ISecureClientOptions

August 18, 2026

ISecureClientOptions

Kind: Interface

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

Part of: Microservices

`ISecureClientOptions` defines TLS/SSL configuration for MQTT client connections in the microservices package. It supplies client credentials, trusted certificate authorities, and certificate-validation behavior when connecting to secure MQTT brokers.

Properties

Property	Type
`key`	`string
`cert`	`string
`ca`	`string
`rejectUnauthorized`	`boolean`

Diagram

```
graph LR
  Client[MQTT Client] --> Options[ISecureClientOptions]
  Options --> Key[key: Client private key]
  Options --> Cert[cert: Client certificate]
  Options --> CA[ca: Trusted CA certificates]
  Options --> Reject[rejectUnauthorized: Validate broker certificate]
  Options --> Broker[Secure MQTT Broker]
```

Usage

```ts
import { readFileSync } from 'node:fs';
import type { ISecureClientOptions } from './mqtt-options.interface';

const secureOptions: ISecureClientOptions = {
  key: readFileSync('./certs/client-key.pem'),
  cert: readFileSync('./certs/client-cert.pem'),
  ca: readFileSync('./certs/ca-cert.pem'),
  rejectUnauthorized: true,
};

// Pass secureOptions to the MQTT client connection configuration.
```

AI Coding Instructions

- Provide `key` and `cert` together when configuring mutual TLS authentication.
- Use `ca` to trust private or self-signed certificate authorities instead of disabling validation.
- Keep `rejectUnauthorized` set to `true` in production to verify the MQTT broker certificate.
- Accept certificate values as PEM strings, `Buffer` instances, or arrays when multiple certificates are required.
- Load private keys and certificates from secure configuration or secret storage; do not hardcode them in source files.