
EaseMqtt Documentation

Release 0.1.0

Michael Palmer

Feb 01, 2019

Contents

1	Installation	3
2	Basic Usage	5
3	Exceptions	7
4	Class: EaseMqtt	9
4.1	Properties	9
4.2	Method: publish(topic, message, qos?)	9
4.3	Method: subscribe(topic, qos?)	10
4.4	Method: end()	10
5	Class: EaseEventHandler	11
5.1	Method: handleIncoming(topic, msgbuf)	11
5.2	Method: handleConnection()	11
5.3	Method: handleError()	12
6	Class: EaseRequest	13
6.1	Properties	13
7	Class: EaseResponse	15
7.1	Properties	15
7.2	Method: reply(message, qos?)	15
8	Util	17

Current Version: v0.1.0

I developed EaseMqtt from the need of a simpler approach to developing services that communicate via MQTT, and also an extension to the existing functionality of the `mqtt` package.

CHAPTER 1

Installation

EaseMqtt is hosted on npm it can be installed using:

```
npm install --save easemqtt
```


CHAPTER 2

Basic Usage

The main difference in implementation is `mqtts` uses a slash `/` delimiter, by default `EaseMqtt` uses a dot `.` delimiter, of course this can be overridden by providing a `delimiter` property to the `EaseMqtt` constructor. It's also possible to use wildcards in the topic name, by default this is disabled but can be enabled with the `wildcard` property.

```
import { connect } from 'mqtt';
import { EaseMqtt } from 'easemqtt';

const client = connect('mqtt://localhost');
const easemqtt = new EaseMqtt(client);

// subscribe
easemqtt.subscribe('some.topic', 2); // topic, optional qos (default 1)

// publish
easemqtt.publish('some.topic', 'hello world'); // topic(s), content (string/object)

easemqtt.on('some.topic', (req, res) => {
  const body = req.body;
  const replyTo = req.replyTo;

  if (replyTo)
    return res.reply('all done'); // string or object
});
```


CHAPTER 3

Exceptions

- `EaseError` - thrown if arguments are invalid
- `EaseClientError` - thrown if the `mqtt` client fails to subscribe

Added: v0.1.0

This is the main constructor of the `EaseMqtt` package, and provides all the functionality by integrating the different classes together.

4.1 Properties

- `wildcard - boolean` - enable wildcard support (default `false`)
- `delimiter - string` - set the delimiter that's used (default `.`)
- `subscribe - string[]` - a list of topics to subscribe to (default `[]`)
- `qos - QoS` - set the default qos level (default `1`)

4.2 Method: `publish(topic, message, qos?)`

Added: v0.1.0

Publish a message to topic(s) returning the message id.

4.2.1 Arguments

- `topic - string | string[]` - the topic(s) to publish the message too
- `message - IEaseMessage` - the unencoded message content
- `qos - QoS` - optional: override the default qos level

4.2.2 Return

- `Promise<number>` - when resolved, the message ID is returned.

4.3 Method: `subscribe(topic, qos?)`

Added: v0.1.0

This method has no return value Subscribe to topic(s) in order to receive published messages.

4.3.1 Arguments

- `topic` - `string | string[]` - the topic(s) to subscribe to
- `qos` - `QoS` - override the default qos level

4.4 Method: `end()`

Added: v0.1.0

This method has no arguments, or return value Terminate the connected client and handle clean up.

Class: EaseEventHandler

Added: v0.1.0

`EaseEventHandler` is responsible for converting the `MqttClient` events, to an `EaseMqtt` event. It also handles incoming messages and firing the relevant events.

5.1 Method: `handleIncoming(topic, msgbuf)`

Added: v0.1.0

An Event listener that's creates the `EaseRequest` and `EaseResponse` objects, and emits the event from the topic name. This method also handles topic name translation, and message decoding.

5.1.1 Arguments

- `topic` - `string` - the topic the incoming message came on
- `msgbuf` - `Buffer | string` - the encoded message content

5.1.2 Return

- `EaseRequest` - for testing purposes, this function returns the `EaseRequest` object.

5.2 Method: `handleConnection()`

Added: v0.1.0

This method has no arguments, or return value An Event listener that emits a `mqtt:connect` event when a `MqttClient` has successfully made a connection to the broker.

5.3 Method: `handleError()`

Added: v0.1.0

This method has no arguments, or return value An Event listener that emits a `mqtt:error` event when a `MqttClient` reported an `Error`.

Class: EaseRequest

Added: v0.1.0

Contains the information relating to an incoming request. This class is mainly a collection of properties but will likely include methods in future.

6.1 Properties

- `topic` - `string` - the topic the message appeared on
- `replyTo` - `string` - the topic to send a response too
- `body` - `IEaseMessage` - the decoded message content
- `easeMqtt` - `IEaseMqtt` - the `EaseMqtt` instance

Class: EaseResponse

Added: v0.1.0

Contains the information relating to a response. This class includes a method that's used to easily reply to a message. It also contains properties.

7.1 Properties

- `request` - `IEaseRequest` - the request instance and properties
- `topic` - `string` - the topic the request was made on
- `replyTo` - `string` - the topic to send the response too

7.2 Method: `reply(message, qos?)`

Added: v0.1.0

Easily send a response message to the `replyTo` topic if provided. This method won't throw an exception if it's not provided.

7.2.1 Arguments

- `message` - `IEaseMessage` - the unencoded message content
- `qos` - `QoS` - override the default qos level

CHAPTER 8

Util

There are a number of Utility functions available which are not exported by default. These will be documented soon.