rabbitman Documentation

Release 0.1

David Szotten

February 25, 2015

Contents

Python client for the RabbitMQ management api.

CHAPTER 1

API

class rabbitman.**Client**(*url*, *username*, *password*)

RabbitMQ Management plugin api

Usage:

```
>>> client = Client('http://localhost:15672', 'guest', 'guest')
>>> client.get_vhosts()
```

create_cluster_name()

Name identifying this RabbitMQ cluster.

create_exchanges_by_vhost_and_name(*vhost*, *name*)

An individual exchange. To PUT an exchange, you will need a body looking something like this:

```
{
    "auto_delete": False,
    "internal": False,
    "type": "direct",
    "durable": True,
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

create_parameters_by_component_and_vhost_and_name(*component*, *vhost*, *name*)

An individual parameter. To PUT a parameter, you will need a body looking something like this:

```
{
    "vhost": "/",
    "component": "federation",
    "name": "local_username",
    "value": "guest"
}
```

Parameters

- **component** (*str*) –
- **vhost** (*str*) –
- **name** (*str*) –

create_permissions_by_vhost_and_user(*vhost*, *user*)
An individual permission of a user and virtual host. To PUT a permission, you will need a body looking something like this:

```
{
    "write": ".*",
    "read": ".*",
    "configure": ".*"
}
```

Parameters

- **vhost** (*str*) –
- **user** (*str*) –

create_policies_by_vhost_and_name(*vhost*, *name*)
An individual policy. To PUT a policy, you will need a body looking something like this:

```
{
    "priority": 0,
    "pattern": "^amq.",
    "apply-to": "all",
    "definition": {
        "federation-upstream-set": "all"
    }
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

create_queues_by_vhost_and_name(*vhost*, *name*)
An individual queue. To PUT a queue, you will need a body looking something like this:

```
{
    "node": "rabbit@smacmullen",
    "auto_delete": False,
    "durable": True,
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

create_users_by_name(*name*)
An individual user. To PUT a user, you will need a body looking something like this:

```
{
    "password": "secret",
    "tags": "administrator"
}
```

Parameters **name** (*str*) –

create_vhosts_by_name(*name*)
An individual virtual host. As a virtual host usually only has a name, you do not need an HTTP body when PUTing one of these. To enable / disable tracing, provide a body looking like:

```
{
    "tracing": True
}
```

Parameters **name** (*str*) –

delete_bindings_by_vhost_and_exchange_and_queue_and_props(*vhost*, *exchange*, *queue*, *props*)
An individual binding between an exchange and a queue. The props

Parameters

- **vhost** (*str*) –
- **exchange** (*str*) –
- **queue** (*str*) –
- **props** (*str*) –

delete_bindings_by_vhost_and_source_and_destination_and_props(*vhost*, *source*, *destination*, *props*)
An individual binding between two exchanges. Similar to the individual binding between an exchange and a queue, above.

Parameters

- **vhost** (*str*) –
- **source** (*str*) –
- **destination** (*str*) –
- **props** (*str*) –

delete_connections_by_name(*name*)
An individual connection. DELETEing it will close the connection. Optionally set the "X-Reason" header when DELETEing to provide a reason.

Parameters **name** (*str*) –

delete_exchanges_by_vhost_and_name(*vhost*, *name*)
An individual exchange. To PUT an exchange, you will need a body looking something like this:

```
{
    "auto_delete": False,
    "internal": False,
    "type": "direct",
    "durable": True,
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

delete_parameters_by_component_and_vhost_and_name(*component*, *vhost*, *name*)
An individual parameter. To PUT a parameter, you will need a body looking something like this:

```
{
    "vhost": "/",
    "component": "federation",
    "name": "local_username",
    "value": "guest"
}
```

Parameters

- **component** (*str*) –
- **vhost** (*str*) –
- **name** (*str*) –

delete_permissions_by_vhost_and_user(*vhost*, *user*)
An individual permission of a user and virtual host. To PUT a permission, you will need a body looking something like this:

```
{
    "write": ".*",
    "read": ".*",
    "configure": ".*"
}
```

Parameters

- **vhost** (*str*) –
- **user** (*str*) –

delete_policies_by_vhost_and_name(*vhost*, *name*)
An individual policy. To PUT a policy, you will need a body looking something like this:

```
{
    "priority": 0,
    "pattern": "^amq.",
    "apply-to": "all",
    "definition": {
        "federation-upstream-set": "all"
    }
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

delete_queues_by_vhost_and_name(*vhost*, *name*)
An individual queue. To PUT a queue, you will need a body looking something like this:

```
{
    "node": "rabbit@smacmullen",
    "auto_delete": False,
    "durable": True,
```

```
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

delete_queues_contents_by_vhost_and_name (*vhost*, *name*)
Contents of a queue. DELETE to purge. Note you can't GET this.

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

delete_users_by_name (*name*)
An individual user. To PUT a user, you will need a body looking something like this:

```
{
    "password": "secret",
    "tags": "administrator"
}
```

Parameters **name** (*str*) –

delete_vhosts_by_name (*name*)
An individual virtual host. As a virtual host usually only has a name, you do not need an HTTP body when PUTing one of these. To enable / disable tracing, provide a body looking like:

```
{
    "tracing": True
}
```

Parameters **name** (*str*) –

get_aliveness_test_by_vhost (*vhost*)
Declares a test queue, then publishes and consumes a message. Intended for use by monitoring tools. If everything is working correctly, will return HTTP status 200 with body:

```
{
    "status": "ok"
}
```

Parameters **vhost** (*str*) –

get_bindings ()
A list of all bindings.

get_bindings_by_vhost (*vhost*)
A list of all bindings in a given virtual host.

Parameters **vhost** (*str*) –

get_bindings_by_vhost_and_exchange_and_queue (*vhost*, *exchange*, *queue*)
A list of all bindings between an exchange and a queue. Remember, an exchange and a queue can be

bound together many times! To create a new binding, POST to this URI. You will need a body looking something like this:

```
{
    "routing_key": "my_routing_key",
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **exchange** (*str*) –
- **queue** (*str*) –

get_bindings_by_vhost_and_exchange_and_queue_and_props(*vhost*, *exchange*, *queue*, *props*)

An individual binding between an exchange and a queue. The props

Parameters

- **vhost** (*str*) –
- **exchange** (*str*) –
- **queue** (*str*) –
- **props** (*str*) –

get_bindings_by_vhost_and_source_and_destination(*vhost*, *source*, *destination*)

A list of all bindings between two exchanges. Similar to the list of all bindings between an exchange and a queue, above.

Parameters

- **vhost** (*str*) –
- **source** (*str*) –
- **destination** (*str*) –

get_bindings_by_vhost_and_source_and_destination_and_props(*vhost*, *source*, *destination*, *props*)

An individual binding between two exchanges. Similar to the individual binding between an exchange and a queue, above.

Parameters

- **vhost** (*str*) –
- **source** (*str*) –
- **destination** (*str*) –
- **props** (*str*) –

get_channels()

A list of all open channels.

get_channels_by_channel(*channel*)

Details about an individual channel.

Parameters **channel** (*str*) –

get_cluster_name()
Name identifying this RabbitMQ cluster.

get_connections()
A list of all open connections.

get_connections_by_name(*name*)
An individual connection. DELETEing it will close the connection. Optionally set the "X-Reason" header when DELETEing to provide a reason.

Parameters **name** (*str*) –

get_connections_channels_by_name(*name*)
List of all channels for a given connection.

Parameters **name** (*str*) –

get_consumers()
A list of all consumers.

get_consumers_by_vhost(*vhost*)
A list of all consumers in a given virtual host.

Parameters **vhost** (*str*) –

get_definitions()
The server definitions - exchanges, queues, bindings, users, virtual hosts, permissions and parameters. Everything apart from messages. POST to upload an existing set of definitions. Note that:

- The definitions are merged. Anything already existing on the server but not in the uploaded definitions is untouched.

get_exchanges()
A list of all exchanges.

get_exchanges_bindings_destination_by_vhost_and_name(*vhost*, *name*)
A list of all bindings in which a given exchange is the destination.

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

get_exchanges_bindings_source_by_vhost_and_name(*vhost*, *name*)
A list of all bindings in which a given exchange is the source.

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

get_exchanges_by_vhost(*vhost*)
A list of all exchanges in a given virtual host.

Parameters **vhost** (*str*) –

get_exchanges_by_vhost_and_name(*vhost*, *name*)
An individual exchange. To PUT an exchange, you will need a body looking something like this:

```
{
    "auto_delete": False,
    "internal": False,
    "type": "direct",
```

```
    "durable": True,
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

get_extensions()
A list of extensions to the management plugin.

get_nodes()
A list of nodes in the RabbitMQ cluster.

get_nodes_by_name(*name*)
An individual node in the RabbitMQ cluster. Add "?memory=true" to get memory statistics, and "?binary=true" to get a breakdown of binary memory use (may be expensive if there are many small binaries in the system).

Parameters **name** (*str*) –

get_overview()
Various random bits of information that describe the whole system.

get_parameters()
A list of all parameters.

get_parameters_by_component(*component*)
A list of all parameters for a given component.

Parameters **component** (*str*) –

get_parameters_by_component_and_vhost(*component*, *vhost*)
A list of all parameters for a given component and virtual host.

Parameters

- **component** (*str*) –
- **vhost** (*str*) –

get_parameters_by_component_and_vhost_and_name(*component*, *vhost*, *name*)
An individual parameter. To PUT a parameter, you will need a body looking something like this:

```
{
    "vhost": "/",
    "component": "federation",
    "name": "local_username",
    "value": "guest"
}
```

Parameters

- **component** (*str*) –
- **vhost** (*str*) –
- **name** (*str*) –

get_permissions()
A list of all permissions for all users.

get_permissions_by_vhost_and_user(*vhost*, *user*)
An individual permission of a user and virtual host. To PUT a permission, you will need a body looking something like this:

```
{
    "write": ".*",
    "read": ".*",
    "configure": ".*"
}
```

Parameters

- **vhost** (*str*) –
- **user** (*str*) –

get_policies()
A list of all policies.

get_policies_by_vhost(*vhost*)
A list of all policies in a given virtual host.

Parameters **vhost** (*str*) –

get_policies_by_vhost_and_name(*vhost*, *name*)
An individual policy. To PUT a policy, you will need a body looking something like this:

```
{
    "priority": 0,
    "pattern": "^amq.",
    "apply-to": "all",
    "definition": {
        "federation-upstream-set": "all"
    }
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

get_queues()
A list of all queues.

get_queues_bindings_by_vhost_and_name(*vhost*, *name*)
A list of all bindings on a given queue.

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

get_queues_by_vhost(*vhost*)
A list of all queues in a given virtual host.

Parameters **vhost** (*str*) –

get_queues_by_vhost_and_name(*vhost*, *name*)
An individual queue. To PUT a queue, you will need a body looking something like this:

```
{
    "node": "rabbit@smacmullen",
    "auto_delete": False,
    "durable": True,
    "arguments": {}
}
```

Parameters

- **vhost** (*str*) –
- **name** (*str*) –

get_users()
A list of all users.

get_users_by_name(*name*)
An individual user. To PUT a user, you will need a body looking something like this:

```
{
    "password": "secret",
    "tags": "administrator"
}
```

Parameters **name** (*str*) –

get_users_permissions_by_user(*user*)
A list of all permissions for a given user.

Parameters **user** (*str*) –

get_vhosts()
A list of all vhosts.

get_vhosts_by_name(*name*)
An individual virtual host. As a virtual host usually only has a name, you do not need an HTTP body when PUTing one of these. To enable / disable tracing, provide a body looking like:

```
{
    "tracing": True
}
```

Parameters **name** (*str*) –

get_vhosts_permissions_by_name(*name*)
A list of all permissions for a given virtual host.

Parameters **name** (*str*) –

get_whoami()
Details of the currently authenticated user.

C

D

G