

# MqttClientOptions

August 17, 2026

## MqttClientOptions

**Kind:** Interface

**Source:** `packages/microservices/external/mqtt-options.interface.ts`

**Part of:** [Microservices](#)

`MqttClientOptions` defines connection and protocol settings for an MQTT client used by the microservices transport layer. It supports direct MQTT/TCP connections as well as WebSocket-based MQTT transports, including client identity, keepalive behavior, and WebSocket-specific options.

## Properties

| Property                     | Type                               |
|------------------------------|------------------------------------|
| <code>port</code>            | <code>number</code>                |
| <code>host</code>            | <code>string</code>                |
| <code>hostname</code>        | <code>string</code>                |
| <code>path</code>            | <code>string</code>                |
| <code>protocol</code>        | <code>'wss'</code>                 |
| <code>wsOptions</code>       | <code>{ [x: string]: any; }</code> |
| <code>keepalive</code>       | <code>number</code>                |
| <code>clientId</code>        | <code>string</code>                |
| <code>protocolId</code>      | <code>string</code>                |
| <code>protocolVersion</code> | <code>number</code>                |
| <code>clean</code>           | <code>boolean</code>               |
| <code>reconnectPeriod</code> | <code>number</code>                |
| <code>connectTimeout</code>  | <code>number</code>                |

| Property        | Type                                                                                                                                                                                                                                                                              |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| username        | string                                                                                                                                                                                                                                                                            |
| password        | string                                                                                                                                                                                                                                                                            |
| incomingStore   | any                                                                                                                                                                                                                                                                               |
| outgoingStore   | any                                                                                                                                                                                                                                                                               |
| queueQoSZero    | boolean                                                                                                                                                                                                                                                                           |
| properties      | { sessionExpiryInterval?: number; receiveMaximum?: number; maximumPacketSize?: number; topicAliasMaximum?: number; requestResponseInformation?: boolean; requestProblemInformation?: boolean; userProperties?: object; authenticationMethod?: string; authenticationData?: any; } |
| reschedulePings | boolean                                                                                                                                                                                                                                                                           |
| servers         | Array<{ host: string; port: number; }>                                                                                                                                                                                                                                            |
| resubscribe     | boolean                                                                                                                                                                                                                                                                           |
| will            | { topic: string; payload: string; qos: QoS; retain: boolean; }                                                                                                                                                                                                                    |
| transformWsUrl  | (url: string, options: any, client: any) => string                                                                                                                                                                                                                                |

## Diagram

```
graph LR
  Client[Microservice MQTT Client] --> Options[MqttClientOptions]
  Options --> Connection[host / hostname / port / path]
  Options --> Transport[protocol]
  Options --> Identity[clientId]
  Options --> Protocol[protocolId / protocolVersion]
  Options --> Keepalive[keepalive]
  Options --> WebSocket[wsOptions]

  Transport --> MQTT[mqtt / mqttS / tcp / ssl]
  Transport --> WS[ws / wss / wx / wxs]
  WebSocket --> WS
```

## Usage

```
import type { MqttClientOptions } from './mqtt-options.interface';

const mqttOptions: MqttClientOptions = {
  host: 'broker.example.com',
```

```
hostname: 'broker.example.com',
port: 8883,
path: '/mqtt',
protocol: 'mqtt',
clientId: 'orders-service',
protocolId: 'MQTT',
protocolVersion: 4,
keepalive: 60,
wsOptions: {},
};

// Pass mqttOptions to the MQTT client or microservice transport configuration.
```

## AI Coding Instructions

- Provide a valid `protocol` value that matches the target transport, such as `mqtt` for secure TCP or `wss` for secure WebSockets.
- Keep `host` / `hostname` , `port` , and `path` consistent with the MQTT broker endpoint; `path` is especially important for WebSocket connections.
- Use a stable, unique `clientId` per running service instance to avoid broker-side client session conflicts.
- Set `protocolId` and `protocolVersion` to values supported by the broker; common MQTT settings are `MQTT` and version `4` .
- Only populate `wsOptions` when using a WebSocket-based protocol and pass broker-required headers, TLS options, or proxy configuration there.

## How it works

`MqttClientOptions` is a TypeScript interface that describes MQTT connection options. It extends `ISecureClientOptions` , so its shape includes optional TLS key, certificate, CA, and `rejectUnauthorized` fields. [mqtt-options.interface.ts:8](#) [mqtt-options.interface.ts:136-150](#)

All of its fields are optional. The declaration itself contains no runtime validation, defaults assignment, error throwing, or side effects. [mqtt-options.interface.ts:8-135](#)