

MqttContext

August 18, 2026

MqttContext

Kind: Class

Source: `packages/microservices/ctx-host/mqtt.context.ts`

Part of: [Microservices](#)

`MqttContext` provides metadata for an incoming MQTT message handled by a NestJS microservice. It exposes the MQTT topic and the raw MQTT packet so message handlers can inspect routing information, QoS, headers, and other transport-level details.

Extends: `BaseRpcContext`

Methods

Method	Signature	Returns
<code>getTopic</code>	<code>getTopic()</code>	<code>void</code>
<code>getPacket</code>	<code>getPacket()</code>	<code>void</code>

Diagram

```
graph LR
  Client["Client[MQTT Client]"] --> Broker["Broker[MQTT Broker]"]
  Broker --> Server["Server[NestJS MQTT Server]"]
  Server --> Handler["Handler[Message Handler]"]
  Server --> Context["Context[MqttContext]"]
  Context --> Topic["Topic[getTopic()]"]
  Context --> Packet["Packet[getPacket()]"]
  Handler --> Context
```

Usage

```
import { Controller } from '@nestjs/common';
import { Ctx, MessagePattern, MqttContext } from '@nestjs/microservices';

@Controller()
export class DeviceController {
  @MessagePattern('devices/+/status')
  handleStatusUpdate(
    payload: { online: boolean },
    @Ctx() context: MqttContext,
  ) {
    const topic = context.getTopic();
    const packet = context.getPacket();

    console.log(`Received message on topic: ${topic}`);
    console.log(`QoS level: ${packet.qos}`);

    return { received: true, online: payload.online };
  }
}
```

AI Coding Instructions

- Use `@Ctx() context: MqttContext` in MQTT `@MessagePattern()` handlers when transport metadata is required.
- Call `getTopic()` to inspect the original MQTT topic, especially when using wildcard topic patterns.
- Call `getPacket()` for MQTT-specific metadata such as QoS, retain flags, duplicate delivery status, and packet properties.
- Treat the packet as transport metadata; keep business logic focused on the message payload whenever possible.
- Ensure handlers are configured under a NestJS microservice using the MQTT transport before relying on `MqttContext`.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `MqttController` — `integration/microservices/src/mqtt/mqtt.controller.ts` :16