Dark Sky direkt in Loxone einbinden

Mit diesem Template können Daten direkt von Dark Sky in den Miniserver geholt werden. Dies funktioniert ohne Umweg über einen PC, und auch ohne LoxBerry, anhand des virtuellen HTTP Eingangs.

> **Unterschied zur LoxBerry Lösung**
>
> Für LoxBerry gibt es eine Wetterlösung (Weather4Loxone), die ebenfalls über DarkSky (oder auch Weatherbit) Informationen bereitstellt. Die Unterschiede zu der Lösung auf dieser Seite sind:
>
> - Weather4Loxone benötigt einen Raspberry (oder ähnliche Hardware) und entsprechende technische Kenntnisse
> - Weather4Loxone bietet auch einen Emulator für das Loxone Wetterservice ("Cloud Weather Emulator") - die Lösung hier liefert der Loxone nur die Werte aber nicht die hübschen Wetterbilder, die das Loxone Wetterservice liefert
>
> wer also "nur" aktuelle Wetterdaten und Vorhersagen für die Loxone Programmierung / Darstellung benötigt, ist mit der Lösung hier vermutlich besser dran.

Token und Ort

Unter https://darksky.net/dev bekommt man wenn man sich registriert (E-Mail, Passwort) einen "secret key" (32 stellig, Zahlen und Kleinbuchstaben). Mit diesem kann man dann direkt im Browser Anfragen stellen:

```
https://api.darksky.net/forecast/[secret-key]/[latitude],[longitude]
```

In der URL des Templates musst du die Felder [secret-key] und [latitude], [longitude] durch deine Angaben ersetzen:

- secret-key = der secret-key, den du bekommen hast
- latitude, longitude = geografische Breite und Länge; einfach über z.B. GoogleMaps rausfindbar (hast du vermutlich ohnedies in der Loxone schon angegeben)

Danach kann man noch eine Reihe von Request-Parametern setzten ("?" und danch mit "&" getrennt). Hier die empfohlenen (genaue Beschreibung aller Parameter siehe https://darksky.net/dev/docs):

- units - Einheiten für Temperatur, Windgeschwindigkeit etc. - us, si oder andere
- lang - Sprache der textuellen Angaben z.B. de
- exclude - Um die Größe der Antworten zu reduzieren, wenn z.B. keine minütlichen Vorhersagen nötig sind

Beispiel:

```
https://api.darksky.net/forecast/[secret-key]/[latitude],[longitude]?units=si&lang=de&exclude=minutely,alerts,
flags
```

> **Abfragelimit**
>
> Die Abfrageanzahl in der kostenlosen Version sind auf 1.000 Abfragen pro Tag limitiert. Für mehr Abfragen müssen z.B. Kreditkarteninfos hinterlegt werden (man zahlt dann pro Abfrage). Bei 1.000 Abfragen pro Tag könnte man alle 87 Sekunden eine Abfrage machen. Für die Wetterdaten ist es aber sicher ausreichend, diese alle 20-30 Minuten zu aktualisieren, alles andere ist vermutlich nur vorgetäuschte Genauigkeit (ob eine Wolke gerade meine PV-Anlage beschattet kann man so vermutlich nicht rausfinden) - Template ist auf 600 Sek. = 10 Min. voreingestellt.

Ergebnis

Als Ergebnis erhält man ein JSON Dokument (menschen- und maschinenlesbare Antwort) mit:

- aktuellen Wetterdaten
- die Vorschau für die nächsten Minuten, Stunden und Tage
- Warnungen und Flags

Die jeweiligen Antwortblöcke sind optional - nicht für jeden Ort gibt es sie (z.B. fehlt oft die Vorschau für die nächsten Minuten)

Die Loxone kann selbst das JSON parsen, dafür muss die Loxone aber angewiesen werden, wo die einzelnen Werte in der Antwort zu finden sind. Genau das macht das Template.

Template

Template herunterladen: VI_Dark Sky.xml

Das Einbinden des Templates funktioniert so: Templates in Loxone Config einbinden

Enthaltene Daten

Alle hier beschriebenen Daten werden je nach Ort/Wetterstation und Datenblock (daily/hourly/...) nur optional geliefert. Siehe auch https://darksky.net/dev/docs#response-format

Eingang	Einheit *)	Beschreibung und Anmerkung	Template Version
apparent_temp	<v.2> °C	gefühlte Temperatur	V2 - nur curr
apparent_temp_high	<v.2> °C	maximale gefühlte Temperatur	-
apparent_temp_high_time		UNIX Zeit der maximalen gefühlten Temperatur	-
apparent_temp_low	<v.2> °C	minimale gefühlte Temperatur	-
apparent_temp_low_time		UNIX Zeit der minimalen gefühlten Temperatur	-
cloud_cover	<v> %	Bedeckung (zwischen 0% und 100%)	V1 - nur curr
dew_point	<v.2> °C	Taupunkt (Schwüle wird bei Taupunkt > 16 °C empfunden, wenn Lufttemperatur = Taupunkt, dann ist die relative Luftfeuchtigkeit 100%)	V1 - nur curr, day1
humidity	<v> %	Luftfeuchtigkeit (zwischen 0% und 100%)	V1 - nur curr
icon		Maschinenlesbarer Text für Zusammenfassung auf Englisch: clear-day, clear-night, rain, snow, sleet, wind, fog, cloudy, partly-cloudy-day, partly-cloudy-night. Passende Icons dazu gibt es z.B. bei http://darkskyapp.github.io/skycons/	V4 - nur curr, day1
moon	0..1	Mondphase. 0 = Neumond, 0,25 erster Halbmond, 0,5 = Vollmond, 0,75 zweiter Halbmond	-
nearest_storm_bearing	<v.2>° (0° .. 360°)	Richtung des nächstgelegenen Sturmes. 0° = Norden	-
nearest_storm_distance	<v.2> km	Entfernung des nächstgelegenen Sturmes	-
ozone		Ozone Wert, https://de.wikipedia.org/wiki/Dobson-Einheit	-
snow_accum	<v.2> cm	Schneefall in der gegebenen Zeiteinheit	V3 - nur day1-day4
precip_intensity		Niederschlagsmenge (falls es regnet - d.h. immer mit precip_prob gemeinsam betrachten)	V5 - nur day1-day4
precip_intensity_error		Standardabweichung für die Niederschlagsmenge	-
precip_intensity_max		größte Niederschlagsmenge	-
precip_intensity_max_time		UNIX Zeit der größten Niederschlagsmenge	-
precip_prob	<v> %	Niederschlagswahrscheinlichkeit (zwischen 0% und 100%)	V1 - nur day1-day4
precip_type		Niederschlagsart: Regen, Schnee oder gefrierender Regen = Eiskörner = Schneeregen	-
pressure		Luftdruck	-
summary		Menschenlesbare Zusammenfassung (nicht fürs Parsen geeignet)	-
sunrise_time		UNIX Zeit des Sonneraufgangs	-
sunset_time		UNIX Zeit des Sonnenuntergangs	-
temp	<v.2> °C	Lufttemperatur	V1 - nur curr
temp_high	<v.2> °C	Tageshöchsttemperatur	V1 - nur day1-day3
temp_high_time		UNIX Zeit der Tageshöchsttemperatur	-

temp_low	<v.2> °C	Tagestiefsttemperatur	V1 - nur day1-day3
temp_low_time		UNIX Zeit der Tagestiefsstemperatur	-
time		UNIX-Zeit des jeweiligen Blocks (Beginn der Minute, Beginn der Stunde, Beginn des Tages)	V1 - nur curr, day1
uv_index		UV-Index	-
uv_index_time		UNIX Zeit des höchsten UV-Index des Tages	-
visibility	<v.2> km	Sichtweite	V4 - nur curr
wind_bearing		Windrichtung (0° = Norden)	V2 - nur curr
wind_gust	<v.2> km/h	Spitzenwindgeschwindigkeit	V1 - nur curr, hr1-hr2
wind_gust_time		UNIX Zeit der höchsten Spitzenwindgeschwindigkeit des Tages	-
wind_speed	<v.2> km/h	Windgeschwindigkeit	V1 - nur curr, hr1-hr2

*) Bei Einheit wird angenommen, dass "units=si" in der Abfrage gesetzt wurde; das <v.2> bedeutet (wie in der Loxone Config) dass der Wert auf 2 Nachkommastellen genau geliefert wird

Aufbau der virtuellen Eingänge des Templates

Alle Eingänge beginnen mit ds (Dark Sky).

- **ds_curr_*** sind die aktuellen Wetterdaten
- **ds_day*_*** sind die Tages-Vorhersagen der nächsten 7 Tage (day0 ist heute, day1 ist morgen, day2 ist übermorgen)
- **ds_hr*_*** sind die stündlichen Vorhersagen bis zum Ende des nächsten Tages (hr1 ist die kommende, volle Stunde, hr2, hr3, hr4 die zweite, dritte und vierte Stunde)

Im Template sind nur die Daten der nächsten 2 Stunden und nächsten 4 Tage enthalten (DarkSky liefert aber bis zu 24 Stunden zurück). Du kannst die Eingänge ergänzen, wenn du weitere Daten benötigst.

Bitte beachte: Für jeden einzelnen Eingangsbefehl wird vom Miniserver eine Suche durchgeführt. Diese Suchen benötigen Leistung des Miniservers. Daher empfehle ich, Eingänge, die du absolut nicht benötigst, aus deiner Programmierung zu löschen.

Wetter visualisieren

Die Werte <icon> und <summary> eignen sich dafür in der Visualisierung dargestellt zu werden. Das Problem dabei ist, dass diese keine gültigen Werte liefern, da bei virtuellen Eingängen keine Texte geparsed werden können. Die beiden liefern daher immer 0.

Es gibt jetzt folgende Lösungen für dieses Problem:

- Nur einzelne Buchstaben parsen damit könnte man mit icon1, icon2, icon3, ... und entsprechender Logik in einem Statusbaustein das korrekte Icon darstellen. Ist aber keine Lösung für summary
- Mittels Pico-C im Programm-Baustein einen Parser für genau diese Werte schreiben Referenzen dazu findet man unter https://groups.google.com/forum/#!topic/loxone-english/KZNGlIQz4qU, https://github.com/netdata-be/loxone/blob/master/weatherservice/wunderground.c, https://www.loxforum.com/forum/faqs-tutorials-howto-s/1049-howto-wunderground-api-in-programm-baustein-abfragen - alles Lösungen für weather-underground und alle möglichen Werte - wir brauchen mMn nur einen Parser für <icon> und <summary>, der Rest ist leichter mit dem virtuellen Eingang gelöst

Template-Versionen

Alle Versionen sind auf der Attachments Seite zu finden

Version	Beschreibung
1 (18.12.2018)	Initiales Template, 23 virtuelle Eingänge
2 (30.12.2018)	2 weitere Eingänge
3 (6.1.2019)	2 weitere Eingänge
4 (19.1.2019)	4 weitere Eingänge
5 (29.5.2019)	4 weitere Einträge, summary entfernt