IClientReconnectOptions

August 18, 2026

IClientReconnectOptions

Kind: Interface

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

Part of: Microservices

`IClientReconnectOptions` defines the MQTT message stores used when an MQTT client reconnects. It provides separate stores for incoming and outgoing packets so message state can be preserved or restored across connection interruptions.

Properties

Property	Type
`incomingStore`	`any`
`outgoingStore`	`any`

Diagram

```
graph LR
  Client[MQTT Client] --> Reconnect[IClientReconnectOptions]
  Reconnect --> Incoming[incomingStore]
  Reconnect --> Outgoing[outgoingStore]
  Incoming --> RestoredIncoming[Restored incoming packet state]
  Outgoing --> ResentOutgoing[Resent pending outgoing packets]
```

Usage

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

const reconnectOptions: IClientReconnectOptions = {
  incomingStore: new Map(),
  outgoingStore: new Map(),
```

```
};

// Pass the stores into the MQTT client reconnect configuration.
function configureReconnect(options: IClientReconnectOptions) {
  return {
    incomingStore: options.incomingStore,
    outgoingStore: options.outgoingStore,
  };
}

const mqttReconnectConfig = configureReconnect(reconnectOptions);
```

AI Coding Instructions

- Provide both `incomingStore` and `outgoingStore` when configuring reconnect behavior.
- Use stores that match the MQTT client's expected store implementation; avoid assuming `any` accepts arbitrary incompatible values.
- Preserve store instances across reconnect attempts when pending packet state must survive temporary disconnections.
- Ensure outgoing stores can retain unacknowledged QoS messages so they can be resent after reconnection.
- Integrate these options with the underlying MQTT client configuration rather than handling packet restoration manually.