_PER_HOUR'

class QueryExecutor

Bases: _GetAll

scanned = 'QUERY_EXECUTOR_SCANNED'

scanned_objects = 'QUERY_EXECUTOR_SCANNED_OBJECTS'

class QueryTargetingScanned

Bases: _GetAll

objects_per_returned = 'QUERY_TARGETING_SCANNED_OBJECTS_PER_RETURNED'

per_returned = 'QUERY_TARGETING_SCANNED_PER_RETURNED'

class TicketsAvailable

Bases: _GetAll

reads = 'TICKETS_AVAILABLE_READS'

writes = 'TICKETS_AVAILABLE_WRITE'

connections = 'CONNECTIONS'

class atlasapi.measurements.AtlasMeasurementValue(*value_dict: dict*)

Bases: object

as_dict() → dict

property as_tuple: Tuple[datetime, OptionalFloat]

Returns a MeasurementValue as a tuple, timestamp first. :rtype: Tuple[datetime,OptionalFloat] :return: A tuple with a datetime and a float

property value_float: Optional[float]

property value_int: Optional[int]

class atlasapi.measurements.StatisticalValues(*data_list: list*)

Bases: object

class atlasapi.measurements.StatisticalValuesFriendly(*data_list: list*, *data_type: Optional[str] = None*)

Bases: object

atlasapi.measurements.clean_list(*data_list: list*) → list

Returns a list with any none values removed

Parameters

data_list (*list*) – The list to be cleaned

Returns (list): The list cleaned of None values.

1.7 atlasapi.whitelist module

class atlasapi.whitelist.**WhitelistEntry**(*cidrBlock: Optional[str] = None*, *comment: Optional[str] = None*, *ipAddress: Optional[str] = None*, *links: Optional[list] = None*, *last_used: Optional[str] = None*, *count: Optional[int] = None*, *last_used_address: Optional[str] = None*)

Bases: object

as_dict() → dict

Dumps obj as a json valid dict. :return:

classmethod fill_from_dict(*data_dict: dict*)

Fills the object from the standard Atlas API dictionary. :type data_dict: dict :param data_dict: :return:

1.8 atlasapi.errors module

Errors module

Provides all specific Exceptions

exception atlasapi.errors.**ErrAtlasBackupError**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Atlas Backup

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasBadRequest**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Bad Request

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasConflict**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Conflict

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasDuplicateClusterName**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Duplicate Clustername

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasForbidden**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Forbidden

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasForbiddenWL**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Forbidden by WhiteList

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasGeneric**(*msg: str*, *c: int*, *details: dict*)

Bases: Exception

Atlas Generic Exception

Constructor

Parameters

- **msg** (*str*) – Short description of the error
- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

getAtlasResponse() → Tuple[int, dict]

Get details about the Atlas response

Returns
HTTP code, Response payload

Return type
int, str

exception atlasapi.errors.**ErrAtlasJobError**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Job error Clustername

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasMethodNotAllowed**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Method Not Allowed

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasNotFound**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Not Found

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasRestoreConflictError**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : RestoreConflictError

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasServerErrors**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Server Errors

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrAtlasUnauthorized**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Unauthorized

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrConfirmationRequested**(*msg*)

Bases: Exception

No Confirmation provided

Constructor

Parameters
msg (*str*) – Short description of the error

exception atlasapi.errors.**ErrMaintenanceError**(*c*, *details*)

Bases: *ErrAtlasGeneric*

Atlas : Atlas MaintenanceRelatedError

Constructor

Parameters

- **c** (*int*) – HTTP code
- **details** (*dict*) – Response payload

exception atlasapi.errors.**ErrPagination**

Bases: Exception

An issue occurs during a "Get All" function

exception atlasapi.errors.**ErrPaginationLimits**(*error_code*)

Bases: Exception

Out of limit on 'pageNum' or 'itemsPerPage' parameters

Constructor

Parameters
error_code (*int*) – ERR_PAGE_NUM or ERR_ITEMS_PER_PAGE

ERR_ITEMS_PER_PAGE = 1

ERR_PAGE_NUM = 0

checkAndRaise(*itemsPerPage*)

Check and Raise an Exception if needed

Parameters

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of items per Page

Raises
ErrPaginationLimits – If we are out of limits

exception atlasapi.errors.**ErrRole**

Bases: Exception

A role is not compatible with Atlas

1.9 atlasapi.network module

Network module

Module which handles the basic network operations with the Atlas API>

class atlasapi.network.**Network**(*user*, *password*, *AuthMethod: ~typing.Union[~requests.auth.HTTPDigestAuth, ~requests.auth.HTTPBasicAuth] = <class 'requests.auth.HTTPDigestAuth'>*)

Bases: object

Network constructor

Parameters

- **user** (*str*) – user
- **password** (*str*) – password

answer(*c*, *details: Union[dict, BytesIO]*)

Answer will provide all necessary feedback for the caller

Parameters

- **c** (*int*) – HTTP Code
- **details** (*dict*) – Response payload

Returns
Response payload

Return type
dict

Raises

- *ErrAtlasBadRequest* –
- *ErrAtlasUnauthorized* –
- *ErrAtlasForbidden* –
- *ErrAtlasNotFound* –
- *ErrAtlasMethodNotAllowed* –
- *ErrAtlasConflict* –
- *ErrAtlasServerErrors* –

delete(*uri*)

Delete request

Parameters
uri (*str*) – URI

Returns
API response

Return type
Json

Raises
Exception – Network issue

get(*uri*)

Get request

Parameters
uri (*str*) – URI

Returns
API response

Return type
Json

Raises
Exception – Network issue

get_big(*uri*, *params: Optional[dict] = None*)

Get request (max results)

This is a temporary fix until we re-factor pagination.

Parameters

- **params** – dict of parameters that should be sent on the path
- **uri** (*str*) – URI

Returns
API response

Return type
Json

Raises
Exception – Network issue

get_file(*uri*)

Get request which returns a binary file

Parameters
uri (*str*) – URI

Returns
API response as file

Return type
Binary File

Raises
Exception – Network issue

patch(*uri*, *payload*)

Patch request

Parameters

- **uri** (*str*) – URI
- **payload** (*dict*) – Content to patch

Returns
API response

Return type
Json

Raises
Exception – Network issue

post(*uri*, *payload*)

Post request

Parameters

- **uri** (*str*) – URI
- **payload** (*dict*) – Content to post

Returns
API response

Return type
Json

Raises
Exception – Network issue

atlasapi.network.**merge**(*dict1*, *dict2*)

1.10 atlasapi.settings module

Settings module

Provides few constants, APIs endpoints.

class atlasapi.settings.**Settings**

Bases: object

ACCEPTED = 202

BAD_REQUEST = 400

BASE_URL = 'https://cloud.mongodb.com'

CONFLICT = 409

CREATED = 201

FORBIDDEN = 403

METHOD_NOT_ALLOWED = 405

NOTFOUND = 404

NO_CONTENT = 204

```
SERVER_ERRORS = 500

SUCCESS = 200

UNAUTHORIZED = 401

URI_STUB = '/api/atlas/v1.0'
```

```
api_resources = {'Alerts': {'Acknowledge an Alert':
'/api/atlas/v1.0/groups/%s/alerts/%s', 'Get All Alerts':
'/api/atlas/v1.0/groups/%s/alerts?pageNum=%d&itemsPerPage=%d', 'Get All Alerts with
status': '/api/atlas/v1.0/groups/%s/alerts?status=%s&pageNum=%d&itemsPerPage=%d',
'Get an Alert': '/api/atlas/v1.0/groups/%s/alerts/%s'}, 'Cloud Backup': {'Delete
snapshot by SNAPSHOT-ID': '/api/atlas/v1.0/groups/{GROUP_ID}/clusters/
{CLUSTER_NAME}/backup/snapshots/{SNAPSHOT_ID}', 'Get all Cloud Backups for cluster':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/backup/snapshots', 'Get
snapshot by SNAPSHOT-ID': '/api/atlas/v1.0/groups/{GROUP_ID}/clusters/
{CLUSTER_NAME}/backup/snapshots/{SNAPSHOT_ID}', 'Take an on-demand snapshot':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/backup/snapshots'},
'Cloud Backup Restore Jobs': {'Cancel manual download restore job by job_id':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/backup/restoreJobs/
{JOB_ID}', 'Get Cloud Backup restore job by cluster':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/backup/restoreJobs/
{JOB_ID}', 'Get all Cloud Backup restore jobs by cluster':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/backup/restoreJobs',
'Restore snapshot by cluster':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/backup/restoreJobs'},
'Clusters': {'Advanced Configuration Options':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/processArgs', 'Create a
Cluster': '/api/atlas/v1.0/groups/{GROUP_ID}/clusters/', 'Delete a Cluster':
'/api/atlas/v1.0/groups/%s/clusters/%s', 'Get All Clusters':
'/api/atlas/v1.0/groups/%s/clusters?pageNum=%d&itemsPerPage=%d', 'Get a Single
Cluster': '/api/atlas/v1.0/groups/%s/clusters/%s', 'Modify a Cluster':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}', 'Test Failover':
'/api/atlas/v1.0/groups/{GROUP_ID}/clusters/{CLUSTER_NAME}/restartPrimaries'},
'Database Users': {'Create a Database User':
'/api/atlas/v1.0/groups/%s/databaseUsers', 'Delete a Database User':
'/api/atlas/v1.0/groups/%s/databaseUsers/admin/%s', 'Get All Database Users':
'/api/atlas/v1.0/groups/%s/databaseUsers?pageNum=%d&itemsPerPage=%d', 'Get a Single
Database User': '/api/atlas/v1.0/groups/%s/databaseUsers/admin/%s', 'Update a
Database User': '/api/atlas/v1.0/groups/%s/databaseUsers/admin/%s'}, 'Events':
{'Get All Project Events': '/api/atlas/v1.0/groups/{group_id}/events?
includeRaw=true&pageNum={page_num}&itemsPerPage={items_per_page}', 'Get Project
Events Since Date': '/api/atlas/v1.0/groups/{group_id}/events?
includeRaw=true&pageNum={page_num}&itemsPerPage={items_per_page}&minDate={min_date}'},
'Maintenance Windows': {'Defer Maintenance Window':
'/api/atlas/v1.0/groups/{GROUP_ID}/maintenanceWindow/defer', 'Delete Maintenance
Window': '/api/atlas/v1.0/groups/{GROUP_ID}/maintenanceWindow', 'Get Maintenance
Window': '/api/atlas/v1.0/groups/{GROUP_ID}/maintenanceWindow', 'Update Maintenance
Window': '/api/atlas/v1.0/groups/{GROUP_ID}/maintenanceWindow'}, 'Monitoring and
Logs': {'Get Available Databases for Process':
'/api/atlas/v1.0/groups/{group_id}/processes/{host}:{port}/databases', 'Get
Available Disks for Process':
'/api/atlas/v1.0/groups/{group_id}/processes/{host}:{port}/disks', 'Get Measurements
of a Database for Process': '/api/atlas/v1.0/groups/{group_id}/processes/
{host}:{port}/databases/{database_name}/measurements', 'Get Measurements of a Disk
for Process': '/api/atlas/v1.0/groups/{group_id}/processes/{host}:{port}/disks/
{disk_name}/measurements', 'Get all processes for group':
'/api/atlas/v1.0/groups/{group_id}/processes?
pageNum={page_num}&itemsPerPage={items_per_page}', 'Get information for process in
group': '/api/atlas/v1.0/groups/%s/processes/%s:&s?pageNum=%d&itemsPerPage=%d',
'Get list of databases for host':
'/api/atlas/v1.0/groups/{GROUP-ID}/processes/{HOST}:{PORT}/databases', 'Get list of
disks or partitions for host.':
'/api/atlas/v1.0/groups/{GROUP-ID}/processes/{HOST}:{PORT}/disks', 'Get measurement
for host': '/api/atlas/v1.0/groups/{group_id}/processes/{host}:{port}/measurements?
granularity={granularity}&period={period}&m={measurement}', 'Get measurements of
database for host.': '/api/atlas/v1.0/groups/{GROUP-ID}/processes/{HOST}:{PORT}/
```

databaseName = 'admin'

file_request_timeout = 360

itemsPerPage: int = 500

itemsPerPageMax: int = 2000

itemsPerPageMin: int = 1

pageNum = 1

requests_timeout = 10

1.11 atlasapi.specs module

Specs module

Provides some high level objects useful to use the Atlas API.

class atlasapi.specs.**AlertStatusSpec**

Bases: object

Alert Status

CLOSED = 'CLOSED'

OPEN = 'OPEN'

TRACKING = 'TRACKING'

class atlasapi.specs.**DatabaseUsersPermissionsSpecs**(*username: str*, *password: Optional[str] = None*, *aws_iam_type: Optional[IAMType] = None*, *databaseName='admin'*)

Bases: object

Permissions spec for Database User

Constructor

Parameters

- **username** (*str*) – Username of the DB
- **password** (*str*) – Password for the username
- **aws_iam_type** (IAMType) – AWS IAM method by which the database applies IAM credentials to authenticates the database user. Atlas defaults to NONE. (optional)

Keyword Arguments
databaseName (*str*) – Auth Database Name

add_role(*databaseName: str*, *roleName: str*, *collectionName: Optional[OptionalStr] = None*)

Add one role

Parameters

- **databaseName** (str :param roleName: :param databaseName: :type collectionName: str) – Database Name

- **roleName** (str) – role

Keyword Arguments
collectionName (*str*) – Collection

Raises
ErrRole – role not compatible with the databaseName and/or collectionName

TODO: Need to test if this works correctly, looks like their may be a type problem.

add_roles(*databaseName: str*, *roleNames: List[*RoleSpecs*]*, *collectionName: Optional[str] = None*)

Add multiple roles

Parameters

- **databaseName** (str :param databaseName: Database Name :param roleNames: roles :param collectionName: Collection) – Database Name
- **roleNames** (*list of* RoleSpecs) – roles

Keyword Arguments
collectionName (*str*) – Collection

Raises
ErrRoleException – role not compatible with the databaseName and/or collectionName

clear_roles()

getSpecs() → dict

Get specs

Returns
Representation of the object

Return type
dict

remove_role(*databaseName*, *roleName*, *collectionName=None*)

Remove one role

Parameters

- **databaseName** (*str*) – Database Name
- **roleName** (RoleSpecs) – role

Keyword Arguments
collectionName (*str*) – Collection

remove_roles(*databaseName*, *roleNames*, *collectionName=None*)

Remove multiple roles

Parameters

- **collectionName** (*str*) –
- **databaseName** (*str*) – Database Name
- **roleNames** (*list of* RoleSpecs) – roles

Keyword Arguments
collectionName (*str*) – Collection

class atlasapi.specs.**DatabaseUsersUpdatePermissionsSpecs**(*password=None*)

Bases: *DatabaseUsersPermissionsSpecs*

Update Permissions spec for Database User

Constructor

Keyword Arguments
password (*str*) – Password for the username

getSpecs()

Get specs

Returns
Representation of the object

Return type
dict

class atlasapi.specs.**Host**(*data: dict*)

Bases: object

add_log_file(*name:* AtlasLogNames, *file: BinaryIO*) → None

Adds the passed log file to the hosts object

Parameters

- **name** (*AtlasLogNames*) – The type of logfile to be appended.
- **file** (*BinaryIO*) – The file to be appended

add_measurements(*measurement*) → None

data_partition_stats(*atlas_obj*, *granularity: Optional[*AtlasGranularities*] = None*, *period: Optional[*AtlasPeriods*] = None*) → Iterable[*AtlasMeasurement*]

Returns disk measurements for the data partition of the host.

Hard codes the name of the partition to *data* and returns all metrics.

Parameters

- **atlas_obj** (*atlasapi.atlas.Atlas*) – Instantiated Atlas instance to access the API
- **granularity** (*Optional[AtlasGranularitues]*) – The granularity for the disk measurements.
- **atlas_obj** – A configured Atlas instance to connect to the API with.

Returns (Iterable[AtlasMeasurement]): A generator yielding AtlasMeasurements

get_databases(*atlas_obj*) → Iterable[str]

Returns all disks(partitions) configured on the Atlas Host

Yields names of databases, and appends them to the databa

Parameters
atlas_obj –

Returns
A list of database names.

Return type
List[str]

get_measurement_for_host(*atlas_obj*, *granularity: Optional[AtlasGranularities] = None*, *period: Optional[AtlasPeriods] = None*, *measurement: Optional[AtlasMeasurementTypes] = None*, *iterable: bool = True*) → Union[dict, Iterable[*AtlasMeasurement*]]

Get measurement(s) for a host

Returns measurements for the Host object.

url: https://docs.atlas.mongodb.com/reference/api/process-measurements/

Accepts either a single measurement, but will retrieve more than one measurement if the measurement (using the AtlasMeasurementTypes class)

/api/atlas/v1.0/groups/{GROUP-ID}/processes/{HOST}:{PORT}/measurements

Keyword Arguments

- **host_obj** (Host) – the host
- **granularity** (AtlasGranularities) – the desired granularity
- **period** (AtlasPeriods) – The desired period
- **measurement** (AtlasMeasurementTypes) – The desired measurement or Measurement class
- **iterable** (*bool*) – To return an iterable high level object instead of a low level API response

Returns
Iterable object representing this function OR Response payload

Return type
Iterable[*AtlasMeasurement*] or dict

Raises:

get_measurements_for_database(*atlas_obj*, *database_name: str*, *granularity: Optional[AtlasGranularities] = None*, *period: Optional[AtlasPeriods] = None*, *iterable: bool = True*) → Iterable[Union[*AtlasMeasurement*, Any]]

Returns All Metrics for a database, for a given period and granularity.

Uses default granularity and period if not passed.

Parameters

- **iterable** (*bool*) – Defaults to true, if not true will return the raw response from API.
- **database_name** (*str*) – The database name (local should always exist, and can be used for testing)
- **period** (*Optional[AtlasPeriods]*) – The period for the disk measurements
- **granularity** (*Optional[AtlasGranularitues]*) – The granularity for the disk measurements.
- **atlas_obj** (atlasapi.atlas.Atlas) – A configured Atlas instance to connect to the API with.

Returns
Yields AtlasMeasirements or the original response.

Return type
Iterable[Union[*AtlasMeasurement*, Any]

get_measurements_for_disk(*atlas_obj*, *partition_name: str*, *granularity: Optional[AtlasGranularities] = None*, *period: Optional[AtlasPeriods] = None*, *iterable: bool = True*) → Iterable[Union[*AtlasMeasurement*, Any]]

Returns All Metrics for a Hosts partition, for a given period and granularity.

Uses default granularity and period if not passed.

Parameters

- **iterable** (*bool*) – Defaults to true, if not true will return the raw response from API.
- **partition_name** – The Atlas partition name (commonly *data*)
- **period** (*Optional[AtlasPeriods]*) – The period for the disk measurements
- **granularity** (*Optional[AtlasGranularitues]*) – The granularity for the disk measurements.
- **atlas_obj** (atlasapi.atlas.Atlas) – A configured Atlas instance to connect to the API with.

Returns
A list of partition names.

Return type
List[str]

get_partitions(*atlas_obj*) → Iterable[str]

Returns names of all disks(partitions) configured on the Atlas Host :param atlas_obj:

Returns
A list of partition names.

Return type
Iterable[str]

class atlasapi.specs.**HostLogFile**(*log_name: Optional[AtlasLogNames] = None*, *log_file_binary: Optional[BinaryIO] = None*)

Bases: object

class atlasapi.specs.**IAMType**(*value*)

Bases: Enum

An enumeration.

NONE = 'None'

ROLE = 'ROLE'

USER = 'USER'

class atlasapi.specs.**ReplicaSetTypes**(*value*)

Bases: Enum

An enumeration.

NO_DATA = 'No data available'

RECOVERING = 'Recovering'

REPLICA_PRIMARY = 'ReplicaSet primary'

REPLICA_SECONDARY = 'ReplicaSet secondary'

SHARD_CONFIG = 'Config server'

SHARD_CONFIG_PRIMARY = 'Config server'

SHARD_CONFIG_SECONDARY = 'Config server'

SHARD_MONGOS = 'Mongos router'

SHARD_PRIMARY = 'Shard primary'

SHARD_SECONDARY = 'Shard secondary'

SHARD_STANDALONE = 'Standalone'

class atlasapi.specs.RoleSpecs

Bases: object

Roles supported by Atlas

atlasAdmin = 'atlasAdmin'

backup = 'backup'

clusterMonitor = 'clusterMonitor'

dbAdmin = 'dbAdmin'

dbAdminAnyDatabase = 'dbAdminAnyDatabase'

enableSharding = 'enableSharding'

read = 'read'

readAnyDatabase = 'readAnyDatabase'

readWrite = 'readWrite'

readWriteAnyDatabase = 'readWriteAnyDatabase'

1.12 atlasapi.maintenance_window module

Maint Window Module

The maintenanceWindow resource provides access to retrieve or update the current Atlas project maintenance window. To learn more about Maintenance Windows, see the Set Preferred Cluster Maintenance Start Time setting on the View/Modify Project Settings page.

class atlasapi.maintenance_window.MaintenanceWindow(*day_of_week:* Weekdays = *Weekdays.SUNDAY*, *hour_of_day: int = 23*, *number_of_deferrals: int = 1*, *start_asap: bool = False*)

Bases: object

as_dict() → dict

Returns the Maintenance object as a serializable dict

Converts enums Returns:

as_update_dict() → dict

Returns a dict with immutable properties removed. Returns: dict

classmethod from_dict(*data_dict: dict*)

Creates a maint window definition from a dict. :param data_dict: An atlas formated dict

Returns:

class atlasapi.maintenance_window.**Weekdays**(*value*)

Bases: Enum

An enumeration.

FRIDAY = 6

MONDAY = 2

SATURDAY = 7

SUNDAY = 1

THURSDAY = 5

TUESDAY = 3

WEDNESDAY = 4

1.13 atlasapi.cloud_backup module

Cloud Backups Module

Provides access to Cloud Backups and Cloud backup restore endpoints

class atlasapi.cloud_backup.**CloudBackupRequest**(*cluster_name: str*, *retention_days: int = 1*, *description: str = 'Created by pyAtlasAPI'*)

Bases: object

property as_dict

class atlasapi.cloud_backup.**CloudBackupSnapshot**(*id: Optional[str] = None*, *cloud_provider: Optional[ProviderName] = None*, *created_at: Optional[datetime] = None*, *description: Optional[str] = None*, *expires_at: Optional[datetime] = None*, *links: Optional[List] = None*, *masterkey_uuid: Optional[str] = None*, *members: Optional[list] = None*, *mongod_version: Optional[str] = None*, *replica_set_name: Optional[str] = None*, *snapshot_ids: Optional[list] = None*, *snapshot_type: Optional[SnapshotType] = None*, *status: Optional[SnapshotStatus] = None*, *storage_size_bytes: Optional[int] = None*, *type: Optional[ClusterType] = None*)

Bases: object

classmethod from_dict(*data_dict: dict*)

class atlasapi.cloud_backup.**DeliveryType**(*value*)

Bases: Enum

An enumeration.

automated = 'Automated restore to Atlas cluster'

download = 'manual download of archived data directory'

pointInTime = 'Automated point in time restore to Atlas Cluster'

class atlasapi.cloud_backup.**SnapshotRestore**(*delivery_type: DeliveryType*, *snapshot_id: str*, *target_cluster_name: Optional[str] = None*, *target_group_id: Optional[str] = None*)

Bases: object

property as_dict: dict

class atlasapi.cloud_backup.**SnapshotRestoreResponse**(*restore_id: str*, *delivery_type: DeliveryType*, *snapshot_id: str*, *target_cluster_name: str*, *target_group_id: str*, *cancelled: bool = False*, *created_at: Optional[datetime] = None*, *expired: bool = False*, *expires_at: Optional[datetime] = None*, *finished_at: Optional[datetime] = None*, *links: Optional[list] = None*, *snapshot_timestamp: Optional[datetime] = None*, *target_deployment_item_name: Optional[str] = None*, *delivery_url: Optional[str] = None*)

Bases: *SnapshotRestore*

classmethod from_dict(*data_dict*)

class atlasapi.cloud_backup.**SnapshotStatus**(*value*)

Bases: Enum

An enumeration.

COMPLETED = 'Completed'

FAILED = 'Failed'

INPROGRESS = 'In Progress'

QUEUED = 'Queued'

class atlasapi.cloud_backup.**SnapshotType**(*value*)

Bases: Enum

An enumeration.

FALLBACK = 'Fallback'

ONDEMAND = 'On Demand'

SCHEDULED = 'Scheduled'

atlasapi.cloud_backup.**try_bool**(*str_in: str*) → bool

atlasapi.cloud_backup.**try_date**(*str_in: str*) → Optional[datetime]

1.14 atlasapi.lib module

class atlasapi.lib.**AtlasGranularities**

Bases: object

Helper class to create ISO 8601 durations to pass to the API

To add more possible granularities, add them here.

DAY = 'P1D'

FIVE_MINUTE = 'PT5M'

HOUR = 'PT1H'

MINUTE = 'PT1M'

TEN_SECOND = 'PT10S'

class atlasapi.lib.**AtlasLogNames**(*value*)

Bases: Enum

The name of the log file that you want to retrieve:

MONGODB = 'mongodb.gz'

MONGOD_AUDIT = 'mongodb-audit-log.gz'

MONGOS = 'mongos.gz'

MONGOS_AUDIT = 'mongos-audit-log.gz'

class atlasapi.lib.**AtlasPeriods**

Bases: object

Helper class to create ISO 8601 durations to send to the Atlas period parameter.

To add more periods, add them here.

HOURS_1 = 'PT1H'

HOURS_24 = 'P1D'

HOURS_48 = 'P2D'

HOURS_8 = 'PT8H'

MINUTES_15 = 'PT15M'

MONTHS_1 = 'P1M'

MONTHS_2 = 'P2M'

WEEKS_1 = 'P7D'

WEEKS_4 = 'P28D'

YEARS_1 = 'P1Y'

YEARS_2 = 'P2Y'

class atlasapi.lib.**AtlasUnits**(*value*)

Bases: Enum

An enumeration.

BYTES = 'BYTES'

BYTES_PER_SECOND = 'BYTES_PER_SECOND'

GIGABYTES = 'GIGABYTES'

GIGABYTES_PER_HOUR = 'GIGABYTES_PER_HOUR'

MEGABYTES_PER_SECOND = 'MEGABYTES_PER_SECOND'

MILLISECONDS = 'MILLISECONDS'

PERCENT = 'PERCENT'

SCALAR = 'SCALAR'

SCALAR_PER_SECOND = 'SCALAR_PER_SECOND'

class atlasapi.lib.**ClusterType**(*value*)

Bases: Enum

The types of clusteres available in Atlas.

GEOSHARDED is a Global write cluster sharded by geo information.

GEOSHARDED = 'Global Cluster'

REPLICASET = 'Replica Set'

SHARDED = 'Sharded Cluster'

SHARDEDCLUSTER = 'Sharded Cluster'

class atlasapi.lib.**LogLine**(*raw_line*)

Bases: object

class atlasapi.lib.**MongoDBMajorVersion**(*value*)

Bases: Enum

An enumeration.

v3_4 = '3.4'

v3_6 = '3.6'

v4_0 = '4.0'

v4_2 = '4.2'

v4_4 = '4.4'

v5_0 = '5.0'

vX_x = 'Unknown'

class atlasapi.lib.**ProviderName**(*value*)

Bases: Enum

An enumeration.

AWS = 'Amazon Web Services'

AZURE = 'Microsoft Azure'

GCP = 'Google Cloud Platform'

TENANT = 'Shared Tier'

1.15 atlasapi.projects module

class atlasapi.projects.**Project**(*name: str*, *org_id: str*, *created_date: Optional[datetime] = None*, *cluster_count: Optional[int] = None*, *id: Optional[str] = None*, *links: Optional[list] = None*, *with_default_alert_settings: Optional[bool] = True*, *project_owner_id: Optional[str] = None*)

Bases: object

property create_dict: dict

A dictionary in the format Atlas API "create project expects"

Returns: A dictionary in the format Atlas API "create project expects"

classmethod for_create(*name: str*, *org_id: str*, *with_default_alert_settings: bool = True*, *project_owner_id: Optional[str] = None*)

Creates a new Project object for use in creating a new project.

Only name and org_id are required.

Parameters

- **project_owner_id** (*str*) – Unique 24-hexadecimal digit string that identifies the Atlas user account to be granted the Project Owner role on the specified project. If you set this parameter, it overrides the default value of the oldest Organization Owner.
- **name** (*str*) – The name of the project. You can use this value for populating the {GROUP-NAME} parameter of the /groups/byName/{GROUP-NAME} endpoint.
- **org_id** (*str*) – The unique identifier of the Atlas organization to which the project belongs.
- **with_default_alert_settings** (*bool*) – Flag that indicates whether to create the new project with the default alert settings enabled. This parameter defaults to true.

Returns: None

classmethod from_dict(*data_dict*)

Creates a Project object from a passed dict, in the format of the Atlas API.

Parameters
data_dict (*dict*) – A dictionary in the format of the Atlas API.

Returns: None

class atlasapi.projects.**ProjectSettings**(*is_collect_db_stats: Optional[bool] = None*, *is_data_explorer: Optional[bool] = None*, *is_performance_advisor: Optional[bool] = None*, *is_realtime_perf: Optional[bool] = None*, *is_schema_advisor: Optional[bool] = None*)

Bases: object

classmethod from_dict(*data_dict: dict*)

1.16 atlasapi.organizations module

class atlasapi.organizations.**Organization**(*name: str*, *is_deleted: bool = False*, *links: Optional[list] = None*, *id: Optional[str] = None*)

Bases: object

classmethod from_dict(*data_dict: dict*)

CHAPTER

TWO

NESTED CLASS FOR ATLASAPI.ATLAS::ATLAS

2.1 Atlas._Clusters

class Atlas.**_Clusters**(*atlas*)

Bases: object

Clusters API

see: https://docs.atlas.mongodb.com/reference/api/clusters/

Constructor

Parameters
atlas (Atlas) – Atlas instance

create_basic_rs(*name: str*, *size:* InstanceSizeName = *InstanceSizeName.M10*, *disk_size: int = 10*, *provider:* ProviderName = *ProviderName.AWS*, *region: str = 'US_WEST_2'*, *version:* MongoDBMajorVersion = *MongoDBMajorVersion.v4_0*) → *AtlasBasicReplicaSet*

Simplified method for creating a basic replica set with basic options.

Return type
AtlasBasicReplicaSet

Parameters

- **name** (str) – The name for the cluster
- **size** (*InstanceSizeName*) – The Atlas Instance size, found in The InstanceSizeName enum
- **disk_size** (int) – The size in GB for disk
- **provider** (*ProviderName*) – The cloud provider, found in ProviderName enum
- **region** (str) – The provider region to place the cluster.
- **version** (*MongoDBMajorVersion*) – The mongodb major version (enum)

Returns
AtlasBasicReplicaSet

create_cluster(*cluster:* ClusterConfig) → dict

Create a cluster

url: POST /api/atlas/v1.0/groups/{GROUP-ID}/clusters

Parameters
cluster (ClusterConfig) – A Cluster Config Object

Returns
Response payload

Return type
dict

`delete_cluster`(*cluster: str*, *areYouSure: bool = False*)

Delete a Cluster

url: https://docs.atlas.mongodb.com/reference/api/clusters-delete-one/

Parameters
cluster (str :param cluster: Cluster name :param areYouSure: safe flag to don't delete a cluster by mistake) – Cluster name

Keyword Arguments
areYouSure (*bool*) – safe flag to don't delete a cluster by mistake

Returns
Response payload

Return type
dict

Raises
ErrConfirmationRequested – Need a confirmation to delete the cluster

Raises
type areYouSure: bool

`get_all_clusters`(*pageNum=1*, *itemsPerPage=500*, *iterable=False*)

Get All Clusters

url: https://docs.atlas.mongodb.com/reference/api/clusters-get-all/

Keyword Arguments

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Users per Page
- **iterable** (*bool*) – To return an iterable high level object instead of a low level API response

Returns
Iterable object representing this function OR Response payload

Return type
AtlasPagination or dict

Raises
ErrPaginationLimits – Out of limits

`get_single_cluster`(*cluster: str*) → dict

Get a Single Cluster

url: https://docs.atlas.mongodb.com/reference/api/clusters-get-one/

Parameters
cluster (*str*) – The cluster name

Returns
Response payload

Return type
dict

`get_single_cluster_advanced_options`(*cluster: str*, *as_obj: bool = True*) → Union[dict, *AdvancedOptions*]

Retrieves advanced options from a cluster, either as a obj, or optionally as a dict.

GET /groups/{GROUP-ID}/clusters/{CLUSTER-NAME}/processArgs

Parameters

- **cluster** (str) –
- **as_obj** (bool) – True to return, AdvancedOptions, false for a dict

Returns
AdvancedOptions object or dict

`get_single_cluster_as_obj`(*cluster*) → Union[*ClusterConfig*, *ShardedClusterConfig*]

Get a Single Cluster as data

url: https://docs.atlas.mongodb.com/reference/api/clusters-get-one/

Parameters
cluster (*str*) – The cluster name

Returns
Response payload

Return type
ClusterConfig

`is_existing_cluster`(*cluster*) → bool

Check if the cluster exists

Not part of Atlas api but provided to simplify some code

Parameters
cluster (*str*) – The cluster name

Returns
The cluster exists or not

Return type
bool

`modify_cluster`(*cluster: str*, *cluster_config: Union[*ClusterConfig*, dict]*) → dict

Modify a Cluster

Modifies an existing cluster in the project. Either from a full ClusterConfig object, or from a simple dict which contains the elements desired.

url: https://docs.atlas.mongodb.com/reference/api/clusters-modify-one/

Return type
dict

Parameters

- **cluster** (str) – The name of the cluster to modify
- **cluster_config** – A ClusterConfig object containing the new configuration, or a dict containing fragment.

Returns
dict: A dictionary of the new cluster config

modify_cluster_advanced_options(*cluster: str*, *advanced_options:* AdvancedOptions, *as_obj: bool = True*) → Union[*AdvancedOptions*, dict]

Modifies cluster advanced options using a AdvancedOptions object.

PATCH /groups/{GROUP-ID}/clusters/{CLUSTER-NAME}/processArgs

Parameters

- **cluster** (str) – The clutster name
- **advanced_options** (*AdvancedOptions*) – An AdvancedOptions object with the options to be set.
- **as_obj** (bool) – Return the new AdvancedOptions as an object.

Returns

modify_cluster_instance_size(*cluster: str*, *new_cluster_size:* InstanceSizeName) → dict

Modifies existing cluster by changing only the instance size.

Helper function using modify_cluster :rtype: dict :type new_cluster_size: *InstanceSizeName* :type cluster: str :param cluster: The cluster name :param new_cluster_size: InstanceSizeName: The new size to use. :return: dict: the new cluster configuration dict

modify_cluster_tls(*cluster: str*, *TLS_protocol:* TLSProtocols, *as_obj: bool = True*) → *TLSProtocols*

Modifies cluster TLS settings.

pause_cluster(*cluster: str*, *toggle_if_paused: bool = False*) → dict

Pauses/Unpauses a cluster.

If you wish to unpause, set the toggle_if_paused param to True. :rtype: dict :type toggle_if_paused: bool :type cluster: str :param cluster: The name of the cluster :param toggle_if_paused: Set to true to unpause a paused clsuter. :return: dict: The updated config

test_failover(*cluster: str*) → Optional[dict]

Triggers a primary failover for a cluster

Used for testing cluster resiliency.

Return type
dict

Parameters
cluster (str) –

Returns
And empty dict

2.2 Atlas._DatabaseUsers

class Atlas.**_DatabaseUsers**(*atlas*)

Bases: object

Database Users API

see: https://docs.atlas.mongodb.com/reference/api/database-users/

Constructor

Parameters
atlas (Atlas) – Atlas instance

create_a_database_user(*permissions*) → dict

Create a Database User

url: https://docs.atlas.mongodb.com/reference/api/database-users-create-a-user/

Parameters
permissions (*DatabaseUsersPermissionsSpec*) – Permissions to apply

Returns
Response payload

Return type
dict

delete_a_database_user(*user: str*) → dict

Delete a Database User

url: https://docs.atlas.mongodb.com/reference/api/database-users-delete-a-user/

Parameters
user (*str*) – User to delete

Returns
Response payload

Return type
dict

get_a_single_database_user(*user: str*) → dict

Get a Database User

url: https://docs.atlas.mongodb.com/reference/api/database-users-get-single-user/

Parameters
user (*str*) – User

Returns
Response payload

Return type
dict

get_all_database_users(*pageNum: int = 1*, *itemsPerPage: int = 500*, *iterable: bool = False*)

Get All Database Users

url: https://docs.atlas.mongodb.com/reference/api/database-users-get-all-users/

Keyword Arguments

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Users per Page
- **iterable** (*bool*) – To return an iterable high level object instead of a low level API response

Returns
Iterable object representing this function OR Response payload

Return type
AtlasPagination or dict

Raises
ErrPaginationLimits – Out of limits

update_a_database_user(*user: str*, *permissions:* DatabaseUsersUpdatePermissionsSpecs) → dict

Update a Database User

url: https://docs.atlas.mongodb.com/reference/api/database-users-update-a-user/

Parameters

- **user** (*str*) – User
- **permissions** (DatabaseUsersUpdatePermissionsSpecs) – Permissions to apply

Returns
Response payload

Return type
dict

2.3 Atlas._Alerts

class Atlas.**_Alerts**(*atlas*)

Bases: object

Alerts API

see: https://docs.atlas.mongodb.com/reference/api/alerts/

Constructor

Parameters
atlas (Atlas) – Atlas instance

acknowledge_an_alert(*alert*, *until*, *comment=None*)

Acknowledge an Alert

url: https://docs.atlas.mongodb.com/reference/api/alerts-acknowledge-alert/

Parameters

- **alert** (*str*) – The alert id
- **until** (*datetime*) – Acknowledge until

Keyword Arguments
comment (*str*) – The acknowledge comment

Returns
Response payload :param comment:

Return type
dict

`acknowledge_an_alert_forever`(*alert*, *comment=None*)

Acknowledge an Alert forever

url: https://docs.atlas.mongodb.com/reference/api/alerts-acknowledge-alert/

Parameters
alert (*str*) – The alert id

Keyword Arguments
comment (*str*) – The acknowledge comment

Returns
Response payload

Return type
dict

`get_all_alerts`(*status=None*, *pageNum=1*, *itemsPerPage=500*, *iterable=False*)

Get All Alerts

url: https://docs.atlas.mongodb.com/reference/api/alerts-get-all-alerts/

Keyword Arguments

- **status** (AlertStatusSpec) – filter on alerts status
- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Users per Page
- **iterable** (*bool*) – To return an iterable high level object instead of a low level API response

Returns
Iterable object representing this function OR Response payload

Return type
AtlasPagination or dict

Raises
ErrPaginationLimits – Out of limits

`get_an_alert`(*alert: str*) → *Alert*

Get an Alert

url: https://docs.atlas.mongodb.com/reference/api/alerts-get-alert/

Parameters
alert (*str*) – The alert id

Returns
Response payload

Return type
dict

`unacknowledge_an_alert`(*alert*)

Acknowledge an Alert

url: https://docs.atlas.mongodb.com/reference/api/alerts-acknowledge-alert/

Parameters
alert (*str*) – The alert id

Returns
Response payload

Return type
dict

2.4 Atlas._MaintenanceWindows

class Atlas.**_MaintenanceWindows**(*atlas*)

Bases: object

Maintenance Windows API

see: https://docs.atlas.mongodb.com/reference/api/maintenance-windows/

The maintenanceWindow resource provides access to retrieve or update the current Atlas project maintenance
window. To learn more about Maintenance Windows, see the Set Preferred Cluster Maintenance Start Time setting on the View/Modify Project Settings page.

Parameters
atlas (Atlas) – Atlas instance

current_config() → *MaintenanceWindow*

The current Maintainable Window configuration.

Returns: MaintainableWindow object

defer() → dict

Defers the currently scheduled maintenance window.

Returns: bool:

set_config(*new_config:* MaintenanceWindow) → bool

Sets the maint configuration to the values in the passed MaintWindow Object

Will only set those values which are not none in the MaintWindow Object. Currently you can not use this method to set a value as null. (This is not supported by the API anyway)

Parameters
new_config – A MaintainenceWindow Object

Returns: bool: True is success

2.5 Atlas._Hosts

class Atlas.**_Hosts**(*atlas*)

Bases: object

Hosts API

see: https://docs.atlas.mongodb.com/reference/api/monitoring-and-logs/#monitoring-and-logs

Constructor

Parameters
atlas (Atlas) – Atlas instance

property cluster_list: Set[str]

Returns a list of clusters found in the hosts for this group.

Returns
A set of cluster names

Return type
Set[str}

fill_host_list(*for_cluster: Optional[str] = None*) → List[*Host*]

Fills the *self.hostname* property with the current hosts for the project/group.

Optionally, one can specify the *for_cluster* parameter to fill the host list with hosts only from the specified cluster.

Parameters
for_cluster (*str*) – The name of the cluster for filter the host list.

Returns
A lost of *Host* objects

Return type
List[*Host*]

get_log_for_host(*host_obj: Host*, *log_name: AtlasLogNames = AtlasLogNames.MONGODB*, *date_from: Optional[datetime] = None*, *date_to: Optional[datetime] = None*) → BinaryIO

Retrieves the designated logfile archive of designated log_name and for the designated dates, and returns a binary file like object.

Parameters

- **host_obj** (Host) – And atlas Host object to retrieve logs for
- **log_name** (AtlasLogNames) – an AtlasLogNames type
- **date_from** (*datetime.datetime*) – The datetime to start from
- **date_to** (*datetime.datetime*) – The datetime to gather till

Returns
A BinaryIO object containing the gzipped log file.

Return type
BinaryIO

get_loglines_for_host(*host_obj: Host*, *log_name: AtlasLogNames = AtlasLogNames.MONGODB*, *date_from: Optional[datetime] = None*, *date_to: Optional[datetime] = None*) → Iterable[*LogLine*]

Gathers the designated log file from Atlas, and then returns the lines therein contained.

Does so by downloading the gzip file into memory, ungzipping and then unpacking each log line as a LogLine Object.

Parameters

- **host_obj** (Host) – An atlas Host object to retrive logs for
- **log_name** (*str*) – an AtlasLogNames type
- **date_from** (*datetime*) – The datetime to start from

- **date_to** (*datetime*) – The datetime to gather till

Returns
Yeilds LogLine objects, one for each logline found in the file.

Return type
Iterable[*LogLine*]

get_logs_for_cluster(*cluster_name: str*, *log_name:* AtlasLogNames = *AtlasLogNames.MONGODB*, *date_from: Optional[datetime] = None*, *date_to: Optional[datetime] = None*) → Iterable[*Host*]

Yields A Host object per Host in the passed cluster with a File-like objects containing the gzipped log file requested for each host in the project using the same date filters and log_name (type) in the log_files property.

Currently the log_file property (List) is usually with only one item. :param log_name: The type of log to be retrieved :type log_name: AtlasLogNames :param date_from: Start of log entries :type date_from: datetime :param date_to: End of log entries :type date_to: datetime

Returns
Yields Host objects, with full host information as well as the logfile in the log_files property.

Return type
Iterable[*Host*]

get_logs_for_project(*log_name:* AtlasLogNames = *AtlasLogNames.MONGODB*, *date_from: Optional[datetime] = None*, *date_to: Optional[datetime] = None*) → Iterable[*Host*]

Yields A Host object per Host in the project with a File-like objects containing the gzipped log file requested for each host in the project using the same date filters and log_name (type) in the log_files property.

Currently the log_file property (List) is usually with only one item. :param log_name: The type of log to be retrieved :type log_name: AtlasLogNames :param date_from: Start of log entries :type date_from: datetime :param date_to: End of log entries :type date_to: datetime

Returns
Yields Host objects, with full host information as well as the logfile in the log_files property.

Return type
Iterable[*Host*]

get_measurement_for_hosts(*granularity: Optional[*AtlasGranularities*] = None*, *period: Optional[*AtlasPeriods*] = None*, *measurement: Optional[*AtlasMeasurementTypes*] = None*, *return_data: bool = False*)

Get measurement(s) for all hosts in the host_list

Populates all hosts in the host_list with the requested metric.

Multiple calls will append additional metrics to the same host object.

Please note that using the *return_data* param will also return the updated host objects, which may unnecessarily consume memory.

Keyword Arguments

- **granularity** (AtlasGranularities) – the desired granularity
- **period** (AtlasPeriods) – The desired period
- **measurement** (AtlasMeasurementTypes) – The desired measurement or Measurement class

host_list_by_cluster(*cluster_name: str*) → Iterable[*Host*]

Returns hosts belonging to the named cluster. :param cluster_name: :type cluster_name: str

Returns
An interator of Host Objects.

Return type
Iterable[*Host*]

property host_list_primaries: Iterable[*Host*]

Yields only hosts which are currently primary.

property host_list_secondaries: Iterable[*Host*]

Yields only hosts which are currently secondaries.

property host_names: Iterable[str]

Returns a simple list of host names without port

Return type
Iterator[str]

update_host_list(*host_obj:* Host) → None

Places a host into the host_list property.

Parameters
host_obj – Host: A host object with measurements.

Return type
None

2.6 Atlas._Events

class Atlas.**_Events**(*atlas*)

Bases: object

Events API

see: https://docs.atlas.mongodb.com/reference/api/events/

Constructor

Parameters
atlas (Atlas) – Atlas instance

property all: ListOfEvents

Returns all events for the current project/group.

Returns
A list of event objects.

Return type
ListOfEvents

all_by_type(*event_type: AtlasEventTypes*) → Iterable[*AtlasEvent*]

Returns all events for the passed AtlasEventType

Parameters
event_type (*AtlasEventTypes*) –

Returns
Iterable[AtlasEvent]

since(*since_datetime: datetime*) → ListOfEvents

Returns all events since the passed datetime. (UTC)

Return type
ListOfEvents

since_by_type(*since_datetime: datetime*, *event_type: AtlasEventTypes*)

Returns all events since the passed detetime (UTC) for the passed AtlasEvent Type

Parameters

- **since_datetime** (*datetime*) –
- **event_type** (*AtlasEventTypes*) –

Returns:

2.7 Atlas._Whitelist

class Atlas.**_Whitelist**(*atlas*)

Bases: object

Whitelist API

see: https://docs.atlas.mongodb.com/reference/api/whitelist/

Constructor

Parameters
atlas (Atlas) – Atlas instance

create_whitelist_entry(*ip_address: str*, *comment: str*) → List[*WhitelistEntry*]

Create a whitelist entry

url: https://docs.atlas.mongodb.com/reference/api/whitelist-add-one/

Parameters

- **ip_address** (*str*) – ip address to add to whitelist
- **comment** (*str*) – comment describing the whitelist entry

Returns
Response payload

Return type
List[*WhitelistEntry*]

delete_a_whitelist_entry(*ip_address: str*) → dict

Delete a whitelist entry

url: https://docs.atlas.mongodb.com/reference/api/whitelist-delete-one/

Parameters
ip_address (*str*) – ip address to delete from whitelist

Returns
Response payload

Return type
dict

get_all_whitelist_entries(*pageNum: int = 1*, *itemsPerPage: int = 500*, *iterable: bool = False*) → Iterable[*WhitelistEntry*]

Get All whitelist entries

url: https://docs.atlas.mongodb.com/reference/api/whitelist-get-all/

Keyword Arguments

- **pageNum** (*int*) – Page number
- **itemsPerPage** (*int*) – Number of Users per Page
- **iterable** (*bool*) – To return an iterable high level object instead of a low level API response

Returns
Iterable object representing this function OR Response payload

Return type
AtlasPagination or dict

Raises
ErrPaginationLimits – Out of limits

get_whitelist_entry(*ip_address: str*) → *WhitelistEntry*

Get a whitelist entry

url: https://docs.atlas.mongodb.com/reference/api/whitelist-get-one-entry/

Parameters
ip_address (*str*) – ip address to fetch from whitelist

Returns
Response payload

Return type
WhitelistEntry

2.8 Atlas._CloudBackups

class Atlas.**_CloudBackups**(*atlas*)

Bases: object

Cloud Backup API

see: https://docs.atlas.mongodb.com/reference/api/cloud-backup/backup/backups/

The CloudBackups resource provides access to retrieve the Cloud provider backup snapshots.

Parameters
atlas (Atlas) – Atlas instance

cancel_snapshot_restore_request(*cluster_name: str*, *restore_id: str*)

Cancels a current backup restore request by restore_id.

Calls: https://docs.atlas.mongodb.com/reference/api/cloud-backup/restore/delete-one-restore-job/

Parameters

- **cluster_name** – The name of the source cluster.
- **restore_id** – The id of the (jobId) of the restore job.

create_snapshot_for_cluster(*cluster_name: str*, *retention_days: int = 7*, *description: Optional[str] = None*, *as_obj: bool = True*) → Union[*CloudBackupSnapshot*, dict]

Creates and on demand snapshot for the passed cluster

Parameters

- **as_obj** –
- **cluster_name** –
- **retention_days** –
- **description** –

get_backup_snapshot_for_cluster(*cluster_name: str*, *snapshot_id: str*, *as_obj: bool = True*) → Union[Iterable[*CloudBackupSnapshot*], Iterable[dict]]

Get singe backup snapshot for a cluster.

Retrieves url: https://docs.atlas.mongodb.com/reference/api/cloud-backup/backup/backups/

Keyword Arguments
cluster_name (*str*) – The cluster name to fetch

Returns
Iterable object representing this function OR Response payload

Return type
AtlasPagination or dict

get_backup_snapshots_for_cluster(*cluster_name: str*, *as_obj: bool = True*) → Union[Iterable[*CloudBackupSnapshot*], Iterable[dict]]

Get backup snapshots for a cluster.

Retrieves url: https://docs.atlas.mongodb.com/reference/api/cloud-backup/backup/backups/

Keyword Arguments
cluster_name (*str*) – The cluster name to fetch

Returns
Iterable object representing this function OR Response payload

Return type
AtlasPagination or dict

get_snapshot_restore_requests(*cluster_name: str*, *restore_id: Optional[str] = None*, *as_obj: bool = True*) → Union[List[Union[dict, *SnapshotRestoreResponse*]], *SnapshotRestoreResponse*, dict]

is_existing_snapshot(*cluster_name: str*, *snapshot_id: str*) → bool

request_snapshot_restore(*source_cluster_name: str*, *snapshot_id: str*, *target_cluster_name: str*, *delivery_type:* DeliveryType *= DeliveryType.automated*, *allow_same: bool = False*) → *SnapshotRestoreResponse*

request_snapshot_restore_to_group(*source_cluster_name: str*, *snapshot_id: str*, *target_cluster_name: str*, *target_group_obj*, *delivery_type:* DeliveryType *= DeliveryType.automated*) → *SnapshotRestoreResponse*

Requests a snapshot restore to another group/project.

Uses the passed target_group_obj, which is an Atlas object, to restore a snapshot from one group/project to another.

This method does not check if the source and destination clusters have the same name, since this would not be dangerous when these are in two groups.

Parameters

- **source_cluster_name** – the text name of the source cluster
- **snapshot_id** – the uuid id of the snapshot to be restored
- **target_cluster_name** – the txt name of the destination cluster
- **target_group_obj** – Atlas: An Atlas object connected to the destination group.
- **delivery_type** – DeliveryType: IF you want to download, or automatically restore on Atlas.

Returns:

2.9 Atlas._Projects

class Atlas.**_Projects**(*atlas*)

Bases: object

Atlas Projects

see: https://www.mongodb.com/docs/atlas/reference/api/projects/

The groups resource provides access to retrieve or create Atlas projects.

Parameters
atlas (Atlas) – Atlas instance

get_project_teams(*group_id: Optional[str] = None*) → Iterable[TeamRoles]

Retrieves all teams assigned to the passed project/group

Returns each team assigned to the project, along with the roles which are assigned.

Returns (Iterable[TeamRoles]): Yields TeamRole Objects.

get_project_users(*group_id: Optional[str] = None*, *flatten_teams: Optional[bool] = None*, *include_org_users: Optional[bool] = None*) → Iterable[AtlasUser]

Yields all users (AtlasUser objects) associated with the group_id.

Parameters

- **group_id** (*str*) – The group id to search, will use the configured group for the Atlas instance if instantiated in this way.
- **flatten_teams** (*bool*) – Flag that indicates whether the returned list should include users who belong to a team that is assigned a role in this project. You might not have assigned the individual users a role in this project.
- **include_org_users** (*bool*) – Flag that indicates whether the returned list should include users with implicit access to the project through the Organization Owner or Organization Read Only role. You might not have assigned the individual users a role in this project.

Returns (Iterable[AtlasUser]: An iterable of AtlasUser objects.

project_by_id(*project_id: str*) → *Project*

Return project by name

Parameters
project_id (*str*) – The project id (group_id) to return

Returns (Project): A single Project

project_by_name(*project_name: str*) → *Project*

Return project by name

Parameters
project_name (*str*) – The project name to return

Returns (Project): A single Project

property projects: Iterable[*Project*]

All Projects accessible by the current authed user/key Gets all projects for which the authed key has access.

Returns (Iterable[Project]): Yields Project Objects.

property settings: *ProjectSettings*

user_count(*group_id: Optional[str] = None*, *flatten_teams: Optional[bool] = None*, *include_org_users: Optional[bool] = None*) → int

Returns count of users added to this project

Parameters

- **group_id** (*str*) – The group id to search, will use the configured group for the Atlas instance if
- **way.** (*instantiated in this*) –
- **flatten_teams** (*bool*) – Flag that indicates whether the returned list should include users who belong to a team that is assigned a role in this project. You might not have assigned the individual users a role in this project.
- **include_org_users** (*bool*) – Flag that indicates whether the returned list should include users with implicit access to the project through the Organization Owner or Organization Read Only role. You

might not have assigned the individual users a role in this project.

Returns (int): Count of users.

2.10 Atlas._Organizations

class Atlas.**_Organizations**(*atlas*)

Bases: object

Atlas Organizations

see: https://www.mongodb.com/docs/atlas/reference/api/organizations/

TThe orgs resource provides access to manage Atlas organizations.

Parameters
atlas (Atlas) – Atlas instance

property count: int

Count of Organizations accessible by the authed user/key.

Returns (int):

get_all_projects_for_org(*org_id: str*) → Iterable[*Project*]

Get projects related to the current Org

url: https://www.mongodb.com/docs/atlas/reference/api/organization-get-all-projects/

Parameters
org_id (*str*) – The organization id which owns the projects.

Returns
Iterable containing projects

Return type
Iterable[*Project*]

organization_by_id(*org_id: str*) → *Organization*

Single organization searched by org_id.

Parameters
org_id (*str*) –

Returns (Organization): a single Organization object.

organization_by_name(*org_name: str*) → *Organization*

Single organization searched by name.

Parameters
org_name – Organization name with which to filter the returned list. Performs a case-insensitive search for organizations which exactly match the specified name.

Returns (Organization): a single Organization object.

property organizations: Iterable[*Organization*]

All Organizations accessible by the current authed user/key Gets all Organizations for which the authed key has access.

Returns (Iterable[Organization]): Yields Organization Objects.

CHAPTER

THREE

ATLASCLI - A COMMAND LINE PROGRAM FOR MONGODB ATLAS

The command line help for atlascli.py:

```
$ python atlascli/cli.py -h
usage: atlascli [-h] [--publickey PUBLICKEY] [--privatekey PRIVATEKEY]
            [--atlasgroup ATLASGROUP] [--format {short,full}]
            [--resource {organization,project,cluster}] [--id ID]
            [--debug] [--list]

A command line interface too the MongoDB Atlasdatabase as a
service.https://www.mongodb.com/cloud/atlas for more infoSee also
https://docs.atlas.mongodb.com/configure-api-access/#programmatic-api-keysFor
how to obtain a programmatic API key required to access the API

optional arguments:
-h, --help            show this help message and exit
--publickey PUBLICKEY
                      MongoDB Atlas public API key
--privatekey PRIVATEKEY
                      MongoDB Atlas private API key
--atlasgroup ATLASGROUP
                      Default group (aka project)
--format {short,full}
                      Format for output of list command [default: short]
--resource {organization,project,cluster}
                      Which resource type are we operating on:organization,
                      project or cluster? [default: cluster]
--id ID               Specify a resource id
--debug               Turn on logging at debug level [default: False]
--list                List a set of resources [default: False]
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

a

atlasapi.alerts, 5
atlasapi.atlas, 3
atlasapi.atlas_types, 5
atlasapi.cloud_backup, 33
atlasapi.clusters, 5
atlasapi.errors, 18
atlasapi.events, 10
atlasapi.lib, 35
atlasapi.maintenance_window, 32
atlasapi.measurements, 11
atlasapi.network, 22
atlasapi.organizations, 38
atlasapi.projects, 37
atlasapi.settings, 24
atlasapi.specs, 27
atlasapi.whitelist, 18

INDEX

A

ACCEPTED (*atlasapi.settings.Settings attribute*), 24
acknowledge_an_alert() (*atlasapi.atlas.Atlas._Alerts method*), 44
acknowledge_an_alert_forever() (*atlasapi.atlas.Atlas._Alerts method*), 45
add_log_file() (*atlasapi.specs.Host method*), 29
add_measurements() (*atlasapi.specs.Host method*), 29
add_role() (*atlasapi.specs.DatabaseUsersPermissionsSpecs method*), 27
add_roles() (*atlasapi.specs.DatabaseUsersPermissionsSpecs method*), 28
AdvancedOptions (*class in atlasapi.clusters*), 5
Alert (*class in atlasapi.alerts*), 5
AlertsGetAll (*class in atlasapi.atlas*), 3
AlertStatusSpec (*class in atlasapi.specs*), 27
all (*atlasapi.atlas.Atlas._Events property*), 49
all_by_type() (*atlasapi.atlas.Atlas._Events method*), 49
answer() (*atlasapi.network.Network method*), 22
api_resources (*atlasapi.settings.Settings attribute*), 25
as_create_dict() (*atlasapi.clusters.ClusterConfig method*), 7
as_create_dict() (*atlasapi.clusters.ReplicationSpecs method*), 9
as_dict (*atlasapi.cloud_backup.CloudBackupRequest property*), 33
as_dict (*atlasapi.cloud_backup.SnapshotRestore property*), 34
as_dict (*atlasapi.clusters.AdvancedOptions property*), 6
as_dict (*atlasapi.measurements.AtlasMeasurement property*), 11
as_dict() (*atlasapi.clusters.AtlasBasicReplicaSet method*), 6
as_dict() (*atlasapi.clusters.ClusterConfig method*), 7
as_dict() (*atlasapi.clusters.ProviderSettings method*), 9
as_dict() (*atlasapi.clusters.ReplicationSpecs method*), 9
as_dict() (*atlasapi.clusters.ShardedClusterConfig method*), 10
as_dict() (*atlasapi.maintenance_window.MaintenanceWindow method*), 32
as_dict() (*atlasapi.measurements.AtlasMeasurementValue method*), 17
as_dict() (*atlasapi.whitelist.WhitelistEntry method*), 18
as_modify_dict() (*atlasapi.clusters.ClusterConfig method*), 7
as_tuple (*atlasapi.measurements.AtlasMeasurementValue property*), 17
as_update_dict() (*atlasapi.maintenance_window.MaintenanceWindow method*), 32
Atlas (*class in atlasapi.atlas*), 3
Atlas._Alerts (*class in atlasapi.atlas*), 44
Atlas._CloudBackups (*class in atlasapi.atlas*), 51
Atlas._Clusters (*class in atlasapi.atlas*), 39
Atlas._DatabaseUsers (*class in atlasapi.atlas*), 43
Atlas._Events (*class in atlasapi.atlas*), 49
Atlas._Hosts (*class in atlasapi.atlas*), 46
Atlas._MaintenanceWindows (*class in atlasapi.atlas*), 46
Atlas._Organizations (*class in atlasapi.atlas*), 54
Atlas._Projects (*class in atlasapi.atlas*), 53
Atlas._Whitelist (*class in atlasapi.atlas*), 50
atlas_event_factory() (*in module atlasapi.events*), 11
atlasAdmin (*atlasapi.specs.RoleSpecs attribute*), 32
atlasapi.alerts
    module, 5
atlasapi.atlas
    module, 3
atlasapi.atlas_types
    module, 5
atlasapi.cloud_backup
    module, 33
atlasapi.clusters
    module, 5
atlasapi.errors
    module, 18
atlasapi.events
    module, 10
atlasapi.lib

module, 35
atlasapi.maintenance_window
module, 32
atlasapi.measurements
module, 11
atlasapi.network
module, 22
atlasapi.organizations
module, 38
atlasapi.projects
module, 37
atlasapi.settings
module, 24
atlasapi.specs
module, 27
atlasapi.whitelist
module, 18
AtlasBasicReplicaSet (*class in atlasapi.clusters*), 6
AtlasClusterEvent (*class in atlasapi.events*), 10
AtlasCPSEvent (*class in atlasapi.events*), 10
AtlasDataExplorerEvent (*class in atlasapi.events*), 10
AtlasEvent (*class in atlasapi.events*), 10
AtlasFeatureEvent (*class in atlasapi.events*), 10
AtlasGranularities (*class in atlasapi.lib*), 35
AtlasHostEvent (*class in atlasapi.events*), 11
AtlasLogNames (*class in atlasapi.lib*), 35
AtlasMeasurement (*class in atlasapi.measurements*), 11
AtlasMeasurementTypes (*class in atlasapi.measurements*), 12
AtlasMeasurementTypes.Asserts (*class in atlasapi.measurements*), 12
AtlasMeasurementTypes.Cache (*class in atlasapi.measurements*), 13
AtlasMeasurementTypes.CPU (*class in atlasapi.measurements*), 12
AtlasMeasurementTypes.CPU.Process (*class in atlasapi.measurements*), 12
AtlasMeasurementTypes.CPU.ProcessNormalized (*class in atlasapi.measurements*), 12
AtlasMeasurementTypes.CPU.System (*class in atlasapi.measurements*), 13
AtlasMeasurementTypes.CPU.SystemNormalized (*class in atlasapi.measurements*), 13
AtlasMeasurementTypes.Cursors (*class in atlasapi.measurements*), 13
AtlasMeasurementTypes.Db (*class in atlasapi.measurements*), 14
AtlasMeasurementTypes.Disk (*class in atlasapi.measurements*), 14
AtlasMeasurementTypes.Disk.Free (*class in atlasapi.measurements*), 14
AtlasMeasurementTypes.Disk.IOPS (*class in atlasapi.measurements*), 14
AtlasMeasurementTypes.Disk.Latency (*class in atlasapi.measurements*), 14
AtlasMeasurementTypes.Disk.Util (*class in atlasapi.measurements*), 14
AtlasMeasurementTypes.DocumentMetrics (*class in atlasapi.measurements*), 15
AtlasMeasurementTypes.ExtraInfo (*class in atlasapi.measurements*), 15
AtlasMeasurementTypes.GlobalLockCurrentQueue (*class in atlasapi.measurements*), 15
AtlasMeasurementTypes.Memory (*class in atlasapi.measurements*), 15
AtlasMeasurementTypes.Namespaces (*class in atlasapi.measurements*), 15
AtlasMeasurementTypes.Network (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.Opcounter (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.Opcounter.Repl (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.Operations (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.Operations.ExecutionTime (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.Oplog (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.QueryExecutor (*class in atlasapi.measurements*), 16
AtlasMeasurementTypes.QueryTargetingScanned (*class in atlasapi.measurements*), 17
AtlasMeasurementTypes.TicketsAvailable (*class in atlasapi.measurements*), 17
AtlasMeasurementValue (*class in atlasapi.measurements*), 17
AtlasPagination (*class in atlasapi.atlas*), 3
AtlasPeriods (*class in atlasapi.lib*), 35
AtlasUnits (*class in atlasapi.lib*), 35
automated (*atlasapi.cloud_backup.DeliveryType attribute*), 34
AWS (*atlasapi.lib.ProviderName attribute*), 37
AZURE (*atlasapi.lib.ProviderName attribute*), 37

B

backup (*atlasapi.specs.RoleSpecs attribute*), 32
BAD_REQUEST (*atlasapi.settings.Settings attribute*), 24
BASE_URL (*atlasapi.settings.Settings attribute*), 24
BYTES (*atlasapi.lib.AtlasUnits attribute*), 36
bytes_id (*atlasapi.measurements.AtlasMeasurementTypes.Network attribute*), 16
bytes_in (*atlasapi.measurements.AtlasMeasurementTypes.Network attribute*), 16
bytes_out (*atlasapi.measurements.AtlasMeasurementTypes.Network attribute*), 16

BYTES_PER_SECOND (*atlasapi.lib.AtlasUnits attribute*), 36
bytes_read (*atlasapi.measurements.AtlasMeasurementTypes.Cache attribute*), 13
bytes_written (*atlasapi.measurements.AtlasMeasurementTypes.Cache attribute*), 13

C

cancel_snapshot_restore_request() (*atlasapi.atlas.Atlas._CloudBackups method*), 51
checkAndRaise() (*atlasapi.errors.ErrPaginationLimits method*), 21
children_kernel (*atlasapi.measurements.AtlasMeasurementTypes.CPU.Process attribute*), 12
children_kernel (*atlasapi.measurements.AtlasMeasurementTypes.CPU.ProcessNormalized attribute*), 12
children_user (*atlasapi.measurements.AtlasMeasurementTypes.CPU.Process attribute*), 12
children_user (*atlasapi.measurements.AtlasMeasurementTypes.CPU.ProcessNormalized attribute*), 13
clean_list() (*in module atlasapi.measurements*), 17
clear_roles() (*atlasapi.specs.DatabaseUsersPermissionsSpecs method*), 28
CLOSED (*atlasapi.specs.AlertStatusSpec attribute*), 27
CloudBackupRequest (*class in atlasapi.cloud_backup*), 33
CloudBackupSnapshot (*class in atlasapi.cloud_backup*), 33
CloudBackupSnapshotsGetAll (*class in atlasapi.atlas*), 4
cluster_list (*atlasapi.atlas.Atlas._Hosts property*), 47
ClusterConfig (*class in atlasapi.clusters*), 7
clusterMonitor (*atlasapi.specs.RoleSpecs attribute*), 32
ClustersGetAll (*class in atlasapi.atlas*), 4
ClusterStates (*class in atlasapi.clusters*), 7
ClusterType (*class in atlasapi.lib*), 36
cmd (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute*), 16
cmd (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter.Repl attribute*), 16
collection_count (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15
commands (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute*), 16
COMPLETED (*atlasapi.cloud_backup.SnapshotStatus attribute*), 34
CONFLICT (*atlasapi.settings.Settings attribute*), 24
connections (*atlasapi.measurements.AtlasMeasurementTypes.Cache attribute*), 17
count (*atlasapi.atlas.Atlas._Organizations property*), 54
create_a_database_user() (*atlasapi.atlas.Atlas._DatabaseUsers method*), 43
create_basic_rs() (*atlasapi.atlas.Atlas._Clusters method*), 39
create_cluster() (*atlasapi.atlas.Atlas._Clusters method*), 39
create_dict (*atlasapi.projects.Project property*), 37
create_snapshot_for_cluster() (*atlasapi.atlas.Atlas._CloudBackups method*), 52
create_whitelist_entry() (*atlasapi.atlas.Atlas._Whitelist method*), 50
CREATED (*atlasapi.settings.Settings attribute*), 24
CREATING (*atlasapi.clusters.ClusterStates attribute*), 7
current_config() (*atlasapi.atlas.Atlas._MaintenanceWindows method*), 46

D

data_partition_stats() (*atlasapi.specs.Host method*), 29
data_size (*atlasapi.measurements.AtlasMeasurementTypes.Db attribute*), 14
data_size (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15
databaseName (*atlasapi.settings.Settings attribute*), 27
DatabaseUsersGetAll (*class in atlasapi.atlas*), 4
DatabaseUsersPermissionsSpecs (*class in atlasapi.specs*), 27
DatabaseUsersUpdatePermissionsSpecs (*class in atlasapi.specs*), 28
date_end (*atlasapi.measurements.AtlasMeasurement property*), 11
date_start (*atlasapi.measurements.AtlasMeasurement property*), 11
DAY (*atlasapi.lib.AtlasGranularities attribute*), 35
dbAdmin (*atlasapi.specs.RoleSpecs attribute*), 32
dbAdminAnyDatabase (*atlasapi.specs.RoleSpecs attribute*), 32
defer() (*atlasapi.atlas.Atlas._MaintenanceWindows method*), 46
delete (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute*), 16
delete (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter.Repl attribute*), 16
delete() (*atlasapi.lib.Network method*), 22
delete_a_database_user() (*atlasapi.atlas.Atlas._DatabaseUsers method*), 43

delete_a_whitelist_entry() (*atlasapi.atlas.Atlas._Whitelist method*), 50
delete_cluster() (*atlasapi.atlas.Atlas._Clusters method*), 40
DELETED (*atlasapi.clusters.ClusterStates attribute*), 8
deleted (*atlasapi.measurements.AtlasMeasurementTypes.DocumentMetrics attribute*), 15
DELETING (*atlasapi.clusters.ClusterStates attribute*), 8
DeliveryType (*class in atlasapi.cloud_backup*), 33
dirty (*atlasapi.measurements.AtlasMeasurementTypes.Cache attribute*), 13
download (*atlasapi.cloud_backup.DeliveryType attribute*), 34

E

enableSharding (*atlasapi.specs.RoleSpecs attribute*), 32
ERR_ITEMS_PER_PAGE (*atlasapi.errors.ErrPaginationLimits attribute*), 21
ERR_PAGE_NUM (*atlasapi.errors.ErrPaginationLimits attribute*), 21
ErrAtlasBackupError, 18
ErrAtlasBadRequest, 18
ErrAtlasConflict, 18
ErrAtlasDuplicateClusterName, 18
ErrAtlasForbidden, 19
ErrAtlasForbiddenWL, 19
ErrAtlasGeneric, 19
ErrAtlasJobError, 19
ErrAtlasMethodNotAllowed, 20
ErrAtlasNotFound, 20
ErrAtlasRestoreConflictError, 20
ErrAtlasServerErrors, 20
ErrAtlasUnauthorized, 20
ErrConfirmationRequested, 21
ErrMaintenanceError, 21
ErrPagination, 21
ErrPaginationLimits, 21
ErrRole, 21
EventsGetForProject (*class in atlasapi.atlas*), 4
EventsGetForProjectAndType (*class in atlasapi.atlas*), 4
extent_count (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15

F

FAILED (*atlasapi.cloud_backup.SnapshotStatus attribute*), 34
FALLBACK (*atlasapi.cloud_backup.SnapshotType attribute*), 34
fetch() (*atlasapi.atlas.AlertsGetAll method*), 3
fetch() (*atlasapi.atlas.EventsGetForProject method*), 4
fetch() (*atlasapi.atlas.EventsGetForProjectAndType method*), 4
fetch() (*atlasapi.atlas.OrganizationProjectsGetAll method*), 5
file_request_timeout (*atlasapi.settings.Settings attribute*), 27
fill_from_dict() (*atlasapi.clusters.AdvancedOptions class method*), 6
fill_from_dict() (*atlasapi.clusters.ClusterConfig class method*), 7
fill_from_dict() (*atlasapi.whitelist.WhitelistEntry class method*), 18
fill_host_list() (*atlasapi.atlas.Atlas._Hosts method*), 47
FIVE_MINUTE (*atlasapi.lib.AtlasGranularities attribute*), 35
for_create() (*atlasapi.projects.Project class method*), 37
FORBIDDEN (*atlasapi.settings.Settings attribute*), 24
FRIDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33
from_dict() (*atlasapi.cloud_backup.CloudBackupSnapshot class method*), 33
from_dict() (*atlasapi.cloud_backup.SnapshotRestoreResponse class method*), 34
from_dict() (*atlasapi.clusters.ProviderSettings class method*), 9
from_dict() (*atlasapi.clusters.ReplicationSpecs class method*), 9
from_dict() (*atlasapi.maintenance_window.MaintenanceWindow class method*), 33
from_dict() (*atlasapi.organizations.Organization class method*), 38
from_dict() (*atlasapi.projects.Project class method*), 37
from_dict() (*atlasapi.projects.ProjectSettings class method*), 38

G

GCP (*atlasapi.lib.ProviderName attribute*), 37
GEOSHARDED (*atlasapi.lib.ClusterType attribute*), 36
get() (*atlasapi.network.Network method*), 23
get_a_single_database_user() (*atlasapi.atlas.Atlas._DatabaseUsers method*), 43
get_all_alerts() (*atlasapi.atlas.Atlas._Alerts method*), 45
get_all_clusters() (*atlasapi.atlas.Atlas._Clusters method*), 40
get_all_database_users() (*atlasapi.atlas.Atlas._DatabaseUsers method*), 43
get_all_projects_for_org() (*atlasapi.atlas.Atlas._Organizations method*),

55
get_all_whitelist_entries() (atlasapi.atlas.Atlas._Whitelist method), 51
get_an_alert() (atlasapi.atlas.Atlas._Alerts method), 45
get_backup_snapshot_for_cluster() (atlasapi.atlas.Atlas._CloudBackups method), 52
get_backup_snapshots_for_cluster() (atlasapi.atlas.Atlas._CloudBackups method), 52
get_big() (atlasapi.network.Network method), 23
get_databases() (atlasapi.specs.Host method), 29
get_file() (atlasapi.network.Network method), 23
get_log_for_host() (atlasapi.atlas.Atlas._Hosts method), 47
get_loglines_for_host() (atlasapi.atlas.Atlas._Hosts method), 47
get_logs_for_cluster() (atlasapi.atlas.Atlas._Hosts method), 48
get_logs_for_project() (atlasapi.atlas.Atlas._Hosts method), 48
get_measurement_for_host() (atlasapi.specs.Host method), 29
get_measurement_for_hosts() (atlasapi.atlas.Atlas._Hosts method), 48
get_measurements_for_database() (atlasapi.specs.Host method), 30
get_measurements_for_disk() (atlasapi.specs.Host method), 30
get_partitions() (atlasapi.specs.Host method), 31
get_project_teams() (atlasapi.atlas.Atlas._Projects method), 53
get_project_users() (atlasapi.atlas.Atlas._Projects method), 53
get_single_cluster() (atlasapi.atlas.Atlas._Clusters method), 40
get_single_cluster_advanced_options() (atlasapi.atlas.Atlas._Clusters method), 41
get_single_cluster_as_obj() (atlasapi.atlas.Atlas._Clusters method), 41
get_snapshot_restore_requests() (atlasapi.atlas.Atlas._CloudBackups method), 52
get_whitelist_entry() (atlasapi.atlas.Atlas._Whitelist method), 51
getAtlasResponse() (atlasapi.errors.ErrAtlasGeneric method), 19
getmore (atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute), 16
getSpecs() (atlasapi.specs.DatabaseUsersPermissionsSpecs method), 28
getSpecs() (atlasapi.specs.DatabaseUsersUpdatePermissionsSpecs method), 29
GIGABYTES (atlasapi.lib.AtlasUnits attribute), 36
GIGABYTES_PER_HOUR (atlasapi.lib.AtlasUnits attribute), 36
guest (atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute), 13
guest (atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNorma attribute), 13

H

Host (class in atlasapi.specs), 29
host_list_by_cluster() (atlasapi.atlas.Atlas._Hosts method), 48
host_list_primaries (atlasapi.atlas.Atlas._Hosts property), 49
host_list_secondaries (atlasapi.atlas.Atlas._Hosts property), 49
host_names (atlasapi.atlas.Atlas._Hosts property), 49
HostLogFile (class in atlasapi.specs), 31
HostsGetAll (class in atlasapi.atlas), 5
HOUR (atlasapi.lib.AtlasGranularities attribute), 35
HOURS_1 (atlasapi.lib.AtlasPeriods attribute), 35
HOURS_24 (atlasapi.lib.AtlasPeriods attribute), 35
HOURS_48 (atlasapi.lib.AtlasPeriods attribute), 35
HOURS_8 (atlasapi.lib.AtlasPeriods attribute), 35

I

IAMType (class in atlasapi.specs), 31
IDLE (atlasapi.clusters.ClusterStates attribute), 8
index_count (atlasapi.measurements.AtlasMeasurementTypes.Namespace attribute), 15
index_size (atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute), 15
INPROGRESS (atlasapi.cloud_backup.SnapshotStatus attribute), 34
insert (atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute), 16
insert (atlasapi.measurements.AtlasMeasurementTypes.Opcounter.Repl attribute), 16
inserted (atlasapi.measurements.AtlasMeasurementTypes.DocumentMetr attribute), 15
InstanceSizeName (class in atlasapi.clusters), 8
iowait (atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute), 13
iowait (atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNorm attribute), 13
irq (atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute), 13
irq (atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNormaliz attribute), 13
is_existing_cluster() (atlasapi.atlas.Atlas._Clusters method), 41
is_existing_snapshot() (atlasapi.atlas.Atlas._CloudBackups method), 52
itemsPerPage (atlasapi.settings.Settings attribute), 27

itemsPerPageMax (*atlasapi.settings.Settings attribute*), 27
itemsPerPageMin (*atlasapi.settings.Settings attribute*), 27

K

kernel (*atlasapi.measurements.AtlasMeasurementTypes.CPU.Access attribute*), 12
kernel (*atlasapi.measurements.AtlasMeasurementTypes.CPU.Normalized attribute*), 13
kernel (*atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute*), 13
kernel (*atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNormalized attribute*), 13

L

LogLine (*class in atlasapi.lib*), 36

M

M0 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M10 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M100 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M140 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M2 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M20 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M200 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M200_NVME (*atlasapi.clusters.InstanceSizeName attribute*), 8
M30 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M300 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M40 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M400 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M400_NVME (*atlasapi.clusters.InstanceSizeName attribute*), 8
M40_NVME (*atlasapi.clusters.InstanceSizeName attribute*), 8
M5 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M50 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M50_NVME (*atlasapi.clusters.InstanceSizeName attribute*), 8
M60 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M60_NVME (*atlasapi.clusters.InstanceSizeName attribute*), 8
M80 (*atlasapi.clusters.InstanceSizeName attribute*), 8
M80_NVME (*atlasapi.clusters.InstanceSizeName attribute*), 8
MaintenanceWindow (*class in atlasapi.maintenance_window*), 32
mapped (*atlasapi.measurements.AtlasMeasurementTypes.Memory attribute*), 15
master_time (*atlasapi.measurements.AtlasMeasurementTypes.Oplog attribute*), 16
measurement_stats (*atlasapi.measurements.AtlasMeasurement property*), 11
measurement_stats_friendly (*atlasapi.measurements.AtlasMeasurement property*), 12
measurements (*atlasapi.measurements.AtlasMeasurement property*), 12
measurements_as_tuples() (*atlasapi.measurements.AtlasMeasurement method*), 12
measurements_count (*atlasapi.measurements.AtlasMeasurement property*), 12
MEGABYTES_PER_SECOND (*atlasapi.lib.AtlasUnits attribute*), 36
merge() (*in module atlasapi.network*), 24
METHOD_NOT_ALLOWED (*atlasapi.settings.Settings attribute*), 24
MILLISECONDS (*atlasapi.lib.AtlasUnits attribute*), 36
MINUTE (*atlasapi.lib.AtlasGranularities attribute*), 35
MINUTES_15 (*atlasapi.lib.AtlasPeriods attribute*), 35
modify_cluster() (*atlasapi.atlas.Atlas._Clusters method*), 41
modify_cluster_advanced_options() (*atlasapi.atlas.Atlas._Clusters method*), 42
modify_cluster_instance_size() (*atlasapi.atlas.Atlas._Clusters method*), 42
modify_cluster_tls() (*atlasapi.atlas.Atlas._Clusters method*), 42
module
    atlasapi.alerts, 5
    atlasapi.atlas, 3
    atlasapi.atlas_types, 5
    atlasapi.cloud_backup, 33
    atlasapi.clusters, 5
    atlasapi.errors, 18
    atlasapi.events, 10
    atlasapi.lib, 35
    atlasapi.maintenance_window, 32
    atlasapi.measurements, 11
    atlasapi.network, 22
    atlasapi.organizations, 38
    atlasapi.projects, 37
    atlasapi.settings, 24
    atlasapi.specs, 27
    atlasapi.whitelist, 18
MONDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33
MONGOD_AUDIT (*atlasapi.lib.AtlasLogNames attribute*), 35
MONGODB (*atlasapi.lib.AtlasLogNames attribute*), 35
MongoDBMajorVersion (*class in atlasapi.lib*), 36
MONGOS (*atlasapi.lib.AtlasLogNames attribute*), 35

MONGOS_AUDIT (*atlasapi.lib.AtlasLogNames attribute*), 35
MONTHS_1 (*atlasapi.lib.AtlasPeriods attribute*), 35
MONTHS_2 (*atlasapi.lib.AtlasPeriods attribute*), 35
msg (*atlasapi.measurements.AtlasMeasurementTypes.Asserts attribute*), 12

N

Network (*class in atlasapi.network*), 22
nice (*atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute*), 13
nice (*atlasapi.measurements.AtlasMeasurementTypes.CPUSystemNormalized attribute*), 13
NO_CONTENT (*atlasapi.settings.Settings attribute*), 24
NO_DATA (*atlasapi.specs.ReplicaSetTypes attribute*), 31
NONE (*atlasapi.specs.IAMType attribute*), 31
NOTFOUND (*atlasapi.settings.Settings attribute*), 24
num_requests (*atlasapi.measurements.AtlasMeasurementTypes.Network attribute*), 16

O

object_count (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15
object_size (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15
objects_per_returned (*atlasapi.measurements.AtlasMeasurementTypes.QueryTargetingScanned attribute*), 17
ONDEMAND (*atlasapi.cloud_backup.SnapshotType attribute*), 34
open (*atlasapi.measurements.AtlasMeasurementTypes.Cursors attribute*), 13
OPEN (*atlasapi.specs.AlertStatusSpec attribute*), 27
Organization (*class in atlasapi.organizations*), 38
organization_by_id() (*atlasapi.atlas.Atlas._Organizations method*), 55
organization_by_name() (*atlasapi.atlas.Atlas._Organizations method*), 55
OrganizationProjectsGetAll (*class in atlasapi.atlas*), 5
organizations (*atlasapi.atlas.Atlas._Organizations property*), 55

P

page_faults (*atlasapi.measurements.AtlasMeasurementTypes.ExtraInfo attribute*), 15
pageNum (*atlasapi.settings.Settings attribute*), 27
patch() (*atlasapi.network.Network method*), 23
pause_cluster() (*atlasapi.atlas.Atlas._Clusters method*), 42
per_returned (*atlasapi.measurements.AtlasMeasurementTypes.QueryTargetingScanned attribute*), 17
PERCENT (*atlasapi.lib.AtlasUnits attribute*), 36
percent_fee (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
percent_free_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
percent_used (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
percent_used_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
POINT_IN_TIME (*atlasapi.cloud_backup.DeliveryType attribute*), 34
post() (*atlasapi.network.Network method*), 24
Project (*class in atlasapi.projects*), 37
project_by_id() (*atlasapi.atlas.Atlas._Projects method*), 54
project_by_name() (*atlasapi.atlas.Atlas._Projects method*), 54
projects (*atlasapi.atlas.Atlas._Projects property*), 54
ProjectSettings (*class in atlasapi.projects*), 37
ProviderName (*class in atlasapi.lib*), 36
ProviderSettings (*class in atlasapi.clusters*), 9
PROVISIONED (*atlasapi.clusters.VolumeTypes attribute*), 10

Q

query (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute*), 16
QUEUED (*atlasapi.cloud_backup.SnapshotStatus attribute*), 34

R

R200 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R300 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R40 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R400 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R50 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R60 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R700 (*atlasapi.clusters.InstanceSizeName attribute*), 9
R80 (*atlasapi.clusters.InstanceSizeName attribute*), 9
rate (*atlasapi.measurements.AtlasMeasurementTypes.Oplog attribute*), 16
read (*atlasapi.measurements.AtlasMeasurementTypes.Disk.IOPS attribute*), 14
read (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Latency attribute*), 14
read (*atlasapi.specs.RoleSpecs attribute*), 32
read_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.IOPS attribute*), 14
read_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Latency attribute*), 14
readOnDatabase (*atlasapi.specs.RoleSpecs attribute*), 32

readers (*atlasapi.measurements.AtlasMeasurementTypes.GlobalLockObjects attribute*), 15
reads (*atlasapi.measurements.AtlasMeasurementTypes.Operations.ExecutionTime attribute*), 16
reads (*atlasapi.measurements.AtlasMeasurementTypes.TicketsAvailable attribute*), 17
readWrite (*atlasapi.specs.RoleSpecs attribute*), 32
readWriteAnyDatabase (*atlasapi.specs.RoleSpecs attribute*), 32
RECOVERING (*atlasapi.specs.ReplicaSetTypes attribute*), 31
RegionConfig (*class in atlasapi.clusters*), 9
regular (*atlasapi.measurements.AtlasMeasurementTypes.Asserts attribute*), 12
remove_role() (*atlasapi.specs.DatabaseUsersPermissionsSpecs method*), 28
remove_roles() (*atlasapi.specs.DatabaseUsersPermissionsSpecs method*), 28
REPAIRING (*atlasapi.clusters.ClusterStates attribute*), 8
REPLICA_PRIMARY (*atlasapi.specs.ReplicaSetTypes attribute*), 31
REPLICA_SECONDARY (*atlasapi.specs.ReplicaSetTypes attribute*), 31
REPLICASET (*atlasapi.lib.ClusterType attribute*), 36
ReplicaSetTypes (*class in atlasapi.specs*), 31
ReplicationSpecs (*class in atlasapi.clusters*), 9
request_snapshot_restore() (*atlasapi.atlas.Atlas._CloudBackups method*), 52
request_snapshot_restore_to_group() (*atlasapi.atlas.Atlas._CloudBackups method*), 52
requests_timeout (*atlasapi.settings.Settings attribute*), 27
resident (*atlasapi.measurements.AtlasMeasurementTypes.Memory attribute*), 15
returned (*atlasapi.measurements.AtlasMeasurementTypes.Document attribute*), 15
ROLE (*atlasapi.specs.IAMType attribute*), 31
RoleSpecs (*class in atlasapi.specs*), 32

S

SATURDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33
SCALAR (*atlasapi.lib.AtlasUnits attribute*), 36
SCALAR_PER_SECOND (*atlasapi.lib.AtlasUnits attribute*), 36
scan_and_order (*atlasapi.measurements.AtlasMeasurementTypes.Operations attribute*), 16
scanned (*atlasapi.measurements.AtlasMeasurementTypes.QueryExecutor attribute*), 17
scanned_objects (*atlasapi.measurements.AtlasMeasurementTypes.QueryExecutor attribute*), 17
SCHEDULED (*atlasapi.cloud_backup.SnapshotType attribute*), 34
SERVER_ERRORS (*atlasapi.settings.Settings attribute*), 24
set_config() (*atlasapi.atlas.Atlas._MaintenanceWindows method*), 46
settings (*atlasapi.atlas.Atlas._Projects property*), 54
Settings (*class in atlasapi.settings*), 24
SHARD_CONFIG (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARD_CONFIG_PRIMARY (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARD_CONFIG_SECONDARY (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARD_MONGOS (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARD_PRIMARY (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARD_SECONDARY (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARD_STANDALONE (*atlasapi.specs.ReplicaSetTypes attribute*), 32
SHARDED (*atlasapi.lib.ClusterType attribute*), 36
SHARDEDCLUSTER (*atlasapi.lib.ClusterType attribute*), 36
ShardedClusterConfig (*class in atlasapi.clusters*), 9
since() (*atlasapi.atlas.Atlas._Events method*), 50
since_by_type() (*atlasapi.atlas.Atlas._Events method*), 50
SnapshotRestore (*class in atlasapi.cloud_backup*), 34
SnapshotRestoreResponse (*class in atlasapi.cloud_backup*), 34
SnapshotStatus (*class in atlasapi.cloud_backup*), 34
SnapshotType (*class in atlasapi.cloud_backup*), 34
softirq (*atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute*), 13
softirqNorm (*atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNorm attribute*), 13
space_free (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
space_free_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
STANDARD (*atlasapi.clusters.VolumeTypes attribute*), 10
StatisticalValues (*class in atlasapi.measurements*), 17
StatisticalValuesFriendly (*class in atlasapi.measurements*), 17
steal (*atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute*), 13
stealNorm (*atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNorma attribute*), 13
storage (*atlasapi.measurements.AtlasMeasurementTypes.Db*

attribute), 14
storage_size (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15
SUCCESS (*atlasapi.settings.Settings attribute*), 25
SUNDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33

T

TEN_SECOND (*atlasapi.lib.AtlasGranularities attribute*), 35
TENANT (*atlasapi.lib.ProviderName attribute*), 37
test_failover() (*atlasapi.atlas.Atlas._Clusters method*), 42
THURSDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33
timed_out (*atlasapi.measurements.AtlasMeasurementTypes.Cursors attribute*), 14
TLS1_0 (*atlasapi.clusters.TLSProtocols attribute*), 10
TLS1_1 (*atlasapi.clusters.TLSProtocols attribute*), 10
TLS1_2 (*atlasapi.clusters.TLSProtocols attribute*), 10
TLS1_3 (*atlasapi.clusters.TLSProtocols attribute*), 10
TLSProtocols (*class in atlasapi.clusters*), 10
total (*atlasapi.measurements.AtlasMeasurementTypes.Disk.IOPS attribute*), 14
total (*atlasapi.measurements.AtlasMeasurementTypes.GlobalLocksInfo attribute*), 15
total_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.IOPS attribute*), 14
TRACKING (*atlasapi.specs.AlertStatusSpec attribute*), 27
try_bool() (*in module atlasapi.cloud_backup*), 34
try_date() (*in module atlasapi.cloud_backup*), 34
TUESDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33

U

unacknowledge_an_alert() (*atlasapi.atlas.Atlas._Alerts method*), 45
UNAUTHORIZED (*atlasapi.settings.Settings attribute*), 25
UNKNOWN (*atlasapi.clusters.ClusterStates attribute*), 8
update (*atlasapi.measurements.AtlasMeasurementTypes.Opcounter attribute*), 16
update (*atlasapi.measurements.AtlasMeasurementTypes.OpcountersRepl attribute*), 16
update_a_database_user() (*atlasapi.atlas.Atlas._DatabaseUsers method*), 44
update_host_list() (*atlasapi.atlas.Atlas._Hosts method*), 49
updated (*atlasapi.measurements.AtlasMeasurementTypes.DiskIOMeasurements attribute*), 15
UPDATING (*atlasapi.clusters.ClusterStates attribute*), 8
URI_STUB (*atlasapi.settings.Settings attribute*), 25
used (*atlasapi.measurements.AtlasMeasurementTypes.Cache attribute*), 13
used (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
used_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Free attribute*), 14
user (*atlasapi.measurements.AtlasMeasurementTypes.Asserts attribute*), 12
user (*atlasapi.measurements.AtlasMeasurementTypes.CPU.Process attribute*), 12
user (*atlasapi.measurements.AtlasMeasurementTypes.CPU.ProcessNormal attribute*), 13
user (*atlasapi.measurements.AtlasMeasurementTypes.CPU.System attribute*), 13
user (*atlasapi.measurements.AtlasMeasurementTypes.CPU.SystemNormali attribute*), 13
USER (*atlasapi.specs.IAMType attribute*), 31
user_count() (*atlasapi.atlas.Atlas._Projects method*), 54
util (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Util attribute*), 14
util_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Util attribute*), 15

V

v3_4 (*atlasapi.lib.MongoDBMajorVersion attribute*), 36
v3_6 (*atlasapi.lib.MongoDBMajorVersion attribute*), 36
v4_0 (*atlasapi.lib.MongoDBMajorVersion attribute*), 36
v4_2 (*atlasapi.lib.MongoDBMajorVersion attribute*), 36
v4_4 (*atlasapi.lib.MongoDBMajorVersion attribute*), 36
v5_0 (*atlasapi.lib.MongoDBMajorVersion attribute*), 36
value_float (*atlasapi.measurements.AtlasMeasurementValue property*), 17
value_int (*atlasapi.measurements.AtlasMeasurementValue property*), 17
view_count (*atlasapi.measurements.AtlasMeasurementTypes.Namespaces attribute*), 15
virtual (*atlasapi.measurements.AtlasMeasurementTypes.Memory attribute*), 15
VolumeTypes (*class in atlasapi.clusters*), 10
vX_x (*atlasapi.lib.MongoDBMajorVersion attribute*), 36

W

warning (*atlasapi.measurements.AtlasMeasurementTypes.Asserts attribute*), 12
WEDNESDAY (*atlasapi.maintenance_window.Weekdays attribute*), 33
Weekdays (*class in atlasapi.maintenance_window*), 33
WEEKS_1 (*atlasapi.lib.AtlasPeriods attribute*), 35
WEEKS_4 (*atlasapi.lib.AtlasPeriods attribute*), 35
WhitelistEntry (*class in atlasapi.whitelist*), 18
WhitelistGetAll (*class in atlasapi.atlas*), 5
write (*atlasapi.measurements.AtlasMeasurementTypes.Disk.IOPS attribute*), 14
write (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Latency attribute*), 14

write_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.IOPS attribute*), 14
write_max (*atlasapi.measurements.AtlasMeasurementTypes.Disk.Latency attribute*), 14
writers (*atlasapi.measurements.AtlasMeasurementTypes.GlobalLockCurrentQueue attribute*), 15
writes (*atlasapi.measurements.AtlasMeasurementTypes.Operations.ExecutionTime attribute*), 16
writes (*atlasapi.measurements.AtlasMeasurementTypes.TicketsAvailable attribute*), 17

Y

YEARS_1 (*atlasapi.lib.AtlasPeriods attribute*), 35
YEARS_2 (*atlasapi.lib.AtlasPeriods attribute*), 35