

# MQTT Gateway - FAQ

Unten findest du häufig gestellte Fragen mit kurzen Antworten. Detailinformationen zu diversen Themen findest du in Unterartikeln.

## Detailliertere Informationen

- [Einrichtung am MQTT-Device](#)
- [FHEM Geräte per MQTT steuern](#)
- [HTTP-Eingänge am Miniserver debuggen](#)
- [MQTT Gateway - Alle FHEM-Readings weitergeben](#)
- [MQTT Gateway - Bridge Mode](#)
- [MQTT Gateway - Konfiguration sichern und auf anderem/neuen LoxBerry wiederherstellen](#)
- [MQTT Gateway - Loxone Logger Receiver](#)
- [MQTT Gateway - Subscriptions und Topics](#)
- [MQTT Gateway - Troubleshooting Guide](#)
- [MQTT Gateway 2.0 What's New](#)
- [MQTT-Gateway - Generischer POST / GET / JSON Empfänger](#)
- [Using multiple LoxBerry's with a single MQTT broker](#)
- [Was ist MQTT? Warum MQTT?](#)

## Wie funktioniert das keepaliveepoch?

Das Topic `<hostname>/mqttgateway/keepaliveepoch` (per HTTP: `<hostname>_mqttgateway_keepaliveepoch`) versendet im Minutentakt den aktuellen [Unix Epoch](#) Zeitstempel.

Es ist nicht relevant, welche Zeit in dem Wert drin steht, sondern nur, dass sich der Wert jede Minute ändert. Diese Änderung kannst du mit dieser Schaltung validieren: [epochtime \(Unix-Zeit\) zum Prüfen auf Datenaktualität nutzen](#)

Das keepaliveepoch wird vom MQTT Gateway an den Broker gesendet. Dieses Topic wird vom MQTT Gateway außerdem automatisch subscribed, sodass dieser Datensatz per MQTT vom Broker wieder zurück zum Gateway kommt, und dadurch an den Miniserver weitergeleitet wird. Damit prüft dieser Zeitstempel sowohl das MQTT Gateway als auch den MQTT Broker. Wenn das keepaliveepoch am Miniserver "stehen" bleibt (sich also nicht mehr ändert), ist irgendwas davon kaputt. Mit dem genannten Check (keine Änderung nach mehr als z.B. 2 Minuten) kannst du am Miniserver z.B. eine Alarmierung auslösen.

Der vom Plugin mitgelieferte Check für den [LoxBerry Selbsttest / Healthcheck](#) prüft u.a. ebenfalls das keepaliveepoch.

## Wie funktioniert der Cache (= "Zwischenspeicher")?

Klassiker: Subscription ist eingerichtet, Daten werden in der Incoming Overview angezeigt, aber es kommt nichts am Miniserver an.

## Klassische Antwort: Cache!

Der Datencache für HTTP- und UDP-Übertragungen sind eine Funktion von **LoxBerry** (nicht des Plugins!) zur Reduktion von sinnlosen Datenübertragungen zum Miniserver.

Wenn ein Datensatz übertragen wird, wird der Name und der Wert am LoxBerry gespeichert. Kommt nun mit dem gleichen Namen eine weitere Übertragung, wird zuerst geprüft, ob sich der Datensatz geändert hat. Wenn JA → Daten übertragen (und wieder im Cache gespeichert). Wenn NEIN → Gleiche Daten werden nicht übertragen, weil es sinnlos wäre (der Miniserver würde gar nicht darauf reagieren).

Was funktioniert, um den Cache zu umgehen?

→ NEIN: MQTT Gateway neustarten. Da der Cache von LoxBerry (und nicht dem Plugin) gehalten wird, ist ein Neustart des Gateways völlig egal, bringt nichts.

→ NEIN: Das MQTT-Gerät veranlassen, die Daten nochmals zu senden. Bringt nichts. Wenn die Daten gleich sind, werden sie gecached.

→ NEIN: Mosquitto Broker neu starten. Wieder das gleiche - gleiche Daten werden gecached.

→ JA: Andere Daten senden. Eine Änderung wird übertragen.

→ JA: In der Incoming Overview unten auf "Delete cache" klicken. Die Daten werden bei der nächsten Übertragung des Geräts neu zum Miniserver gesendet.

→ JA: Per UDP an das Gateway folgenden Befehl senden: reconnect (nur dieses Wort, sonst nichts). Das setzt das Cache-Kennzeichen zurück, die nächsten Daten werden auf jeden Fall übertragen.

→ JA: Beim entsprechenden Eintrag in der Incoming Overview auf "Disable Cache" stellen. Bei der nächsten Übertragung des Geräts wird neu zum Miniserver übertragen. Das Häkchen danach wieder löschen.

→ JA: Warten. Alle 60 Minuten löscht LoxBerry das Cache-Kennzeichen und überträgt einkommende Daten (auch wenn sie gleich sind) erneut.

→ JA: LoxBerry neu starten. Der Cache liegt in der RAM-Disk, die beim Reboot gelöscht wird.

## Ich habe ein Gerät (z.B. Taster), das immer den gleichen Wert sendet - wie mach ich das?

- In der Incoming Overview das Häkchen "Show details" aktivieren.
- Bei deinem Wert das Häkchen "Reset after send" aktivieren.

Dadurch wird nach jeder Übertragung ein "0" nachgeschickt. Damit verhält sich der Virtuelle Eingang wie ein Impuls.

## Was bedeutet "Expand JSON data"?

MQTT-Geräte übermitteln Daten in der Regel auf zwei Arten: Entweder jedes Topic enthält genau einen Wert. Oder aber das Gerät überträgt mehrere Werte in einem Topic im **JSON-Format** (ein Standard zum Austausch von Daten). In so einem JSON-Datensatz können mehrere oder gar viele Werte stehen, auch in verschachtelten Strukturen.

Da der Loxone Miniserver mit dem JSON-Datenformat nichts anfangen kann, liest das MQTT Gateway diese Daten aus, und erzeugt daraus automatisch entsprechend viele Einzeldatensätze, die übertragen werden. Das Übertragungs-Caching (siehe oben) wird dabei auf jeden Einzelwert angewendet. Überträgt ein Gerät also z.B. 10 Werte in einem JSON-Paket, und nur ein Wert hat sich geändert, wird auch nur dieser eine Wert übertragen.

## Welche Übertragungsart ist besser: HTTP oder UDP?

→ HTTP ist für den Miniserver schonender, da jeder Eingang gezielt angesprochen wird.

→ Im Gegensatz dazu muss bei UDP der Miniserver bei jedem übertragenen Datensatz über alle Eingangsbefehle hinweg eine Suche durchführen. Das potenziert sich mit der Anzahl der Geräte und Befehle.

→ Nur per HTTP kann Text übertragen werden (virtueller Texteingang).

## Was ist der Unterschied zwischen Publish und Retain?

**Publ**ish wird gesendet und fertig ("Fire-and-forget"). Alle Geräte, die gerade online sind, bekommen die Daten.

**Retain** bzw. das Retain-Flag bedeutet, dass der Datensatz am Broker gespeichert bleibt. Alle Geräte, die online sind, bekommen die Daten sofort. Stoßen später neue Clients dazu, bekommen auch diese den Datensatz geliefert, auch wenn das eigentliche Übermitteln schon länger her ist.

Wenn beispielsweise das MQTT Gateway startet, erhält es automatisch vom Broker *rückwirkend* alle Daten, die zuvor mit retained übermittelt wurden.

Das Retain-Flag ("Retain", "Retained") kann man oft an MQTT-Geräten setzen. Bei der Übertragung von Miniserver zum Gateway kannst du im Kommando angeben, ob **publish** oder **retain**.

Retain bei MQTT ist so ähnlich wie die Remanenz beim Miniserver.

## Was unterscheidet die Verwendung von MQTT von der Verwendung der in Loxone möglichen Virtuellen HTTP-Eingänge?

Hier eine Gegenüberstellung: [MQTT vs. HTTP - Vorteile und Nachteile](#)

## Welche Befehle gibt es, die ich per MQTT Gateway an mein MQTT-fähiges Gerät senden kann?

Das weiß ich nicht, das wird vom Hersteller des Geräts festgelegt. Meist gibt beim Hersteller eine Dokumentation. Halte Ausschau nach "Schnittstellenbeschreibung", "API", "Developer" bzw. "Entwickler", "Datenübertragung" oder dergleichen. Oft gibt es bei Herstellern eine eigene Unterseite für die Schnittstellenbeschreibung.

## **Ein Datensatz ist aus der Incoming Overview verschwunden, obwohl das Gerät noch funktioniert**

Die Incoming Overview hebt die Daten nur 24 Stunden lang auf, außer ein Datensatz wurde zwischenzeitlich wieder gesendet. Wenn dein Gerät seit 24 Stunden nichts gesendet hat, verschwinden die Daten. Das heißt aber nicht zwangsläufig, dass die Verbindung gestört ist.

## **Wie kann ich ein einzelnes MQTT-Gerät auf dessen Zustand/Verfügbarkeit überwachen?**

Suche in der MQTT-Dokumentation oder auf der Konfigurationsseite des Geräts nach den Ausdrücken "LWT", "Last Will", "Last Will and Testament". Dieses Topic ist das "Testament" des Geräts, das unmittelbar beim *Start der Verbindung* vom Geräts zum Broker übergeben wird. Fällt das Gerät aus, exekutiert *der Broker* den letzten Willen, nämlich das genannte Topic so zu setzen, wie das Gerät es anfangs im "Testament" angefordert hat. In der Regel geht das Gerät auf "Offline", "Disconnected" oder dergleichen. Diesen Status kannst du überwachen - er wird automatisch gesetzt, wenn das Gerät die Verbindung verliert.

## **Mein MQTT Gateway Logfile ist so groß**

Schön, dass du bemerkt hast, dass es ein Log gibt (würde ich mir bei manch Useranfrage wünschen ).

Das Logfile liegt in der RAM-Disk (also im Speicher), es belastet NICHT die SD-Karte.

Das Logfile wird von LoxBerry automatisch gestutzt, wenn es zu groß wird. Solange der LoxBerry Selbsttest nicht schreit, brauchst du dir keine Sorgen darum machen.

## **Wenn ich einen Virtuellen Ausgangs-Befehle in Richtung MQTT Gateway zu meinem Rollladen/Licht/etc. auslöse, fängt dieses Gerät zu Flackern/Stottern an.**

Heißt dein Virtueller **Ausgangs**-Befehl zufällig genauso wie das Topic, an das du sendest? Dann hast du eine Loop erzeugt!

Beispiel: Du hast im Gateway `shellies/#` abonniert. Dein Virtueller Ausgangsbefehl heißt `shellies_shelly123_relay` und sendet an das Topic `shellies/shelly123/relay` den Befehl UP.

Wenn du den Virtuellen Ausgangsbefehl triggerst, sendet der Miniserver per UDP ans Gateway. Das

Gateway sendet den Befehl weiter an MQTT. Sofort empfängt das Gateway selbst per MQTT diesen Befehl. Das Gateway übermittle die Daten nun zurück an den Virtuellen Eingang shellies\_shelly123\_relay. Da es sich aber nicht um einen Virtuellen Eingang, sondern einen Virtuellen Ausgang handelt, den wir beschreiben, sendet der Miniserver nun den neuen Wert per UDP an das MQTT Gateway.....

→ Virtuelle Ausgänge Richtung MQTT Gateway dürfen nicht so heißen wie die Virtuellen Eingänge, die vom Gateway an den Miniserver übermittle werden!

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

[https://wiki.loxberry.de/konfiguration/widget\\_help/widget\\_mqtt/mqtt\\_gateway\\_faq/start](https://wiki.loxberry.de/konfiguration/widget_help/widget_mqtt/mqtt_gateway_faq/start)

Last update: **2023/04/15 08:18**