ADMIN OPERATIONS

An admin API request will be done on a URI that starts with the configurable ‘admin’ resource entry point. Authorization for the admin API duplicates the S3 authorization mechanism. Some operations require that the user holds special administrative capabilities. The response entity type (XML or JSON) may be specified as the ‘format’ option in the request and defaults to JSON if not specified.

GET OBJECT

Get an existing object. NOTE: Does not require owner to be non-suspended.

SYNTAX

```
GET /{admin}/bucket?object&format=json HTTP/1.1
Host {fqdn}
```

REQUEST PARAMETERS

`bucket`

Description: The bucket containing the object to be retrieved.
Type: String
Example: `foo_bucket`
Required: Yes

`object`

Description: The object to be retrieved.
Type: String
Example: `foo.txt`
Required: Yes

RESPONSE ENTITIES

If successful, returns the desired object.

`object`

Description: The desired object.
Type: Object

SPECIAL ERROR RESPONSES

`NoSuchObject`

Description: Specified object does not exist.
Code: 404 Not Found

HEAD OBJECT

Verify the existence of an object. If the object exists, metadata headers for the object will be returned.

SYNTAX

```
HEAD /{admin}/bucket?object HTTP/1.1
```

```
Host {fqdn}
```

REQUEST PARAMETERS

`bucket`

Description: The bucket containing the object to be retrieved.
Type: String
Example: `foo_bucket`
Required: Yes

`object`

Description: The object to be retrieved.
Type: String
Example: `foo.txt`
Required: Yes

RESPONSE ENTITIES

None.

SPECIAL ERROR RESPONSES

`NoSuchObject`

Description: Specified object does not exist.
Code: 404 Not Found

GET ZONE INFO

Get cluster information.

SYNTAX

```
GET /{admin}/zone&format=json HTTP/1.1
Host {fqdn}
```

RESPONSE ENTITIES

If successful, returns cluster pool configuration.

`zone`

Description: Contains current cluster pool configuration.
Type: Container

`domain_root`

Description: root of all buckets.
Type: String
Parent: `cluster`

`control_pool`

Description:
Type: String
Parent: `cluster`

`gc_pool`

Description: Garbage collection pool.
Type: String
Parent: `cluster`

`log_pool`

Description: Log pool.
Type: String
Parent: `cluster`

`intent_log_pool`

Description: Intent log pool.
Type: String
Parent: `cluster`

`usage_log_pool`

Description: Usage log pool.
Type: String
Parent: `cluster`

`user_keys_pool`

Description: User key pool.
Type: String
Parent: `cluster`

`user_email_pool`

Description: User email pool.
Type: String
Parent: `cluster`

`user_swift_pool`

Description: Pool of swift users.
Type: String
Parent: `cluster`

SPECIAL ERROR RESPONSES

None.

EXAMPLE RESPONSE

```
HTTP/1.1 200
Content-Type: application/json

{
  "domain_root": ".rgw",
  "control_pool": ".rgw.control",
  "gc_pool": ".rgw.gc",
  "log_pool": ".log",
  "intent_log_pool": ".intent-log",
  "usage_log_pool": ".usage",
  "user_keys_pool": ".users",
  "user_email_pool": ".users.email",
  "user_swift_pool": ".users.swift",
  "user_uid_pool ": ".users.uid"
}
```

ADD PLACEMENT POOL

Make a pool available for data placement.

SYNTAX

```
PUT /{admin}/pool?format=json HTTP/1.1
Host {fqdn}
```

REQUEST PARAMETERS

`pool`

Description: The pool to be made available for data placement.
Type: String
Example: `foo_pool`
Required: Yes

`create`

Description: Creates the data pool if it does not exist.
Type: Boolean
Example: False [False]
Required: No

RESPONSE ENTITIES

TBD.

SPECIAL ERROR RESPONSES

TBD.

REMOVE PLACEMENT POOL

Make a pool unavailable for data placement.

SYNTAX

```
DELETE /{admin}/pool?format=json HTTP/1.1
Host {fqdn}
```

REQUEST PARAMETERS

`pool`

Description: The existing pool to be made available for data placement.
Type: String
Example: `foo_pool`
Required: Yes

`destroy`

Description: Destroys the pool after removing it from the active set.
Type: Boolean
Example: False [False]

Required: No

RESPONSE ENTITIES

TBD.

SPECIAL ERROR RESPONSES

TBD.

LIST AVAILABLE DATA PLACEMENT POOLS

List current pools available for data placement.

SYNTAX

```
GET /{admin}/pool?format=json HTTP/1.1
Host {fqdn}
```

RESPONSE ENTITIES

If successful, returns a list of pools available for data placement.

`pools`

Description: Contains currently available pools for data placement.
Type: Container

LIST EXPIRED GARBAGE COLLECTION ITEMS

List objects scheduled for garbage collection.

SYNTAX

```
GET /{admin}/garbage?format=json HTTP/1.1
Host {fqdn}
```

REQUEST PARAMETERS

None.

RESPONSE ENTITIES

If expired garbage collection items exist, a list of such objects will be returned.

`garbage`

Description: Expired garbage collection items.
Type: Container

`object`

Description: A container garbage collection object information.
Type: Container
Parent: `garbage`

`name`

Description: The name of the object.
Type: String
Parent: `object`

`expired`

Description: The date at which the object expired.
Type: String
Parent: `object`

SPECIAL ERROR RESPONSES

TBD.

MANUALLY PROCESSES GARBAGE COLLECTION ITEMS

List objects scheduled for garbage collection.

SYNTAX

```
DELETE /{admin}/garbage?format=json HTTP/1.1
Host {fqdn}
```

REQUEST PARAMETERS

None.

RESPONSE ENTITIES

If expired garbage collection items exist, a list of removed objects will be returned.

`garbage`

Description: Expired garbage collection items.
Type: Container

`object`

Description: A container garbage collection object information.
Type: Container
Parent: `garbage`

`name`

Description: The name of the object.
Type: String
Parent: `object`

`expired`

Description: The date at which the object expired.
Type: String
Parent: `object`

SPECIAL ERROR RESPONSES

TBD.

SHOW LOG OBJECTS

Show log objects

SYNTAX

```
GET /{admin}/log?format=json HTTP/1.1
Host {fqdn}
```

REQUEST PARAMETERS

`object`

Description: The log object to return.
Type: String:
Example: `2012-10-11-09-4165.2-foo_bucket`
Required: No

RESPONSE ENTITIES

If no object is specified, returns the full list of log objects.

`log-objects`

Description: A list of log objects.
Type: Container

`object`

Description: The name of the log object.
Type: String

`log`

Description: The contents of the log object.
Type: Container

SPECIAL ERROR RESPONSES

None.

STANDARD ERROR RESPONSES

`AccessDenied`

Description: Access denied.
Code: 403 Forbidden

`InternalError`

Description: Internal server error.
Code: 500 Internal Server Error

`NoSuchUser`

Description: User does not exist.
Code: 404 Not Found

`NoSuchBucket`

Description: Bucket does not exist.
Code: 404 Not Found

`NoSuchKey`

Description: No such access key.
Code: 404 Not Found