Plugin-Daten		
Autor	Florian	
Logo	Tibber	
Status	BETA	
Version	0.3.2	
Min. LB Version	2.0.0	
Pre-Release Download	https://github.com/flod-1/LoxBerry-Plugin-Tibber2MQTT/archive/refs/tags/v0.3.2-beta.zip	
Beschreibung	Das Plugin ruft die stündlichen Preise über die Tibber API ab und stellt sie über das MQTT Plugin dem Miniserver zur Verfügung.	
Sprachen	EN, DE	

Tibber 2 MQTT

Version History...

https://github.com/flod-1/LoxBerry-Plugin-Tibber2MQTT/releases

Funktion des Plugins

Das Plugin ruft die stündlichen Preise über die Tibber API ab und stellt sie über das MQTT Plugin (siehe hier) dem Miniserver zur Verfügung. Folgende Werte können genutzt werden:

- Absolute Preise
 - Total (Strom + Steuer)
 - Strom
 - Steuer
 - Level von Tibber übermitteltes Preislevel und Wert in Loxone:
 - Very Cheap = -2
 - Cheap = -1
 - Normal = 0
 - High = 1
 - Very High = 2
- Relative Preise
 - Total (Strom + Steuer)
 - $\circ \ \text{Strom}$
 - \circ Steuer
 - $\circ\,$ Level von Tibber übermitteltes Preislevel und Wert in Loxone:
 - Very Cheap = -2
 - Cheap = -1
 - Normal = 0
 - High = 1

- Very High = 2
- Niedrigster Total-Preis + Zeitpunkt
- Höchster Total-Preis + Zeitpunkt
- Übermittlung in Euro/kWh oder Cent/kWh

Der Abruf bzw. die Übermittlung an den Miniserver erfolgt:

- per Cronjob stündlich; inkl. Cache, um die API-Aufrufe zu minimieren
- manuell über die Plugin-Verwaltung

Download

Der Download kann über obigen Link oder aber direkt auf Github erfolgen.

Bitte beachtet, dass sich das Plugin aktuell noch im BETA-Status befindet - Fehler können jederzeit auftreten.

Installation

Verwendet ihr Loxberry 2.x muss vor der Installation dieses Plugins das MQTT Plugin (verfügbar hier) installiert werden.

Verwendet ihr Loxberry 3.x ist dies nicht erforderlich, da das MQTT Plugin dort zum Loxberry Core gehört.

Während der Installation legt das Plugin automatisch die erforderliche MQTT Subscription an:

Subscriptions defined by other plugins	0
	Tibber 2 MQTT tibber/#

Konfigurationsoptionen

- 1. Bitte trage deinen persönlichen Tibber Token im ersten Feld ein. Wenn du diesen noch nicht hast, erstelle ihn über dein persönliches Konto siehe Link auf der Plugin Konfigurationsseite.
- 2. Zur Vermeidung von unnötig Traffic auf dem Miniserver sind alle übergebenen Werte konfigurierbar. So könnt ihr zielgenau je nach eurer Anforderungen für die abzubildende Konfiguration im Miniserver Werte aktivieren bzw. deaktivieren.
- Derzeit empfehle ich die Aktivierung der Checkbox "Cent statt Euro", da Loxone nur 3 Nachkommastellen akzeptiert und dadurch bei Übermittlung in Euro die letzte Stelle verloren geht.

Tibber 2 MQTT Einstellungen

Bitte hier alle Tibber Einstellungen pflegen.	
Tibber Token Der Tibber-Token kann nach dem Login mit dem eigenen Tibebr	Benutzer auf <u>https://developer.tibber.com/</u> erstellt werden.
Hake die absoluten Preise an, die verwendet werden sollen.	Absolute Preise benutzen
	Verwende Total Preis je Stunde(erforderlich)
	Verwende Energie Preis je Stunde
	Verwende Steuer je Stunde
	Verwende Preislevel je Stunde
Hake die relativen (jetzt + x) Preise an, die verwendet werden sollen.	Relative Preise benutzen
	Verwende Total Preis je Stunde(erforderlich)
	Verwende Energie Preis je Stunde
	Verwende Steuer je Stunde
	Verwende Preislevel je Stunde
Anhaken, wenn Min/Max Werte verwendet werden sollen	Minimaler Preis in zukünftigen Preisen, basierend auf Total Preis.
	Maximaler Preis in zukünftigen Preisen, basierend auf Total Preis.
Entscheide, ob die Preise in Cents oder Euro übertragen werden sollen.	Verwende Cent statt Euro

Einrichtung in der Loxone Config Software

Je nach Bedarf und eingesetzter Loxone Config, kann die Konfiguration dort sehr unterschiedlich sein. Grundsätzlich gilt, dass die Werte per MQTT an den Miniserver übergeben werden. Je nach Konfiguration des MQTT Servers erfolgt dies per HTTP oder UDP. Ersteres ist aus Performance-Gründen empfohlen. Die genaue Konfigurationsanleitung ist auf dieser Seite des MQTT Servers gut beschrieben. Kurz zusammengefasst:

- 1. Öffnet den Incoming Overview des MQTT Servers
- 2. Klickt auf "copy", um den genauen Titel zu kopieren
- 3. Legt einen "virtuellen Eingang" (Achtung, nicht virtuellen HTTP Eingang) in Loxone Config an. Dabei ist zu beachten:
 - 1. Fügt den zuvor kopierten Namen als Bezeichnung ein, damit sie exakt übereinstimmen.
 - 2. Der virtuelle Eingang darf NICHT als digitaler Eingang gekennzeichnet sein.
 - 3. Setzt den gültigen Wertebereich auf -100 bis 200. (Damit seid könnt ihr alle Preise zwischen -1 Euro und +2 Euro empfangen; das sollte hoffentlich reichen ;))
 - 4. Je nachdem, ob ihr Euros oder Cents übertragt, sollte die Einheit v.2 (bei Cents) oder v.3 (bei Euros) betragen. Siehe auch Erklärung oben.

<u>Beispiel:</u>

Eigenschaften (Virtueller Eingang)		
T		
Eigenschaft	Wert	
Objekttyp	Virtueller Eingang	
Visualisierung		
Verwenden		
Visualisierungskennwort		
Bewertung	***	
Statistiken		
Statistik		
Berechtigungen		
Berechtigte Benutzer / Gruppen	Bearbeiten	
Einstellungen		
Fehlerausgang anzeigen		
Als Digitaleingang verwenden		
Logging/Mail/Call/Track		
Logger	nicht zugeordnet	
Mailer	nicht zugeordnet	
Caller	nicht zugeordnet	
Tracker	nicht zugeordnet	
Meldung bei EIN/Analogänderung		
Meldung bei AUS		
Minimaler Zeitabstand	5	
Validierung		
✓ Validierung überwachen		
Minimaler Wert	-100	
Maximaler Wert	200	
Zeitüberschreitung Empfang	0	
Standardwert	0	
Standardwert bei Start		
Anzeige		
Einheit	<v.2></v.2>	
Simulation/LiveView		
Frequenzmodus	Rechteck	
Frequenz	0	

Die virtuellen Eingänge lassen sich dann zur Steuerung verschiedener Dinge verwenden. Nachfolgend ein Beispiel, das ich aktuell verwende. Dieses basiert auf dem Spotprice-Optimizer, der mit Loxone Config 13.2 ausgeliefert wird. Der Ablauf lässt sich so zusammenfassen: Die Strompreise werden (im virtuellem Eingang) werden an die Eingänge des Spotprice-Optimizers übergeben. Täglich um 0:03 Uhr wird per Impuls der günstigste Zeitraum ermittelt. Der Zeitraum und der zu betrachtende Zeithorizont wird im Baustein hinterlegt. Sobald dieser Zeitpunkt erreicht ist, wird der Ausgang O eingeschaltet. Da bei Verwendung von absoluten Preisen immer nur die Preise des aktuellen Tages bekannt sind, vergleiche ich zusätzlich den Minimal ermittelten Preis des Bausteins mit dem in diesem Plugin ermittelten Minimalpreis. Denn dieser deckt - je nach Zeitpunkt des Triggers - bereits die Werte des darauffolgenden Tages mit ab. Ist der Preis auch in diesem Vergleich am günstigsten, schalte ich die Wallbox zur Ladung frei.

Im Screenshot-Beispiel ist dies nicht von großer Bedeutung, da die Werte bei Verwendung eines 0:03 Uhr Impulses ohnehin bereits als absolute Werte hinterlegt sind. Wenn allerdings bspw. der Trigger mit dem Einstecken des Autos in der Wallbox verbunden wird - weil ihr bspw. ermitteln wollt, wann ab diesem Zeitpunkt der günstigste Ladezeitpunkt ist - dann ist dies relevant. Beispiel: Das Auto wird Abends um 18 Uhr eingesteckt und der Trigger aufgerufen. Der Baustein kennt bei Verwendung absoluter Preise zu diesem Zeitpunkt nur die Preise bis 0 Uhr am gleichen Tag. Häufig sind aber vor allem die Preise in den frühen Morgenstunden am Folgetag am günstigsten. Durch die entsprechende Prüfung wird in diesem Beispiel nun verhindert, dass das Auto im günstigsten Zeitpunkt vor 0 Uhr lädt, insofern am Folgetag bereits günstigere Preise bekannt sind.



Anmerkung: Am 29.01.2023 befindet sich der Spotprice Optimizer von Loxone noch im BETA-Status, kann aber prinzipiell bereits verwendet werden. Als ich die Verwendung mit relativen Preisen getestet habe, die der Baustein grundsätzlich auch unterstützt, bin ich auf ein Problem gestoßen: Alle per virtuellem HTTP-Eingang gemeldeten Werte, die nicht in der aktuellen Stunde des Triggers übertragen wurden, werden verworfen. Dies trifft Stand jetzt leider nicht auf "normale" [kein HTTP] virtuelle Eingänge zu. Dies ist aber enorm wichtig, da bei Verwendung von relativen Preisen - je nach Übermittlungszeitpunkt - keine 24 Werte zur Verfügung stehen; es bleiben dann die zuletzt verwendeten Werte in den virtuellen Eingängen stehen und werden fälschlicherweise verwendet. Diesen - aus meiner Sicht Fehler - habe ich bereits Loxone gemeldet. Bis dahin bin ich auf die obere Logik ausgewichen, da aus meiner Sicht vor allem die Nacht bzw. frühen Morgenstunden am Folgetag wichtig sind. Bei absoluten Preisen habe ich bisher keinen anderen Weg gefunden, diese zu berücksichtigen.

Roadmap

- Abruf der Consumptions über die Tibber API
- Evtl. Logging Funktionalität, um die Daten langfristig zu monitoren/visualisieren
- Evtl. Demo-Token per Checkbox nutzbar machen

Fragen stellen und Fehler melden

Da sich das Plugin noch im BETA-Release befindet, können jederzeit Fehler auftreten. Außerdem ist die Vielfalt an Konfigurationsmöglichkeiten in Loxone riesig.

Wenn ihr weitere Werte benötigt oder einen Fehler entdeckt habt, tragt diese bitte direkt im Github ein: Issues · Issue Tracker auf Github

From: https://wiki.loxberry.de/ - LoxBerry Wiki - BEYOND THE LIMITS

Permanent link: https://wiki.loxberry.de/plugins/tibber2mqtt/start

Last update: 2024/02/04 22:02