RabbitMQ Management API Client Documentation

Release 0.1

Alchemy

August 13, 2015

Contents

CHAPTER 1

Introduction

RabbitMQ Managemenet API Client is an object oriented PHP client for the RabbitMQ Management API provided by the RabbitMQ Management Plugin

This library depends on Guzzle and Doctrine Common.

Installation

We rely on composer to use this library. If you do no still use composer for your project, you can start with this `composer.json` at the root of your project:

```
{
    "require": {
        "alchemy/rabbitmq-management-client": "master"
    }
}
```

Install composer :

```
# Install composer
curl -s http://getcomposer.org/installer | php
# Upgrade your install
php composer.phar install
```

You now just have to autoload the library to use it :

```
<?php
require 'vendor/autoload.php';
```

This is a very short intro to composer. If you ever experience an issue or want to know more about composer, you will find help on their website http://getcomposer.org/.

CHAPTER 3

Basic Usage

Here is a simple way to instantiate the APIClient an retrieve a queue :

```php
<?php
use RabbitMQ\Management\APIClient;

$client = APIClient::factory(array('url'=>'localhost'));

$queue = $client->getQueue('/', 'queue.leuleu');
```

The APIClient factory requires the url option to build. Other available options are :

- scheme : The scheme to access the API endpoint (default to 'http')
- port : The port number of the API endpoint (default to '55672')
- username : The username to connect to the API endpoint (default to 'guest')
- password : The password to connect to the API endpoint (default to 'guest')

For all available methods, it is recommended to browse the API.

CHAPTER 4

Guarantees

What you probably want to do with this library is to ensure the RabbitMQ queues, exchanges and bindings settings. This can be easily done with the `Guarantee` Component.

`Guarantee` will look in the configuration to find if what you ask for is already correctly set up and eventually fix it if you ask for it.

4.1 Probing a queue

Let's probe the status of a queue ; the probe will return one of the following constants :

- `RabbitMQ\Management\Guarantee::PROBE_RESULT_OK` If the probed entity is set up with

correct options - `RabbitMQ\Management\Guarantee::PROBE_RESULT_MISCONFIGURED` If the probed entity is set up with wrong options - `RabbitMQ\Management\Guarantee::PROBE_RESULT__ABSENT` if the probed entity is absent

```php
<?php
use RabbitMQ\Management\APIClient;
use RabbitMQ\Management\Entity\Queue;
use RabbitMQ\Management\Guarantee;

$client = APIClient::factory(array('url'=>'localhost'));
$manager = new Guarantee($client);

$queue = new Queue();
$queue->vhost = '/';
$queue->name = 'queue.leuleu';
$queue->durable = true;
$queue->auto_delete = false;

$status = $manager->probeQueue($queue);

switch ($status) {
    case Guarantee::PROBE_ABSENT;
        echo "The queue does not exists";
        break;
    case Guarantee::PROBE_MISCONFIGURED;
        echo "The queue exists but is not well configured";
        break;
    case Guarantee::PROBE_OK;
```

```
        echo "The queue exists and is well configured";
        break;
}
```

4.2 Probing an exchange

The same is available for exchanges :

```php
<?php
use RabbitMQ\Management\Entity\Exchange;

$exchange = new Exchange();
$exchange->vhost = '/';
$exchange->name = 'exchange.dispatcher';
$exchange->type = 'fanout';

$status = $manager->probeExchange($exchange);
```

4.3 Ensure queue configuration

Let's now ensure a queue is set up as required :

```php
<?php
use RabbitMQ\Management\APIClient;
use RabbitMQ\Management\Entity\Queue;
use RabbitMQ\Management\Guarantee;

$client = APIClient::factory(array('url'=>'localhost'));
$manager = new Guarantee($client);

$queue = new Queue();
$queue->vhost = '/';
$queue->name = 'queue.leuleu';
$queue->durable = true;
$queue->auto_delete = false;

// Will modify the queue if it is not configured yet
$manager->ensureQueue($queue);
```

CHAPTER 5

Recipes

These recipes are samples of code you could re-use. Most of these are about guarantees that are also provided by the `Guarantee` component.

5.1 Monitor a queue

```
<?php
use RabbitMQ\Management\Exception\EntityNotFoundException;
use RabbitMQ\Management\Entity\Queue;

try {
    $queue = $client->getQueue('/', 'queue.leuleu');

    sprintf("Queue contains %d messages", $queue->messages);
    sprintf("Queue is idle since %s", $queue->idle_since);

} catch (EntityNotFoundException $e) {
    echo "The queue is not found";
}
```

Handling Exceptions

RabbitMQ Management API Client throws 4 different types of exception :

- `RabbitMQ\Management\Exception\EntityNotFoundException` is thrown when an entity is not found.
- `RabbitMQ\Management\Exception\InvalidArgumentException` is thrown when an invalid argument (name, vhost, ...) is provided
- `RabbitMQ\Management\Exception\PreconditionFailedException` is thrown when you try to add an existing queue/exchange with different parameters (similar to HTTP 406).
- `RabbitMQ\Management\Exception\RuntimeException` which extends SPL RuntimeException

All these Exception implements `RabbitMQ\Management\Exception\ExceptionInterface` so you can catch any of these exceptions by catching this exception interface.

CHAPTER 7

Report a bug

If you experience an issue, please report it in our issue tracker. Before reporting an issue, please be sure that it is not already reported by browsing open issues.

CHAPTER 8

Ask for a feature

We would be glad you ask for a feature ! Feel free to add a feature request in the issues manager on GitHub !

CHAPTER 9

Contribute

You find a bug and resolved it ? You added a feature and want to share ? You found a typo in this doc and fixed it ? Feel free to send a Pull Request on GitHub, we will be glad to merge your code.

CHAPTER 10

Run tests

RabbitMQ Management Client relies on PHPUnit for unit tests. To run tests on your system, ensure you have PHPUnit installed, and, at the root of the project, execute it :

```
phpunit
```

CHAPTER 11

About

RabbitMQ Management Client has been written by Romain Neutron @ Alchemy for Gloubster.

CHAPTER 12

License

RabbitMQ Management API client is licensed under the MIT License