
atlasapi Documentation

Release 0.14.1

Matthew G. Monteleone

Jun 09, 2022

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	10
1.7	atlasapi.whitelist module	10
1.8	atlasapi.errors module	10
1.9	atlasapi.network module	14
1.10	atlasapi.settings module	15
1.11	atlasapi.specs module	16
1.12	atlasapi.maintenance_window module	24
1.13	atlasapi.cloud_backup module	25
1.14	atlasapi.lib module	27
1.15	atlasapi.projects module	29
1.16	atlasapi.organizations module	30
2	Nested class for atlasapi.atlas::Atlas	31
2.1	Atlas._Clusters	31
2.2	Atlas._DatabaseUsers	34
2.3	Atlas._Alerts	35
2.4	Atlas._MaintenanceWindows	37
2.5	Atlas._Hosts	37
2.6	Atlas._Events	40
2.7	Atlas._Whitelist	40
2.8	Atlas._CloudBackups	41
2.9	Atlas._Projects	43
2.10	Atlas._Organizations	44
3	atlascli - A Command line program for MongoDB Atlas	47
4	Indices and tables	49
	Python Module Index	51
	Index	53

Python Bindings for the Atlas Public API

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: *atlasapi.atlas.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: str = None, auth_method: 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
- **override to Basic if needed for use with a Proxy.** (*can*) –

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: *atlasapi.atlas.AtlasPagination*

Pagination for Database User : Get All

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

Pagination for Clusters : Get All

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

Pagination for Database User : Get All

class atlasapi.atlas.**EventsGetForProject** (*atlas: atlasapi.atlas.Atlas, since_datetime: date-time.datetime, pageNum: int, itemsPerPage: int*)
Bases: *atlasapi.atlas.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: atlasapi.atlas.Atlas, pageNum: int, itemsPerPage: int*)
Bases: *atlasapi.atlas.AtlasPagination*

Pagination for Processes : Get All

class atlasapi.atlas.**WhitelistGetAll** (*atlas, pageNum, itemsPerPage*)
Bases: *atlasapi.atlas.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: Op-
                                         tional[atlasapi.clusters.TLSProtocols] = None, no-
                                         TableScan: Optional[bool] = None, oplogSizeMB:
                                         Optional[int] = None, sampleSizeBIConnector:
                                         Optional[int] = None, sampleRefreshIntervalBI-
                                         Connector: 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 mongosql process re-samples data to create its relational schema.

as_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: atlas-
api.clusters.InstanceSizeName =
<InstanceSizeName.M10: 'M10'>,
disk_size: int = 10, provider: atlas-
api.lib.ProviderName = <Provider-
Name.AWS: 'Amazon Web Services'>,
region: str = 'US_WEST_2', version:
atlasapi.lib.MongoDBMajorVersion =
<MongoDBMajorVersion.v4_0: '4.0'>)
```

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: atlas-
api.lib.ClusterType = <ClusterType.REPLICASET:
'Replica Set'>, disk_size_gb: int = 32, name:
str = None, mongodb_major_version: atlas-
api.lib.MongoDBMajorVersion = <MongoDB-
MajorVersion.v4_0: '4.0'>, mongodb_version: Op-
tional[str] = None, num_shards: int = 1, mongo_uri:
Optional[str] = None, mongo_uri_updated: Op-
tional[str] = None, mongo_uri_with_options:
Optional[str] = None, paused: bool = False,
pit_enabled: bool = False, replication_factor:
Optional[int] = None, state_name: Op-
tional[atlasapi.clusters.ClusterStates] = None,
autoscaling: dict = None, replication_specs: atlas-
api.clusters.ReplicationSpecs = None, srv_address:
Optional[str] = None, providerSettings: Op-
tional[atlasapi.clusters.ProviderSettings] = None,
links: list = None, id: Optional[str] = None)
```

Bases: object

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

<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

Bases: enum.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

Bases: enum.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: atlasapi.clusters.InstanceSizeName =
    <InstanceSizeName.M10: 'M10'>, provider:
    atlasapi.lib.ProviderName = <Provider-
    Name.AWS: 'Amazon Web Services'>, re-
    gion: str = 'US_WEST_1', autoScaling:
    Optional[dict] = None, diskIOPS: int = None,
    encryptEBSVolume: bool = True, volumeType:
    atlasapi.clusters.VolumeTypes = <Volume-
    Types.STANDARD: '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] = '2ad23799-539d-4802-aac4-376833c33a1b', 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: atlasapi.lib.ClusterType = <ClusterType.REPLICASET: 'Replica Set'>, disk_size_gb: int = 32, name: str = None, mongodb_major_version: atlasapi.lib.MongoDBMajorVersion = <MongoDBMajorVersion.v4_0: '4.0'>, 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[atlasapi.clusters.ClusterStates] = None, autoscaling: dict = {}, replication_specs: list = [], srv_address: Optional[str] = None, providerSettings: Optional[atlasapi.clusters.ProviderSettings] = None, links: list = None, id: Optional[str] = None)
```

Bases: *atlasapi.clusters.ClusterConfig*

```
as_dict () → dict
```

```
class atlasapi.clusters.TLSProtocols
```

Bases: enum.Enum

An enumeration.

```
TLS1_0 = 'TLS1_0'
```

```
TLS1_1 = 'TLS1_1'
```

```
TLS1_2 = 'TLS1_2'
```

```
TLS1_3 = 'TLS1_3'
```

```
class atlasapi.clusters.VolumeTypes
```

Bases: enum.Enum

The volume types available on atlas

```
PROVISIONED = 'Provisioned'
```

```
STANDARD = 'Standard'
```

1.5 atlasapi.events module

```
class atlasapi.events.AtlasCPSEvent (value_dict: dict)
    Bases: atlasapi.events._AtlasBaseEvent

class atlasapi.events.AtlasClusterEvent (value_dict: dict)
    Bases: atlasapi.events._AtlasBaseEvent

class atlasapi.events.AtlasDataExplorerEvent (value_dict: dict)
    Bases: atlasapi.events._AtlasUserBaseEvent

class atlasapi.events.AtlasEvent (value_dict: dict)
    Bases: atlasapi.events._AtlasBaseEvent

class atlasapi.events.AtlasFeatureEvent (value_dict: dict)
    Bases: atlasapi.events._AtlasUserBaseEvent

class atlasapi.events.AtlasHostEvent (value_dict: dict)
    Bases: atlasapi.events._AtlasBaseEvent

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

1.6 atlasapi.measurements module

1.7 atlasapi.whitelist module

```
class atlasapi.whitelist.WhitelistEntry (cidrBlock: str = None, comment: str = None, ipAd-
    dress: str = None, links: list = None, last_used: str
    = None, count: int = None, last_used_address: 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: atlasapi.errors.ErrAtlasGeneric

    Atlas : Atlas Backup

    Constructor
```

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Bad Request

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Conflict

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Duplicate Clustername

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Forbidden

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Forbidden by WhiteList

Constructor

Parameters

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

exception atlasapi.errors.**ErrAtlasGeneric** (*msg, c, details*)

Bases: Exception

Atlas Generic Exception

Constructor

Parameters

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

getAtlasResponse ()

Get details about the Atlas response

Returns HTTP code, Response payload

Return type int, str

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Job error Clustername

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Method Not Allowed

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Not Found

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : RestoreConflictError

Constructor

Parameters

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

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

Bases: *atlasapi.errors.ErrAtlasGeneric*

Atlas : Server Errors

Constructor

Parameters

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

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

Bases: *atlasapi.errors.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.**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:
                                Union[requests.auth.HTTPDigestAuth,
                                quests.auth.HTTPBasicAuth] = <class 're-
                                quests.auth.HTTPDigestAuth'>)
```

Bases: `object`

Network constructor

Parameters

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

answer (*c*, *details*: `Union[dict, _io.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`
- `ErrAtlasServerError`

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_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

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/alert.
databaseName = 'admin'
file_request_timeout = 360
itemsPerPage = 1000
itemsPerPageMax = 2000
itemsPerPageMin = 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.AtlasMeasurement (name: atlasapi.specs.AtlasMeasurementTypes,
                                         period: atlasapi.lib.AtlasPeriods, granularity:
                                         atlasapi.lib.AtlasGranularities, measurements:
                                         List[atlasapi.specs.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
- **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

as_dict

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

Returns

Return type dict

date_end

The date of the last measurement

Returns The date of the last measurement.

Return type datetime

date_start

The date of the first measurement.

Returns The date of the first measurement.

Return type datetime

measurement_stats

Returns a statistical info for measurement data

measurement_stats_friendly_bytes

Returns statistical info for measurement data in friendly bytes format

measurement_stats_friendly_number

Returns statistical info for measurement data in friendly bytes format

measurements

Getter for the measurements.

Returns An iterator containing values objects.

Return type Iterator[[AtlasMeasurementValue](#)]

measurements_as_tuples()**measurements_count**

The count of measurements

Returns The count of measurements in the set

Return type int

class atlasapi.specs.AtlasMeasurementTypes

Bases: `atlasapi.lib._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: `atlasapi.lib._GetAll`

`msg = 'ASSERT_MSG'`

```
regular = 'ASSERT_REGULAR'
user = 'ASSERT_USER'
warning = 'ASSERT_WARNING'

class CPU
    Bases: atlasapi.lib._GetAll

    class Process
        Bases: atlasapi.lib._GetAll

        children_kernel = 'PROCESS_CPU_CHILDREN_KERNEL'
        children_user = 'PROCESS_CPU_CHILDREN_USER'
        kernel = 'PROCESS_CPU_KERNEL'
        user = 'PROCESS_CPU_USER'

    class ProcessNormalized
        Bases: atlasapi.lib._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: atlasapi.lib._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: atlasapi.lib._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: atlasapi.lib._GetAll

    bytes_read = 'CACHE_BYTES_READ_INTO'
    bytes_written = 'CACHE_BYTES_WRITTEN_FROM'
    dirty = 'CACHE_DIRTY_BYTES'
    used = 'CACHE_USED_BYTES'

class Cursors
    Bases: atlasapi.lib._GetAll

    open = 'CURSORS_TOTAL_OPEN'
    timed_out = 'CURSORS_TOTAL_TIMED_OUT'

class Db
    Bases: atlasapi.lib._GetAll

    data_size = 'DB_DATA_SIZE_TOTAL'
    storage = 'DB_STORAGE_TOTAL'

class DocumentMetrics
    Bases: atlasapi.lib._GetAll

    deleted = 'DOCUMENT_METRICS_DELETED'
    inserted = 'DOCUMENT_METRICS_INSERTED'
    returned = 'DOCUMENT_METRICS_RETURNED'
    updated = 'DOCUMENT_METRICS_UPDATED'

class ExtraInfo
    Bases: atlasapi.lib._GetAll

    page_faults = 'EXTRA_INFO_PAGE_FAULTS'

class GlobalLockCurrentQueue
    Bases: atlasapi.lib._GetAll

    readers = 'GLOBAL_LOCK_CURRENT_QUEUE_READERS'
    total = 'GLOBAL_LOCK_CURRENT_QUEUE_TOTAL'
    writers = 'GLOBAL_LOCK_CURRENT_QUEUE_WRITERS'

class Memory
    Bases: atlasapi.lib._GetAll

    mapped = 'MEMORY_MAPPED'
    resident = 'MEMORY_RESIDENT'
    virtual = 'MEMORY_VIRTUAL'

class Network
    Bases: atlasapi.lib._GetAll

    bytes_id = 'NETWORK_BYTES_IN'
    bytes_in = 'NETWORK_BYTES_IN'
    bytes_out = 'NETWORK_BYTES_OUT'
    num_requests = 'NETWORK_NUM_REQUESTS'
```

```
class Opcounter
    Bases: atlasapi.lib._GetAll

    class Repl
        Bases: atlasapi.lib._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: atlasapi.lib._GetAll

    class ExecutionTime
        Bases: atlasapi.lib._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: atlasapi.lib._GetAll

    master_time = 'OPLOG_MASTER_TIME'

    rate = 'OPLOG_RATE_GB_PER_HOUR'

class QueryExecutor
    Bases: atlasapi.lib._GetAll

    scanned = 'QUERY_EXECUTOR_SCANNED'

    scanned_objects = 'QUERY_EXECUTOR_SCANNED_OBJECTS'

class QueryTargetingScanned
    Bases: atlasapi.lib._GetAll

    objects_per_returned = 'QUERY_TARGETING_SCANNED_OBJECTS_PER_RETURNED'

    per_returned = 'QUERY_TARGETING_SCANNED_PER_RETURNED'

class TicketsAvailable
    Bases: atlasapi.lib._GetAll

    reads = 'TICKETS_AVAILABLE_READS'

    writes = 'TICKETS_AVAILABLE_WRITES'

connections = 'CONNECTIONS'
```



```

class atlasapi.specs.AtlasMeasurementValue (value_dict: dict)
    Bases: object

    as_dict () → dict

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

    value_float

    value_int

class atlasapi.specs.DatabaseUsersPermissionsSpecs (username: str, password: str
                                                    = None, aws_iam_type: Op-
                                                    tional[atlasapi.specs.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: NewType.<locals>.new_type = 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[atlasapi.specs.RoleSpecs], collectionName: 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 collection-
        Name

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: *atlasapi.specs.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: atlasapi.lib.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

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

```
class atlasapi.specs.HostLogFile(log_name: atlasapi.lib.AtlasLogNames = None,
                                log_file_binary: BinaryIO = None)
```

Bases: `object`

```
class atlasapi.specs.IAMType
```

Bases: `enum.Enum`

An enumeration.

NONE = 'None'

ROLE = 'ROLE'

USER = 'USER'

```
class atlasapi.specs.ReplicaSetTypes
```

Bases: `enum.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'

class atlasapi.specs.StatisticalValues (data_list: list)
    Bases: object

class atlasapi.specs.StatisticalValuesFriendly (data_list: list, data_type: str = None)
    Bases: object

atlasapi.specs.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.
```

1.12 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: atlas-
    api.maintenance_window.Weekdays
    = <Weekdays.SUNDAY: 1>,
    hour_of_day: int = 23, num-
    ber_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 formatted dict

Returns:

```
class atlasapi.maintenance_window.Weekdays
```

Bases: `enum.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`

```
as_dict
```

```
class atlasapi.cloud_backup.CloudBackupSnapshot (id: Optional[str] = None, cloud_provider: Optional[atlasapi.lib.ProviderName] = None, created_at: Optional[datetime.datetime] = None, description: Optional[str] = None, expires_at: Optional[datetime.datetime] = None, links: Optional[List[T]] = 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[atlasapi.cloud_backup.SnapshotType] = None, status: Optional[atlasapi.cloud_backup.SnapshotStatus] = None, storage_size_bytes: Optional[int] = None, type: Optional[atlasapi.lib.ClusterType] = None))
```

Bases: `object`

```
    classmethod from_dict (data_dict: dict)

class atlasapi.cloud_backup.DeliveryType
    Bases: enum.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: atlas-
                                             api.cloud_backup.DeliveryType,
                                             snapshot_id: str, target_cluster_name: str =
                                             None, target_group_id: str = None)

    Bases: object

    as_dict

class atlasapi.cloud_backup.SnapshotRestoreResponse (restore_id: str, de-
                                                    livery_type: atlas-
                                                    api.cloud_backup.DeliveryType,
                                                    snapshot_id: str, tar-
                                                    get_cluster_name: str, tar-
                                                    get_group_id: str, cancelled:
                                                    bool = False, created_at:
                                                    datetime.datetime = None,
                                                    expired: bool = False, ex-
                                                    pires_at: datetime.datetime
                                                    = None, finished_at: date-
                                                    time.datetime = None, links: list
                                                    = None, snapshot_timestamp:
                                                    datetime.datetime = None, tar-
                                                    get_deployment_item_name:
                                                    str = None, delivery_url: str =
                                                    None)

    Bases: atlasapi.cloud_backup.SnapshotRestore

    classmethod from_dict (data_dict)

class atlasapi.cloud_backup.SnapshotStatus
    Bases: enum.Enum

    An enumeration.

    COMPLETED = 'Completed'
    FAILED = 'Failed'
    INPROGRESS = 'In Progress'
    QUEUED = 'Queued'

class atlasapi.cloud_backup.SnapshotType
    Bases: enum.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.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'
```

```
HOURL = 'PT1H'
```

```
MINUTE = 'PT1M'
```

```
class atlasapi.lib.AtlasLogNames
```

Bases: enum.Enum

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

```
MONGODB = 'mongodb.gz'
```

```
MONGODB_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
    Bases: enum.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
    Bases: enum.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
    Bases: enum.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
    Bases: enum.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.datetime] = None, cluster_count: Optional[int] = None, id: Optional[str] = None, links: list = None, with_default_alert_settings: Optional[bool] = True, project_owner_id: str = None)
```

Bases: object

create_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: 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*)

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: atlasapi.clusters.InstanceSizeName = <InstanceSizeName.M10: 'M10'>, disk_size: int = 10, provider: atlasapi.lib.ProviderName = <ProviderName.AWS: 'Amazon Web Services'>, region: str = 'US_WEST_2', version: atlasapi.lib.MongoDBMajorVersion = <MongoDBMajorVersion.v4_0: '4.0'>*) → *atlasapi.clusters.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 Instance-SizeName 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: atlasapi.clusters.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
- type `areYouSure: bool`

get_all_clusters (*pageNum=1, itemsPerPage=1000, 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, atlasapi.clusters.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[atlasapi.clusters.ClusterConfig, atlasapi.clusters.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[atlasapi.clusters.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: atlasapi.clusters.AdvancedOptions, as_obj: bool = True*) → Union[atlasapi.clusters.AdvancedOptions, dict]

Modifies cluster advanced options using a *AdvancedOptions* object.

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

Parameters

- **cluster** (*str*) – The cluster 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: atlas-api.clusters.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: atlasapi.clusters.TLSProtocols, as_obj: bool = True*) → atlasapi.clusters.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 cluster. :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 = 1000, 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: atlasapi.specs.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=1000, 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*) → atlasapi.alerts.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 () → `atlasapi.maintenance_window.MaintenanceWindow`

The current Maintainable Window configuration.

Returns: `MaintainableWindow` object

defer () → dict

Defers the currently scheduled maintenance window.

Returns: bool:

set_config (*new_config: atlasapi.maintenance_window.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 `MaintenanceWindow` 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

cluster_list

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[atlasapi.specs.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 list of *Host* objects

Return type List[*Host*]

get_log_for_host (*host_obj*: *atlasapi.specs.Host*, *log_name*: *atlasapi.lib.AtlasLogNames* = *<AtlasLogNames.MONGODB: 'mongodb.gz'>*, *date_from*: *datetime.datetime* = *None*, *date_to*: *datetime.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*: *atlasapi.specs.Host*, *log_name*: *atlasapi.lib.AtlasLogNames* = *<AtlasLogNames.MONGODB: 'mongodb.gz'>*, *date_from*: *datetime.datetime* = *None*, *date_to*: *datetime.datetime* = *None*) → *Iterable[atlasapi.lib.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 retrieve 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 Yields *LogLine* objects, one for each logline found in the file.

Return type *Iterable[LogLine]*

get_logs_for_cluster (*cluster_name*: *str*, *log_name*: *atlasapi.lib.AtlasLogNames* = *<AtlasLogNames.MONGODB: 'mongodb.gz'>*, *date_from*: *datetime.datetime* = *None*, *date_to*: *datetime.datetime* = *None*) → *Iterable[atlasapi.specs.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: atlasapi.lib.AtlasLogNames = <AtlasLogNames.MONGODB: 'mongodb.gz'>, date_from: datetime.datetime = None, date_to: datetime.datetime = None*) → `Iterable[atlasapi.specs.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[atlasapi.lib.AtlasGranularities] = None, period: Optional[atlasapi.lib.AtlasPeriods] = None, measurement: Optional[atlasapi.specs.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[atlasapi.specs.Host]`

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

Returns An iterator of Host Objects.

Return type `Iterable[Host]`

host_list primaries

Yields only hosts which are currently primary.

host_list secondaries

Yields only hosts which are currently secondaries.

host_names

Returns a simple list of host names without port

Return type `Iterator[str]`

update_host_list (*host_obj: atlasapi.specs.Host*) → `None`

Places a host into the `host_list` property.

Parameters `host_obj` – Host: A host object with measurements.

Returns**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**all**

Returns all events for the current project/group.

Returns A list of event objects.**Return type** `ListOfEvents`**since** (*since_datetime: datetime.datetime*) → `NewType.<locals>.new_type`

Returns all events since the passed datetime. (UTC)

Returns**Return type** `ListOfEvents`

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[atlasapi.whitelist.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 = 1000, iterable: bool = False*) → Iterable[atlasapi.whitelist.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*) → atlasapi.whitelist.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: str = None, as_obj: bool = True) → Union[atlasapi.cloud_backup.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[atlasapi.cloud_backup.CloudBackupSnapshot], Iterable[dict]]
```

Get single 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[atlasapi.cloud_backup.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: str = None, as_obj: bool = True) → Union[List[Union[dict, atlasapi.cloud_backup.SnapshotRestoreResponse]], atlasapi.cloud_backup.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: atlasapi.cloud_backup.DeliveryType = <DeliveryType.automated: 'Automated restore to Atlas cluster'>, allow_same: bool = False) → atlasapi.cloud_backup.SnapshotRestoreResponse
```

```
request_snapshot_restore_to_group (source_cluster_name: str, snapshot_id: str, target_cluster_name: str, target_group_obj, delivery_type: atlasapi.cloud_backup.DeliveryType = <DeliveryType.automated: 'Automated restore to Atlas cluster'>) → atlasapi.cloud_backup.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: str = None*) → `Iterable[atlasapi.teams.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: str = None, flatten_teams: Optional[bool] = None, include_org_users: Optional[bool] = None*) → `Iterable[atlasapi.atlas_users.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*) → `atlasapi.projects.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*) → atlasapi.projects.Project

Return project by name

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

Returns (Project): A single Project

projects

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

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

settings

user_count (*group_id: 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
- **in this way.** (*instantiated*) –
- **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/>

The orgs resource provides access to manage Atlas organizations.

Parameters **atlas** (`Atlas`) – Atlas instance

count

Count of Organizations accessible by the authenticated user/key.

Returns (int):

organization_by_id (*org_id: str*) → atlasapi.organizations.Organization

Single organization searched by org_id.

Parameters **org_id** (*str*) –

Returns (Organization): a single Organization object.

organization_by_name (*org_name: str*) → atlasapi.organizations.Organization

Singe 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.

organizations

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.

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 4

Indices and tables

- `genindex`
- `modindex`
- `search`

a

- `atlasapi.alerts`, 5
- `atlasapi.atlas`, 3
- `atlasapi.atlas_types`, 5
- `atlasapi.cloud_backup`, 25
- `atlasapi.clusters`, 5
- `atlasapi.errors`, 10
- `atlasapi.events`, 10
- `atlasapi.lib`, 27
- `atlasapi.maintenance_window`, 24
- `atlasapi.network`, 14
- `atlasapi.organizations`, 30
- `atlasapi.projects`, 29
- `atlasapi.settings`, 15
- `atlasapi.specs`, 16
- `atlasapi.whitelist`, 10

A

- ACCEPTED (*atlasapi.settings.Settings* attribute), 15
- acknowledge_an_alert() (*atlasapi.atlas.Atlas._Alerts* method), 35
- acknowledge_an_alert_forever() (*atlasapi.atlas.Atlas._Alerts* method), 36
- add_log_file() (*atlasapi.specs.Host* method), 22
- add_measurements() (*atlasapi.specs.Host* method), 22
- add_role() (*atlasapi.specs.DatabaseUsersPermissionsSpecs* method), 21
- add_roles() (*atlasapi.specs.DatabaseUsersPermissionsSpecs* method), 21
- AdvancedOptions (*class in atlasapi.clusters*), 5
- Alert (*class in atlasapi.alerts*), 5
- AlertsGetAll (*class in atlasapi.atlas*), 3
- AlertStatusSpec (*class in atlasapi.specs*), 16
- all (*atlasapi.atlas.Atlas._Events* attribute), 40
- answer() (*atlasapi.network.Network* method), 14
- api_resources (*atlasapi.settings.Settings* attribute), 16
- as_create_dict() (*atlasapi.clusters.ClusterConfig* method), 6
- as_create_dict() (*atlasapi.clusters.ReplicationSpecs* method), 9
- as_dict (*atlasapi.cloud_backup.CloudBackupRequest* attribute), 25
- as_dict (*atlasapi.cloud_backup.SnapshotRestore* attribute), 26
- as_dict (*atlasapi.clusters.AdvancedOptions* attribute), 5
- as_dict (*atlasapi.specs.AtlasMeasurement* attribute), 17
- as_dict() (*atlasapi.clusters.AtlasBasicReplicaSet* method), 6
- as_dict() (*atlasapi.clusters.ClusterConfig* method), 7
- as_dict() (*atlasapi.clusters.ProviderSettings* method), 8
- as_dict() (*atlasapi.clusters.ReplicationSpecs* method), 9
- as_dict() (*atlasapi.clusters.ShardedClusterConfig* method), 9
- as_dict() (*atlasapi.maintenance_window.MaintenanceWindow* method), 24
- as_dict() (*atlasapi.specs.AtlasMeasurementValue* method), 21
- as_dict() (*atlasapi.whitelist.WhitelistEntry* method), 10
- as_modify_dict() (*atlasapi.clusters.ClusterConfig* method), 7
- as_tuple (*atlasapi.specs.AtlasMeasurementValue* attribute), 21
- as_update_dict() (*atlasapi.maintenance_window.MaintenanceWindow* method), 24
- Atlas (*class in atlasapi.atlas*), 3
- Atlas._Alerts (*class in atlasapi.atlas*), 35
- Atlas._CloudBackups (*class in atlasapi.atlas*), 41
- Atlas._Clusters (*class in atlasapi.atlas*), 31
- Atlas._DatabaseUsers (*class in atlasapi.atlas*), 34
- Atlas._Events (*class in atlasapi.atlas*), 40
- Atlas._Hosts (*class in atlasapi.atlas*), 37
- Atlas._MaintenanceWindows (*class in atlasapi.atlas*), 37
- Atlas._Organizations (*class in atlasapi.atlas*), 44
- Atlas._Projects (*class in atlasapi.atlas*), 43
- Atlas._Whitelist (*class in atlasapi.atlas*), 40
- atlas_event_factory() (*in module atlasapi.events*), 10
- atlasAdmin (*atlasapi.specs.RoleSpecs* attribute), 24
- atlasapi.alerts (*module*), 5
- atlasapi.atlas (*module*), 3
- atlasapi.atlas_types (*module*), 5
- atlasapi.cloud_backup (*module*), 25
- atlasapi.clusters (*module*), 5
- atlasapi.errors (*module*), 10
- atlasapi.events (*module*), 10
- atlasapi.lib (*module*), 27

`atlasapi.maintenance_window` (*module*), 24
`atlasapi.network` (*module*), 14
`atlasapi.organizations` (*module*), 30
`atlasapi.projects` (*module*), 29
`atlasapi.settings` (*module*), 15
`atlasapi.specs` (*module*), 16
`atlasapi.whitelist` (*module*), 10
`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*), 27
`AtlasHostEvent` (*class in atlasapi.events*), 10
`AtlasLogNames` (*class in atlasapi.lib*), 27
`AtlasMeasurement` (*class in atlasapi.specs*), 16
`AtlasMeasurementTypes` (*class in atlasapi.specs*), 17
`AtlasMeasurementTypes.Asserts` (*class in atlasapi.specs*), 17
`AtlasMeasurementTypes.Cache` (*class in atlasapi.specs*), 18
`AtlasMeasurementTypes.CPU` (*class in atlasapi.specs*), 18
`AtlasMeasurementTypes.CPU.Process` (*class in atlasapi.specs*), 18
`AtlasMeasurementTypes.CPU.ProcessNormalized` (*class in atlasapi.specs*), 18
`AtlasMeasurementTypes.CPU.System` (*class in atlasapi.specs*), 18
`AtlasMeasurementTypes.CPU.SystemNormalized` (*class in atlasapi.specs*), 18
`AtlasMeasurementTypes.Cursors` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.Db` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.DocumentMetrics` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.ExtraInfo` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.GlobalLockCurrentQueue` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.Memory` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.Network` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.Opcounter` (*class in atlasapi.specs*), 19
`AtlasMeasurementTypes.Opcounter.Repl` (*class in atlasapi.specs*), 20
`AtlasMeasurementTypes.Operations` (*class in atlasapi.specs*), 20
`AtlasMeasurementTypes.Operations.ExecutionTime` (*class in atlasapi.specs*), 20
`AtlasMeasurementTypes.Oplog` (*class in atlasapi.specs*), 20
`AtlasMeasurementTypes.QueryExecutor` (*class in atlasapi.specs*), 20
`AtlasMeasurementTypes.QueryTargetingScanned` (*class in atlasapi.specs*), 20
`AtlasMeasurementTypes.TicketsAvailable` (*class in atlasapi.specs*), 20
`AtlasMeasurementValue` (*class in atlasapi.specs*), 20
`AtlasPagination` (*class in atlasapi.atlas*), 4
`AtlasPeriods` (*class in atlasapi.lib*), 27
`AtlasUnits` (*class in atlasapi.lib*), 27
`automated` (*atlasapi.cloud_backup.DeliveryType attribute*), 26
`AWS` (*atlasapi.lib.ProviderName attribute*), 28
`AZURE` (*atlasapi.lib.ProviderName attribute*), 28

B

`backup` (*atlasapi.specs.RoleSpecs attribute*), 24
`BAD_REQUEST` (*atlasapi.settings.Settings attribute*), 15
`BASE_URL` (*atlasapi.settings.Settings attribute*), 15
`BYTES` (*atlasapi.lib.AtlasUnits attribute*), 28
`bytes_id` (*atlasapi.specs.AtlasMeasurementTypes.Network attribute*), 19
`bytes_in` (*atlasapi.specs.AtlasMeasurementTypes.Network attribute*), 19
`bytes_out` (*atlasapi.specs.AtlasMeasurementTypes.Network attribute*), 19
`BYTES_PER_SECOND` (*atlasapi.lib.AtlasUnits attribute*), 28
`bytes_read` (*atlasapi.specs.AtlasMeasurementTypes.Cache attribute*), 19
`bytes_written` (*atlasapi.specs.AtlasMeasurementTypes.Cache attribute*), 19

C

`cancel_snapshot_restore_request()` (*atlasapi.atlas.Atlas._CloudBackups method*), 41
`checkAndRaise()` (*atlasapi.errors.ErrPaginationLimits method*), 13
`children_kernel` (*atlasapi.specs.AtlasMeasurementTypes.CPU.Process attribute*), 18
`children_kernel` (*atlasapi.specs.AtlasMeasurementTypes.CPU.ProcessNormalized attribute*), 18
`children_user` (*atlasapi.specs.AtlasMeasurementTypes.CPU.Process*

attribute), 18
 children_user (atlasapi.specs.AtlasMeasurementTypes.CPU.ProcessNormalized attribute), 18
 clean_list() (in module atlasapi.specs), 24
 clear_roles() (atlasapi.specs.DatabaseUsersPermissionsSpecs method), 21
 CLOSED (atlasapi.specs.AlertStatusSpec attribute), 16
 CloudBackupRequest (class in atlasapi.cloud_backup), 25
 CloudBackupSnapshot (class in atlasapi.cloud_backup), 25
 CloudBackupSnapshotsGetAll (class in atlasapi.atlas), 4
 cluster_list (atlasapi.atlas.Atlas._Hosts attribute), 37
 ClusterConfig (class in atlasapi.clusters), 6
 clusterMonitor (atlasapi.specs.RoleSpecs attribute), 24
 ClustersGetAll (class in atlasapi.atlas), 4
 ClusterStates (class in atlasapi.clusters), 7
 ClusterType (class in atlasapi.lib), 28
 cmd (atlasapi.specs.AtlasMeasurementTypes.Opcounter attribute), 20
 cmd (atlasapi.specs.AtlasMeasurementTypes.Opcounter.Repl attribute), 20
 commands (atlasapi.specs.AtlasMeasurementTypes.Operations.ExecutionTime attribute), 20
 COMPLETED (atlasapi.cloud_backup.SnapshotStatus attribute), 26
 CONFLICT (atlasapi.settings.Settings attribute), 15
 connections (atlasapi.specs.AtlasMeasurementTypes attribute), 20
 count (atlasapi.atlas.Atlas._Organizations attribute), 44
 create_a_database_user() (atlasapi.atlas.Atlas._DatabaseUsers method), 34
 create_basic_rs() (atlasapi.atlas.Atlas._Clusters method), 31
 create_cluster() (atlasapi.atlas.Atlas._Clusters method), 31
 create_dict (atlasapi.projects.Project attribute), 29
 create_snapshot_for_cluster() (atlasapi.atlas.Atlas._CloudBackups method), 41
 create_whitelist_entry() (atlasapi.atlas.Atlas._Whitelist method), 40
 CREATED (atlasapi.settings.Settings attribute), 16
 CREATING (atlasapi.clusters.ClusterStates attribute), 7
 current_config() (atlasapi.atlas.Atlas._MaintenanceWindows method), 37

D

data_size (atlasapi.specs.AtlasMeasurementTypes.Db attribute), 19
 databaseName (atlasapi.settings.Settings attribute), 16
 DatabaseUsersGetAll (class in atlasapi.atlas), 4
 DatabaseUsersPermissionsSpecs (class in atlasapi.specs), 21
 DatabaseUsersUpdatePermissionsSpecs (class in atlasapi.specs), 22
 date_end (atlasapi.specs.AtlasMeasurement attribute), 17
 date_start (atlasapi.specs.AtlasMeasurement attribute), 17
 DAY (atlasapi.lib.AtlasGranularities attribute), 27
 dbAdmin (atlasapi.specs.RoleSpecs attribute), 24
 dbAdminAnyDatabase (atlasapi.specs.RoleSpecs attribute), 24
 defer() (atlasapi.atlas.Atlas._MaintenanceWindows method), 37
 delete (atlasapi.specs.AtlasMeasurementTypes.Opcounter attribute), 20
 delete (atlasapi.specs.AtlasMeasurementTypes.Opcounter.Repl attribute), 20
 delete() (atlasapi.network.Network method), 14
 delete_a_database_user() (atlasapi.atlas.Atlas._DatabaseUsers method), 34
 delete_a_whitelist_entry() (atlasapi.atlas.Atlas._Whitelist method), 40
 delete_cluster() (atlasapi.atlas.Atlas._Clusters method), 32
 DELETED (atlasapi.clusters.ClusterStates attribute), 7
 deleted (atlasapi.specs.AtlasMeasurementTypes.DocumentMetrics attribute), 19
 DELETING (atlasapi.clusters.ClusterStates attribute), 7
 DeliveryType (class in atlasapi.cloud_backup), 26
 dirty (atlasapi.specs.AtlasMeasurementTypes.Cache attribute), 19
 download (atlasapi.cloud_backup.DeliveryType attribute), 26

E

enableSharding (atlasapi.specs.RoleSpecs attribute), 24
 ERR_ITEMS_PER_PAGE (atlasapi.errors.ErrPaginationLimits attribute), 13
 ERR_PAGE_NUM (atlasapi.errors.ErrPaginationLimits attribute), 13
 ErrAtlasBackupError, 10
 ErrAtlasBadRequest, 11
 ErrAtlasConflict, 11
 ErrAtlasDuplicateClusterName, 11

ErrAtlasForbidden, 11
ErrAtlasForbiddenWL, 11
ErrAtlasGeneric, 11
ErrAtlasJobError, 12
ErrAtlasMethodNotAllowed, 12
ErrAtlasNotFound, 12
ErrAtlasRestoreConflictError, 12
ErrAtlasServerErrors, 13
ErrAtlasUnauthorized, 13
ErrConfirmationRequested, 13
ErrPagination, 13
ErrPaginationLimits, 13
ErrRole, 13
EventsGetForProject (class in *atlasapi.atlas*), 4

F

FAILED (*atlasapi.cloud_backup.SnapshotStatus* attribute), 26
FALLBACK (*atlasapi.cloud_backup.SnapshotType* attribute), 26
fetch() (*atlasapi.atlas.AlertsGetAll* method), 3
fetch() (*atlasapi.atlas.EventsGetForProject* method), 4
file_request_timeout (*atlasapi.settings.Settings* attribute), 16
fill_from_dict() (*atlasapi.clusters.AdvancedOptions* class method), 5
fill_from_dict() (*atlasapi.clusters.ClusterConfig* class method), 7
fill_from_dict() (*atlasapi.whitelist.WhitelistEntry* class method), 10
fill_host_list() (*atlasapi.atlas.Atlas._Hosts* method), 37
FIVE_MINUTE (*atlasapi.lib.AtlasGranularities* attribute), 27
for_create() (*atlasapi.projects.Project* class method), 29
FORBIDDEN (*atlasapi.settings.Settings* attribute), 16
FRIDAY (*atlasapi.maintenance_window.Weekdays* attribute), 25
from_dict() (*atlasapi.cloud_backup.CloudBackupSnapshot* class method), 25
from_dict() (*atlasapi.cloud_backup.SnapshotRestoreResponse* class method), 26
from_dict() (*atlasapi.clusters.ProviderSettings* class method), 8
from_dict() (*atlasapi.clusters.ReplicationSpecs* class method), 9
from_dict() (*atlasapi.maintenance_window.MaintenanceWindow*

class method), 24
from_dict() (*atlasapi.organizations.Organization* class method), 30
from_dict() (*atlasapi.projects.Project* class method), 29
from_dict() (*atlasapi.projects.ProjectSettings* class method), 29

G

GCP (*atlasapi.lib.ProviderName* attribute), 28
GEOSHARDED (*atlasapi.lib.ClusterType* attribute), 28
get() (*atlasapi.network.Network* method), 14
get_a_single_database_user() (*atlasapi.atlas.Atlas._DatabaseUsers* method), 35
get_all_alerts() (*atlasapi.atlas.Atlas._Alerts* method), 36
get_all_clusters() (*atlasapi.atlas.Atlas._Clusters* method), 32
get_all_database_users() (*atlasapi.atlas.Atlas._DatabaseUsers* method), 35
get_all_whitelist_entries() (*atlasapi.atlas.Atlas._Whitelist* method), 41
get_an_alert() (*atlasapi.atlas.Atlas._Alerts* method), 36
get_backup_snapshot_for_cluster() (*atlasapi.atlas.Atlas._CloudBackups* method), 42
get_backup_snapshots_for_cluster() (*atlasapi.atlas.Atlas._CloudBackups* method), 42
get_file() (*atlasapi.network.Network* method), 15
get_log_for_host() (*atlasapi.atlas.Atlas._Hosts* method), 38
get_loglines_for_host() (*atlasapi.atlas.Atlas._Hosts* method), 38
get_logs_for_cluster() (*atlasapi.atlas.Atlas._Hosts* method), 38
get_logs_for_project() (*atlasapi.atlas.Atlas._Hosts* method), 39
get_measurement_for_host() (*atlasapi.specs.Host* method), 22
get_measurement_for_hosts() (*atlasapi.atlas.Atlas._Hosts* method), 39
get_project_teams() (*atlasapi.atlas.Atlas._Projects* method), 43
get_project_users() (*atlasapi.atlas.Atlas._Projects* method), 43
get_single_cluster() (*atlasapi.atlas.Atlas._Clusters* method), 32
get_single_cluster_advanced_options() (*atlasapi.atlas.Atlas._Clusters* method), 32
get_single_cluster_as_obj() (*atlasapi.atlas.Atlas._Clusters* method), 33

[get_snapshot_restore_requests\(\)](#) ([atlasapi.atlas.Atlas._CloudBackups](#) method), 42
[get_whitelist_entry\(\)](#) ([atlasapi.atlas.Atlas._Whitelist](#) method), 41
[getAtlasResponse\(\)](#) ([atlasapi.errors.ErrAtlasGeneric](#) method), 12
[getmore\(\)](#) ([atlasapi.specs.AtlasMeasurementTypes.Opcounter](#) attribute), 20
[getSpecs\(\)](#) ([atlasapi.specs.DatabaseUsersPermissionsSpecs](#) method), 21
[getSpecs\(\)](#) ([atlasapi.specs.DatabaseUsersUpdatePermissionsSpecs](#) method), 22
[GIGABYTES](#) ([atlasapi.lib.AtlasUnits](#) attribute), 28
[GIGABYTES_PER_HOUR](#) ([atlasapi.lib.AtlasUnits](#) attribute), 28
[guest](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.System](#) attribute), 18
[guest](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.SystemNormalized](#) attribute), 18
H
[Host](#) (class in [atlasapi.specs](#)), 22
[host_list_by_cluster\(\)](#) ([atlasapi.atlas.Atlas._Hosts](#) method), 39
[host_list primaries](#) ([atlasapi.atlas.Atlas._Hosts](#) attribute), 39
[host_list secondaries](#) ([atlasapi.atlas.Atlas._Hosts](#) attribute), 39
[host_names](#) ([atlasapi.atlas.Atlas._Hosts](#) attribute), 39
[HostLogFile](#) (class in [atlasapi.specs](#)), 23
[HostsGetAll](#) (class in [atlasapi.atlas](#)), 4
[HOUR](#) ([atlasapi.lib.AtlasGranularities](#) attribute), 27
[HOURS_1](#) ([atlasapi.lib.AtlasPeriods](#) attribute), 27
[HOURS_24](#) ([atlasapi.lib.AtlasPeriods](#) attribute), 27
[HOURS_48](#) ([atlasapi.lib.AtlasPeriods](#) attribute), 27
[HOURS_8](#) ([atlasapi.lib.AtlasPeriods](#) attribute), 27
I
[IAMType](#) (class in [atlasapi.specs](#)), 23
[IDLE](#) ([atlasapi.clusters.ClusterStates](#) attribute), 7
[INPROGRESS](#) ([atlasapi.cloud_backup.SnapshotStatus](#) attribute), 26
[insert](#) ([atlasapi.specs.AtlasMeasurementTypes.Opcounter](#) attribute), 20
[insert](#) ([atlasapi.specs.AtlasMeasurementTypes.Opcounter.Repl](#) attribute), 20
[inserted](#) ([atlasapi.specs.AtlasMeasurementTypes.DocumentMetric](#) attribute), 19
[InstanceSizeName](#) (class in [atlasapi.clusters](#)), 7
[iowait](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.System](#) attribute), 18
[iowait](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.SystemNormalized](#) attribute), 18
[irq](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.System](#) attribute), 18
[irq](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.SystemNormalized](#) attribute), 18
[is_existing_cluster\(\)](#) ([atlasapi.atlas.Atlas._Clusters](#) method), 33
[is_existing_snapshot\(\)](#) ([atlasapi.atlas.Atlas._CloudBackups](#) method), 42
[itemsPerPage](#) ([atlasapi.settings.Settings](#) attribute), 16
[itemsPerPageMax](#) ([atlasapi.settings.Settings](#) attribute), 16
[itemsPerPageMin](#) ([atlasapi.settings.Settings](#) attribute), 16
K
[kernel](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.ProcessNormalized](#) attribute), 18
[kernel](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.System](#) attribute), 18
[kernel](#) ([atlasapi.specs.AtlasMeasurementTypes.CPU.SystemNormalized](#) attribute), 18
L
[LogLine](#) (class in [atlasapi.lib](#)), 28
M
[M0](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M10](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M100](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M140](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M2](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M20](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M200](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M200_NVME](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M30](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M300](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M40](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M400](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M400_NVME](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 7
[M40_NVME](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 8
[M5](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 8
[M50](#) ([atlasapi.clusters.InstanceSizeName](#) attribute), 8
[M500_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*), 24
mapped (*atlasapi.specs.AtlasMeasurementTypes.Memory* attribute), 19
master_time (*atlasapi.specs.AtlasMeasurementTypes.Oplog* attribute), 20
measurement_stats (*atlasapi.specs.AtlasMeasurement* attribute), 17
measurement_stats_friendly_bytes (*atlasapi.specs.AtlasMeasurement* attribute), 17
measurement_stats_friendly_number (*atlasapi.specs.AtlasMeasurement* attribute), 17
measurements (*atlasapi.specs.AtlasMeasurement* attribute), 17
measurements_as_tuples() (*atlasapi.specs.AtlasMeasurement* method), 17
measurements_count (*atlasapi.specs.AtlasMeasurement* attribute), 17
MEGABYTES_PER_SECOND (*atlasapi.lib.AtlasUnits* attribute), 28
METHOD_NOT_ALLOWED (*atlasapi.settings.Settings* attribute), 16
MILLISECONDS (*atlasapi.lib.AtlasUnits* attribute), 28
MINUTE (*atlasapi.lib.AtlasGranularities* attribute), 27
MINUTES_15 (*atlasapi.lib.AtlasPeriods* attribute), 27
modify_cluster() (*atlasapi.atlas.Atlas._Clusters* method), 33
modify_cluster_advanced_options() (*atlasapi.atlas.Atlas._Clusters* method), 33
modify_cluster_instance_size() (*atlasapi.atlas.Atlas._Clusters* method), 33
modify_cluster_tls() (*atlasapi.atlas.Atlas._Clusters* method), 34
MONDAY (*atlasapi.maintenance_window.Weekdays* attribute), 25
MONGODB_AUDIT (*atlasapi.lib.AtlasLogNames* attribute), 27
MONGODB (*atlasapi.lib.AtlasLogNames* attribute), 27
MongoDBMajorVersion (class in *atlasapi.lib*), 28
MONGOS (*atlasapi.lib.AtlasLogNames* attribute), 27
MONGOS_AUDIT (*atlasapi.lib.AtlasLogNames* attribute), 27
MONTHS_1 (*atlasapi.lib.AtlasPeriods* attribute), 27
MONTHS_2 (*atlasapi.lib.AtlasPeriods* attribute), 27
msg (*atlasapi.specs.AtlasMeasurementTypes.Asserts* attribute), 17

N

Network (class in *atlasapi.network*), 14

nice (*atlasapi.specs.AtlasMeasurementTypes.CPU.System* attribute), 18
nice (*atlasapi.specs.AtlasMeasurementTypes.CPU.SystemNormalized* attribute), 18
NO_CONTENT (*atlasapi.settings.Settings* attribute), 16
NO_DATA (*atlasapi.specs.ReplicaSetTypes* attribute), 23
NONE (*atlasapi.specs.IAMType* attribute), 23
NOTFOUND (*atlasapi.settings.Settings* attribute), 16
num_requests (*atlasapi.specs.AtlasMeasurementTypes.Network* attribute), 19

O

objects_per_returned (*atlasapi.specs.AtlasMeasurementTypes.QueryTargetingScanned* attribute), 20
ONDEMAND (*atlasapi.cloud_backup.SnapshotType* attribute), 26
OPEN (*atlasapi.specs.AlertStatusSpec* attribute), 16
open (*atlasapi.specs.AtlasMeasurementTypes.Cursors* attribute), 19
Organization (class in *atlasapi.organizations*), 30
organization_by_id() (*atlasapi.atlas.Atlas._Organizations* method), 44
organization_by_name() (*atlasapi.atlas.Atlas._Organizations* method), 44
organizations (*atlasapi.atlas.Atlas._Organizations* attribute), 45

P

page_faults (*atlasapi.specs.AtlasMeasurementTypes.ExtraInfo* attribute), 19
pageNum (*atlasapi.settings.Settings* attribute), 16
patch() (*atlasapi.network.Network* method), 15
pause_cluster() (*atlasapi.atlas.Atlas._Clusters* method), 34
per_returned (*atlasapi.specs.AtlasMeasurementTypes.QueryTargetingScanned* attribute), 20
PERCENT (*atlasapi.lib.AtlasUnits* attribute), 28
pointInTime (*atlasapi.cloud_backup.DeliveryType* attribute), 26
post() (*atlasapi.network.Network* method), 15
Project (class in *atlasapi.projects*), 29
project_by_id() (*atlasapi.atlas.Atlas._Projects* method), 43
project_by_name() (*atlasapi.atlas.Atlas._Projects* method), 44
projects (*atlasapi.atlas.Atlas._Projects* attribute), 44
ProjectSettings (class in *atlasapi.projects*), 29
ProviderName (class in *atlasapi.lib*), 28

ProviderSettings (class in atlasapi.clusters), 8

PROVISIONED (atlasapi.clusters.VolumeTypes attribute), 9

Q

query (atlasapi.specs.AtlasMeasurementTypes.Opcounter attribute), 20

QUEUED (atlasapi.cloud_backup.SnapshotStatus attribute), 26

R

R200 (atlasapi.clusters.InstanceSizeName attribute), 8

R300 (atlasapi.clusters.InstanceSizeName attribute), 8

R40 (atlasapi.clusters.InstanceSizeName attribute), 8

R400 (atlasapi.clusters.InstanceSizeName attribute), 8

R50 (atlasapi.clusters.InstanceSizeName attribute), 8

R60 (atlasapi.clusters.InstanceSizeName attribute), 8

R700 (atlasapi.clusters.InstanceSizeName attribute), 8

R80 (atlasapi.clusters.InstanceSizeName attribute), 8
rate (atlasapi.specs.AtlasMeasurementTypes.Oplog attribute), 20

read (atlasapi.specs.RoleSpecs attribute), 24

readAnyDatabase (atlasapi.specs.RoleSpecs attribute), 24

readers (atlasapi.specs.AtlasMeasurementTypes.GlobalLockCurrentOwner attribute), 19

reads (atlasapi.specs.AtlasMeasurementTypes.Operations.ExecutionTime attribute), 20

reads (atlasapi.specs.AtlasMeasurementTypes.TicketsAvailable attribute), 20

readWrite (atlasapi.specs.RoleSpecs attribute), 24

readWriteAnyDatabase (atlasapi.specs.RoleSpecs attribute), 24

RECOVERING (atlasapi.specs.ReplicaSetTypes attribute), 23

RegionConfig (class in atlasapi.clusters), 8

regular (atlasapi.specs.AtlasMeasurementTypes.Asserts attribute), 17

remove_role() (atlasapi.specs.DatabaseUsersPermissionsSpecs method), 22

remove_roles() (atlasapi.specs.DatabaseUsersPermissionsSpecs method), 22

REPAIRING (atlasapi.clusters.ClusterStates attribute), 7

REPLICA_PRIMARY (atlasapi.specs.ReplicaSetTypes attribute), 23

REPLICA_SECONDARY (atlasapi.specs.ReplicaSetTypes attribute), 23

REPLICASET (atlasapi.lib.ClusterType attribute), 28

ReplicaSetTypes (class in atlasapi.specs), 23

ReplicationSpecs (class in atlasapi.clusters), 8

request_snapshot_restore() (atlasapi.atlas.Atlas._CloudBackups method),

42

request_snapshot_restore_to_group() (atlasapi.atlas.Atlas._CloudBackups method), 42

requests_timeout (atlasapi.settings.Settings attribute), 16

resident (atlasapi.specs.AtlasMeasurementTypes.Memory attribute), 19

returned (atlasapi.specs.AtlasMeasurementTypes.DocumentMetrics attribute), 19

ROLE (atlasapi.specs.IAMType attribute), 23

RoleSpecs (class in atlasapi.specs), 24

S

SATURDAY (atlasapi.maintenance_window.Weekdays attribute), 25

SCALAR (atlasapi.lib.AtlasUnits attribute), 28

SCALAR_PER_SECOND (atlasapi.lib.AtlasUnits attribute), 28

scan_and_order (atlasapi.specs.AtlasMeasurementTypes.Operations attribute), 20

scanned (atlasapi.specs.AtlasMeasurementTypes.QueryExecutor attribute), 20

scanned_objects (atlasapi.specs.AtlasMeasurementTypes.QueryExecutor attribute), 20

SCHEDULED (atlasapi.cloud_backup.SnapshotType attribute), 26

SERVER_ERRORS (atlasapi.settings.Settings attribute), 16

set_config() (atlasapi.atlas.Atlas._MaintenanceWindows method), 37

settings (atlasapi.atlas.Atlas._Projects attribute), 44

Settings (class in atlasapi.settings), 15

SHARD_CONFIG (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARD_CONFIG_PRIMARY (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARD_CONFIG_SECONDARY (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARD_MONGOS (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARD_PRIMARY (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARD_SECONDARY (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARD_STANDALONE (atlasapi.specs.ReplicaSetTypes attribute), 23

SHARDED (atlasapi.lib.ClusterType attribute), 28

SHARDEDCLUSTER (atlasapi.lib.ClusterType attribute), 28

ShardedClusterConfig (class in atlasapi.clusters), 9

- since() (*atlasapi.atlas.Atlas._Events method*), 40
- SnapshotRestore (*class in atlasapi.cloud_backup*), 26
- SnapshotRestoreResponse (*class in atlasapi.cloud_backup*), 26
- SnapshotStatus (*class in atlasapi.cloud_backup*), 26
- SnapshotType (*class in atlasapi.cloud_backup*), 26
- softirq (*atlasapi.specs.AtlasMeasurementTypes.CPU.System attribute*), 18
- softirq (*atlasapi.specs.AtlasMeasurementTypes.CPU.System attribute*), 18
- STANDARD (*atlasapi.clusters.VolumeTypes attribute*), 9
- StatisticalValues (*class in atlasapi.specs*), 24
- StatisticalValuesFriendly (*class in atlasapi.specs*), 24
- steal (*atlasapi.specs.AtlasMeasurementTypes.CPU.System attribute*), 18
- steal (*atlasapi.specs.AtlasMeasurementTypes.CPU.System attribute*), 18
- storage (*atlasapi.specs.AtlasMeasurementTypes.Db attribute*), 19
- SUCCESS (*atlasapi.settings.Settings attribute*), 16
- SUNDAY (*atlasapi.maintenance_window.Weekdays attribute*), 25
- ## T
- TENANT (*atlasapi.lib.ProviderName attribute*), 28
- test_failover() (*atlasapi.atlas.Atlas._Clusters method*), 34
- THURSDAY (*atlasapi.maintenance_window.Weekdays attribute*), 25
- timed_out (*atlasapi.specs.AtlasMeasurementTypes.Cursor attribute*), 19
- TLS1_0 (*atlasapi.clusters.TLSProtocols attribute*), 9
- TLS1_1 (*atlasapi.clusters.TLSProtocols attribute*), 9
- TLS1_2 (*atlasapi.clusters.TLSProtocols attribute*), 9
- TLS1_3 (*atlasapi.clusters.TLSProtocols attribute*), 9
- TLSProtocols (*class in atlasapi.clusters*), 9
- total (*atlasapi.specs.AtlasMeasurementTypes.GlobalLock attribute*), 19
- TRACKING (*atlasapi.specs.AlertStatusSpec attribute*), 16
- try_bool() (*in module atlasapi.cloud_backup*), 27
- try_date() (*in module atlasapi.cloud_backup*), 27
- TUESDAY (*atlasapi.maintenance_window.Weekdays attribute*), 25
- ## U
- unacknowledge_an_alert() (*atlasapi.atlas.Atlas._Alerts method*), 36
- UNAUTHORIZED (*atlasapi.settings.Settings attribute*), 16
- UNKNOWN (*atlasapi.clusters.ClusterStates attribute*), 7
- update (*atlasapi.specs.AtlasMeasurementTypes.Opcounter.Repl attribute*), 20
- update (*atlasapi.specs.AtlasMeasurementTypes.Opcounter.Repl attribute*), 20
- update_a_database_user() (*atlasapi.atlas.Atlas._DatabaseUsers method*), 35
- update_host_list() (*atlasapi.atlas.Atlas._Hosts method*), 39
- updated (*atlasapi.specs.AtlasMeasurementTypes.DocumentMetrics attribute*), 19
- uri_normalized (*atlasapi.clusters.ClusterStates attribute*), 7
- URI_STUB (*atlasapi.settings.Settings attribute*), 16
- used (*atlasapi.specs.AtlasMeasurementTypes.Cache attribute*), 19
- user (*atlasapi.specs.AtlasMeasurementTypes.Asserts attribute*), 18
- user (*atlasapi.specs.AtlasMeasurementTypes.CPU.Process attribute*), 18
- user (*atlasapi.specs.AtlasMeasurementTypes.CPU.ProcessNormalized attribute*), 18
- user (*atlasapi.specs.AtlasMeasurementTypes.CPU.System attribute*), 18
- user (*atlasapi.specs.AtlasMeasurementTypes.CPU.SystemNormalized attribute*), 18
- USER (*atlasapi.specs.IAMType attribute*), 23
- user_count() (*atlasapi.atlas.Atlas._Projects method*), 44
- ## V
- v3_4 (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- v3_6 (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- v4_0 (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- v4_2 (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- v4_4 (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- v5_0 (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- value_float (*atlasapi.specs.AtlasMeasurementValue attribute*), 21
- value_int (*atlasapi.specs.AtlasMeasurementValue attribute*), 21
- CurrentQueue (*atlasapi.specs.AtlasMeasurementTypes.Memory attribute*), 19
- VolumeTypes (*class in atlasapi.clusters*), 9
- vX_x (*atlasapi.lib.MongoDBMajorVersion attribute*), 28
- ## W
- warning (*atlasapi.specs.AtlasMeasurementTypes.Asserts attribute*), 18
- WEDNESDAY (*atlasapi.maintenance_window.Weekdays attribute*), 25
- Weekdays (*class in atlasapi.maintenance_window*), 25
- WEEKS_1 (*atlasapi.lib.AtlasPeriods attribute*), 27
- WEEKS_4 (*atlasapi.lib.AtlasPeriods attribute*), 27
- WhitelistEntry (*class in atlasapi.whitelist*), 10
- WhitelistGetAll (*class in atlasapi.atlas*), 4

writers (*atlasapi.specs.AtlasMeasurementTypes.GlobalLockCurrentQueue attribute*), [19](#)

writes (*atlasapi.specs.AtlasMeasurementTypes.Operations.ExecutionTime attribute*), [20](#)

writes (*atlasapi.specs.AtlasMeasurementTypes.TicketsAvailable attribute*), [20](#)

Y

YEARS_1 (*atlasapi.lib.AtlasPeriods attribute*), [27](#)

YEARS_2 (*atlasapi.lib.AtlasPeriods attribute*), [27](#)