Plugin-Daten	
Autor	Sven Thierfelder
Logo	
Status	STABLE
Version	2.0.1.1
Min. LB Version	1.2
Release Download	https://github.com/svethi/LoxBerry-CalDAV4Lox/archive/refs/tags/caldav4lox-2.0.1.1.zip
Pre-Release Download	https://github.com/mschlenstedt/LoxBerry-CalDAV4Lox/archive/refs/heads/master.zip
Beschreibung	Das CalDAV Plugin bietet die Möglichkeit, Kalendereinträge in CalDAV oder Google Kalendern für Funktionen aller Art im Loxone MiniServer zu verwenden.
Sprachen	EN, DE
Diskussion	https://www.loxforum.com/forum/german/software-konfiguration-programm-und-visualisierung/60811-loxberry-caldav-4-lox-plugin-verf%C3%BCgbar

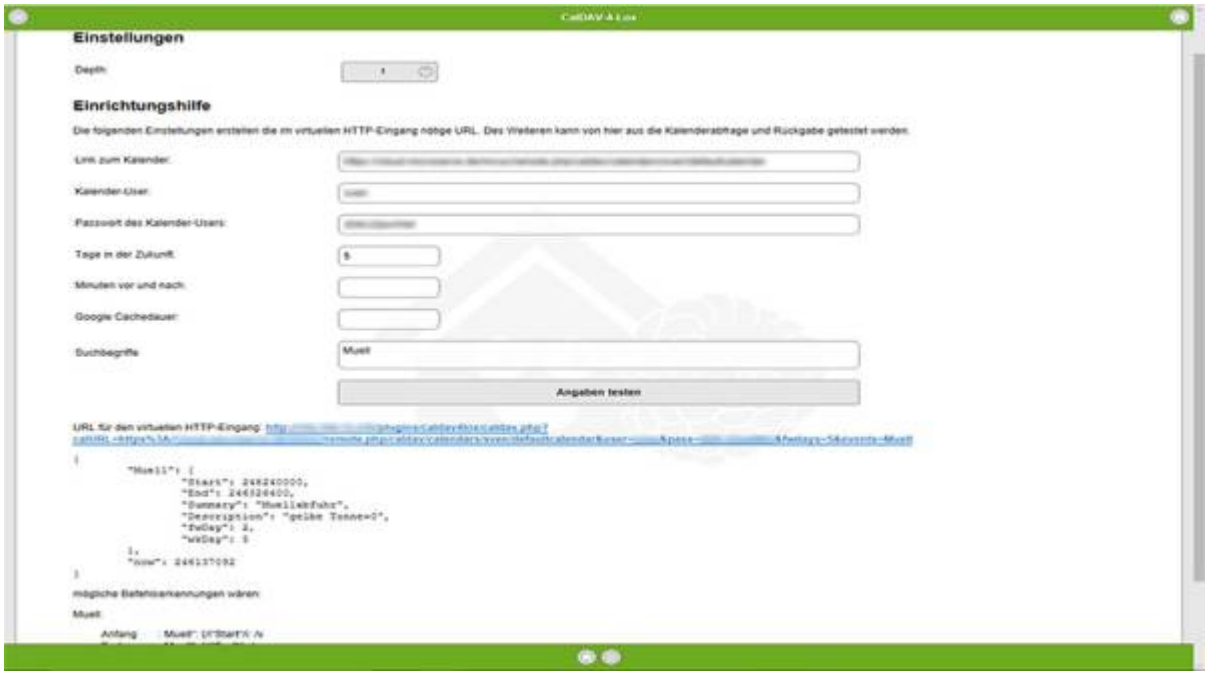
CalDAV-4-Lox

Version History...

Siehe Releases im GitHub Repository: https://github.com/svethi/LoxBerry-CalDAV4Lox/releases

---

Konfigurationsoptionen



- `Depth`: Als einzige Einstellung lässt sich das Plugin mit dieser Option an den verwendeten CalDAV-Server anpassen. Einige Server erwarten für die Abfrage eine Abfragetiefe von 0, andere eine 1. Da neuere Server meistens Depth 1 verwenden ist dies als Standard hier eingestellt. Sollte trotz korrekter URL zum Kalender kein Termin gefunden werden, kann die Einstellung 0 hier zum Erfolg führen.

Einrichtungshilfe

Mit der Einrichtungshilfe steht optional ein Tool zur Verfügung, mit dem man sich die URL für den virtuellen HTTP-Eingang zusammenbasteln lassen kann. Wer Schwierigkeiten damit hat sich die URL aus den einzelnen Komponenten zusammenzusetzen, gibt hier einfach die entsprechenden Daten ein und klickt auf "Angaben testen". Es erscheint daraufhin die benötigte URL. Des Weiteren wird der erzeugte Link auch gleich getestet und die Rückgabe unformatiert ausgegeben. Hier können Teile heraus kopiert und in der Befehlserkennung verwendet werden. Noch weiter unten werden einige gängige Befehlserkennungen zum direkten Verwenden für die angegebenen Suchbegriffe aufgelistet. Die in der Befehlserkennung erforderlichen Zeichenfolgen sind farblich hinterlegt.

Die Bedeutung der einzelnen Parameter wird im Bereich "Funktion des Plugin" beschrieben

Funktion des Plugin

Das Plugin ist aus den bereits verfügbaren Scripts für die CalDAV Anbindung aus dem Loxforum entstanden. Erwähnt sei hier auch noch einmal Christian Fenzl, der die Kalenderabfrage windowstauglich gemacht hat sowie das Caching des Google Kalenders beigesteuert hat.

Wie funktioniert nun das Ganze?

Das Plugin ist so konzipiert, dass im Plugin selbst keine Einstellungen vorgenommen werden müssen. Alle Daten, die zum Abfragen des Kalenders benötigt werden, werden in der URL des HTTP-Einganges als Parameter angegeben. So ist es auch möglich verschiedene Kalender auf unterschiedlichen Servern abzufragen.

Solch eine URL sieht in etwas so aus:

`http://loxberry/plugins/caldav4lox/caldav.php?calURL=http%3A//localhost/remote.php/caldav/calendars/USER/defaultcalendar&user=USER&pass=PASSWORT&fwdays=5&events=abfuhr`

Die einzelnen Angaben bedeuten folgendes:

- `http://loxberry/plugins/caldav4lox/caldav.php?` - ist die Adresse des Plugin auf dem LoxBerry. Dieser ist unter Umständen anzupassen.
- `calURL` - ist die URL des entsprechenden Kalenders so wie sie z.B. im Thunderbird/Lightning anzugeben ist. Bei den meisten Kalendern kann man diese URL in den Eigenschaften des Kalenders in Erfahrung bringen. Bei OwnCloud z.B. ist neben dem Kalendernamen auch ein Linksymbol mit dem Hilfetext "CalDAV-Link". Klickt man dieses Symbol an, erscheint unter den Kalendern ein Textfeld mit dem benötigten Link. Es ist aber darauf zu achten, dass hier der Doppelpunkt hinter http(s) in %3A geändert werden muss (Urlencoded). Bei der Konfiguration per Plugin-Einstellungshilfe wird die URL automatisch URL-encoded.
- `user` - Euer Benutzername für den Kalender
- `pass` - Euer Passwort für den Kalender
- `fwdays` - da man keine Möglichkeit hat eine flexible Zeitspanne in die URL einzufügen, habe ich diese Variante gewählt. hier wird die Anzahl der Tage angegeben, für die im Voraus im Kalender nachgesehen werden soll. Vom aktuellen Datum der Anfrage an. Null für nur den aktuellen Tag. Wird der Parameter nicht angegeben, wird 0 angenommen.

- `delay` (im Beispiel nicht benutzt) - ist die Anzahl der Minuten, die vor und nach der aktuellen Zeit nachgesehen wird. Wird der Parameter nicht angegeben, wird 60 bentzt. (Beispiel: ein Termin 14:30 - 15:00 Uhr im Kalender, wird zwischen 13:30Uhr und 16:00Uhr angezeigt wenn delay nicht angegeben wurde.
- `events` - eine Liste der Terminnamen im Kalender, nach denen gesucht werden soll mit Pipe/senkrechter Strich getrennt. Es wird auch nach Wortteilen gesucht. Ein Termin "Muellabfuhr" wird also auch durch Angabe von "llabf" gefunden. Der Suchbegriff muss URL-encoded sein! Aus "Leerung Gelbe Tonne" wird also "Leerung%20Gelbe%20Tonne". Ihr könnt dazu einen Webdienst nutzen wie z. B. https://www.urlencoder.org/de/
- `cache` (im Beispiel nicht benutzt) - ist die Anzahl der Minuten, die eine ICS/iCal Kalenderabfrage auf dem LoxBerry zwischengespeichert und benutzt wird bis der Kalender erneut "befragt" wird.

Als Rückgabe erhält der virtuelle HTTP-Eingang Daten wie folgende:

```
{
"Muellabfuhr": {
"Start": 194832000,
"End": 194918400,
"Description": "gelbe Tonne=1",
"fwDay": 1,
"wkDay": 5
}
"MSTest": {
"Start": 194802300,
"End": 194804100,
"Description": "",
"fwDay": 0,
"wkDay": 4
}
"now": 194801634
}
```

Das Ergebnis ist wird im JSON Format ausgegeben. Für jeden Suchbegriff gibt es ein Objekt. Gibt es zu dem Suchbegriff keine aktuellen Termin, so sind die Werte **-1** und Description **""**. Die zurückgegebenen Zeitwerte sind Loxone Zeitwerte und können über `<v.u>` humanreadable angezeigt werden. Zum Vergleich der Zeit mit der aktuellen Loxone-Zeit muss erst aus den Zeitvariablen des Systems die Loxone-Zeit errechnet werden. Das geht über einen Formelbaustein mit Eingängen **AI1 = Tage seit 2009**, **AI2 = Stunde**, **AI3 = Minuten** und **AI3 = Sekunden** und der Formel `((I1*24+I2)*60+I3)*60+I4`

Description sind die Notizen zum Termin. Hier unterscheide ich z.B. welche Tonne abgeholt wird. ***fwDay*** ist der Tag in der Zukunft, an dem der Termin stattfindet. Rufst Du Montags ab und hast `fwdays=5` angegeben und der Termin, der gefunden wurde ist am Mittwoch, so ist der Wert 2. ***wkDay*** ist der Wochentag des Termins 1=Mo bis 7=So. Im vorigen Beispiel wäre das also 3. ***now*** ist die aktuelle Zeit zur Antwortzeit. Diese kann für Vergleiche/Berechnungen im Programm verwendet werden.

Nur in der Developer-Version verfügbar:

Die folgenden Funktionen sind aktuell ausschließlich im PreRelease / Developerversion verfügbar!

In der aktuellen Developer-Version (siehe Pre-Release) gibt es weitere Funktionen, die der Abruf-URL des Kalenders hinzugefügt werden können:

events=Suchbegriff@@REGEX

Für Events kann man auch einen echten Regex als Suchbefehl mitgeben, dieser wird mit @@ abgetrennt. Der Teil vor dem @@ dient dabei nur als Name, gesucht wird mittels des Regex hinter dem @@. Wie oben erwähnt muss ein Suchbegriff URL-encoded werden - das gilt auch für das @@! Auf Grund einfacherer Lesbarkeit ist hier die Darstellung ohne URL-encoding zusätzlich gewählt. NUtzt einen Webdienst, um euren Suchbegriff zu URL-encoden, z. B. https://www.urlencoder.org/de/

Beispiel: Man möchte alle Einträge finden, die mit dem Begriff "Urlaub" beginnen, nicht aber Urlaub mitten im Titel haben. Der Befehl

```
&events=Urlaub@@^Urlaub.*
```

bzw. dann korrekt URL-encoded: `&events=Urlaub%40%40%5EUrlaub.%2A`

findet den Kalendereintrag "Urlaub Familie", nicht aber den Eintrag "Michael's Urlaub".

Man kann auch Regex-Suchen mit normalen Suchen kombinieren, zum Beispiel:

```
&events=Urlaub@@^Urlaub.*|Klavier|Hannes@@.*Hannes.*
```

bzw. dann korrekt URL-encoded:
`&events=Urlaub%40%40%5EUrlaub.%2A|Klavier|Hannes%40%40.%2AHannes.%2A`

Benötigt man eine PIPE in seinem Regex, muss diese escapet werden: `\|` wird zu `|` im Regex.
Benötigt man eine escapete PIPE, muss man diese doppelt escapen:: `\\|` wird zu `\|` im Regex.

Hilfen zu regulären Ausdrücken (Regex) findet man im Web überall, z. B. hier:
https://www.massiveart.com/de-de/blog/regex-zeichenfolgen-die-das-entwickler-leben-erleichtern

httpauth=AUTH-METHODE

Diese Funktion wird nur verwendet, wenn ihr einen iCal-Kalender im Plugin per URL herunterlädt - es muss sich .ics in der URL befinden! Bei einer echten CalDAV-Abfrage wird sie nicht verwendet.

Benötigt ihr gegenüber dem Server eine andere Authentifizierungsmethode als BASIC, könnt ihr dem Plugin hier die Authentifizierungsmethode mitgeben. Es können alle Authentifizierungsmethoden verwendet werden, die CURL unterstützt: https://curl.se/libcurl/c/CURLOPT_HTTPAUTH.html

Um zum Beispiel DIGEST zu verwenden, nutzt ihr:

```
&httpauth=DIGEST
```

Benötigt eure Authentifizierungsmethode einen Bearer (aktuell nur OAuth2.0), so verwendet ihr:

```
&httpauth=BEARER&bearer=geheimertokenvomprovider
```

downhelper=1

Nutzt euer CalDAV-Server (nicht ICS-Download - siehe oben die Option httpauth) eine andere Authentifizierungsmethode als BASIC (z. B. Digest), so kann das Plugin standardmäßig nicht auf diesen Server zugreifen. Das ist zum Beispiel beim Provider All-Inkl. der Fall. In diesem Fall könnt ihr versuchen den Download-Helper zu verwenden. Hierbei wird mit einer alternativen Methode auf den Server zugegriffen, die weitere Authentifizierungsmöglichkeiten nutzt. Sie lässt sich allerdings nicht weiter konfigurieren. Aktiviert dazu einfach den Download Helper mit

```
&downhelper=1
```

MQTT-Support

Mit Version 1.0.2 wurde der MQTT-Support eingefügt. Die MQTT Funktionalität bezieht sich in diesem Fall auf das Publizieren von Nachrichten. Für diese Funktion ist das installierte MQTT-Gateway Plugin von Christian Fenzl notwendig (in LoxBerry 3.0 ist es bereits enthalten!) und wird dann auch automatisch verwendet. Der Vorteil dieser Variante liegt darin, dass es auf diesem Weg z.B. einfach möglich wird, Texte auf den MiniServer zu übertragen.

Die verwendeten Topics folgen einem Schema. Das Topic beginnt mit `caldav4lox/events/`. Es folgt der übergebene Suchbegriff und daran anschließend dann sämtliche Werte, die auch im JSON übergeben werden. Wenn also als gesuchtes Event „Test“ angegeben wird, dann wäre das Topic `caldav4lox/events/Test/`. Wenn man die Description dann im Miniserver erhalten will, würde man `caldav4lox/events/Test/description` verwenden.

Einrichtung:

1. MQTT Gateway
    1. MQTT Gateway installieren
    2. Im MQTT Gateway eine Subscription anlegen: `caldav4lox/events/#`
2. CalDAV-Plugin
    1. Im CalDAV-Plugin die URL(s) zusammenstellen (lassen)
    2. Mit dem Webbrowser die URL testen
3. Loxone Config
    1. Für den Abruf des Kalenders beim CalDAV-Plugin einen "Virtuellen Ausgang" erstellen
    2. Virtuelle Ausgangs-Befehle erzeugen, die die URL bzw. bei mehreren Suchen die URLs abrufen
    3. Die Virtuellen Ausgangsbefehle auf eine Seite ziehen und einen Trigger anschließen (z.B. Stundenimpuls)
    4. Im MQTT Gateway die "Incoming Overview" öffnen. Dort werden die Daten angezeigt.
    5. Entsprechend MQTT - Schritt für Schritt: MQTT -> Loxone virtuelle Eingänge für die Daten von CalDAV anlegen

Bitte beachten, dass das MQTT Gateway identische Daten nicht mehrfach an den Miniserver sendet (Cache).

MQTT ohne Miniserver

Wenn der Trigger nicht vom Miniserver ausgelöst werden soll oder kann, kann auch ein Cronjob oder

ein anderes Tool die URL des Plugins abrufen, und somit den Kalender über CalDAV4Lox „befragen“. Das Plugin übergibt die Informationen an MQTT und das MQTT-Gateway pushed die Daten (wie oben beschrieben) an den Miniserver. Außerdem können die Kalenderdaten auch von anderen MQTT-fähigen Geräten genutzt werden, ohne dass überhaupt ein Miniserver verwendet wird.

Einrichtung in der Loxone Config Software

Das Plugin wurde für die Verwendung mit dem virtuellen HTTP-Eingang entwickelt. Detailierte Informationen hierzu gibt es in der Loxone Dokumentation.

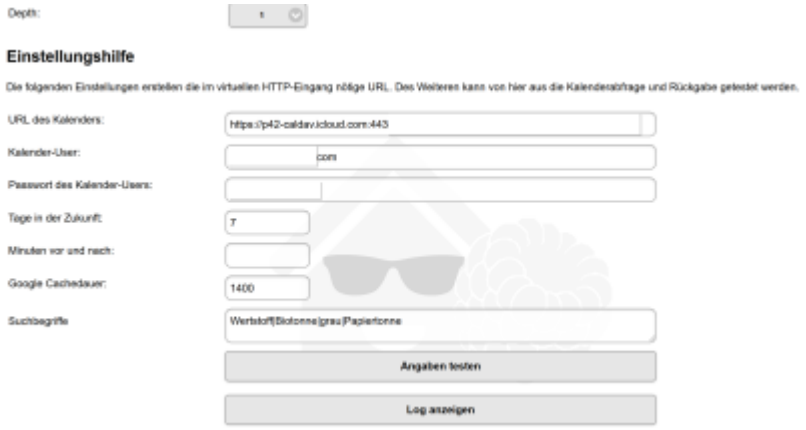
Alternativ kann die Einbindung über das in LoxBerry eingebaute MQTT Gateway erfolgen. Siehe Kapitel oben „MQTT Support“.

Einrichtung eines iCloud Kalenders

Zuerst müsst ihr diese Anleitung abarbeiten:
https://community.openhab.org/t/solved-apple-icloud-caldav-connection/32510
(Arbeitszeit ca. 15 Minuten)

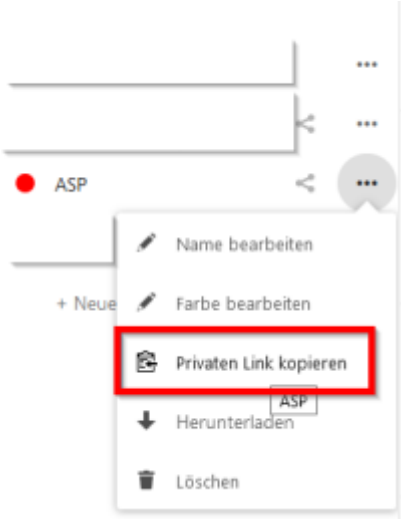
Apple arbeitet auch mit **Depth:1.**

Dann könnt ihr euren Link zu dem Kalender **URL des Kalenders:** eintragen. Im Anschluss gebt ihr noch euren Benutzernamen und als Passwort das bei Apple generierte Passowrt ein - **nicht euer Apple-ID Passwort!**



Einrichtung eines Nextcloud Kalenders

Meldet euch in der Nextcloud an und öffnet euren Kalender. In der Kalender APP könnt ihr den Kalender über "…" einen Link erzeugen.



Den Wert "Depth:" stellt ihr auf **1.**

Diesen Link könnt ihr dann im caldav4lox PlugIn im Feld "**URL des Kalenders**" reinkopieren.

Die Felder "**Kalender-User:**" und "**Passwort des Kalender-Users:**" füllt ihr mit euren Nextclouddaten.

Wenn ihr nun auf "Angaben testen" klickt sollte euch der nächste Termin aus eurem Kalender angezeigt werden.

Die "**URL für den virtuellen HTTP-Eingang:**" könnt ihr dann in der Loxone Config als Virtuellen HTTP Eingang benutzen.

Eigenschaften (Virtueller HTTP Eingang)

Eigenschaft	Wert
Allgemein	
Bezeichnung	NextcloudAbfall
Beschreibung	ASP im Nextcloud KAlender
Objekttyp	Virtueller HTTP Eingang
Einstellungen	
URL	http://[redacted]/plugins/caldav4lox/caldav.php?callURL=https%3A//[redacted]/remote.php/dav/calendars/[redacted]
Abfragezyklus [s]	10
Timeout [ms]	4000
Anzahl erlaubte Timeouts	10

Den Abfragezyklus könnt ihr auf 86400 (ein Tag) einstellen.

Einrichten eines Google Kalenders

```
Depth: 1
```

Auf https://calendar.google.com/ zu den Einstellungen des Kalenders gehen (oben Zahnrad / "Einstellungen" / "Einstellungen für meine Kalender" / Kalender auswählen), dort dann unter "Kalender integrieren" die "Privatadresse im iCal Format" kopieren und im Feld URL des Kalenders eingeben

User: und Passwort: leer lassen

Der Standard Timeout von 4000ms kann bei Google Kalender u.U. nicht reichen. Es ist besser ihn auf den Maximalwert von 8000 zu setzen.

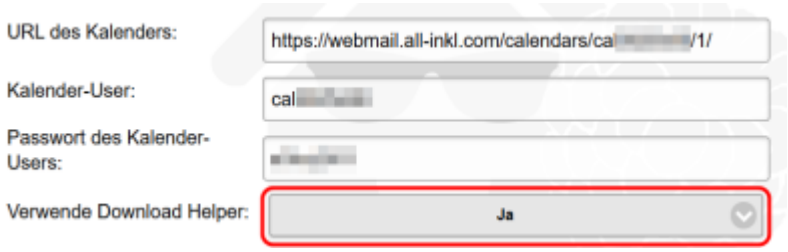
Einrichtung eines All-Inkl. Kalenders (Roundcube)

Der Zugriff funktioniert aktuell nur mit der Developer-Version (Pre-Release!)

Richtet im Webmail euren Kalender ein und bergesst nicht ein separates Passwort für den Kalender zu vergeben! Anleitung:
https://all-inkl.com/wichtig/anleitungen/programme/e-mail/caldav-kalenderfunktion/einstellungen-im-webmail-anzeigen_457.html

Da All-Inkl. nicht mit der herkömmlichen BASIC-Authentifizierung arbeitet, muss der Download Helper für den Zugriff aktiviert werden!

```
&downhelper=1
```



Bekannte Probleme

Aktuelle Kalendertools benutzen beim Verschieben eines Termins aus einer Terminreihe gern mal die Variante einen neuen Termin dafür anzulegen und dazu anzugeben welcher Termin aus welcher Reihe

damit ersetzt wird (RECURRENCE-ID, UID, SEQUENCE). Diese Variante wird nicht unterstützt. Beim Verschieben eines Termins sollte dieser Einzeltermin gelöscht und dafür ein neuer Einzeltermin angelegt werden.

PHP führ manchmal zu einem SSL Context Creation Failure → Neustart des Loxberrys hilft

Fehler melden

Fehlerberichte bitte direkt im Loxforum melden.

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

Permanent link:
https://wiki.loxberry.de/plugins/caldav_4_lox/start

Last update: **2025/04/29 06:01**