

# JUNG Visu Pro Device Editor

## Process Interface Online Weather Request

### Version 2

#### Content:

<b>DISCLAIMER</b> .....	<b>2</b>
<b>FUNCTIONALITY</b> .....	<b>4</b>
1. Configuration.....	5
2. Datapoint Folders.....	6
2.1. Datapoint Folder 'Current Weather'.....	6
2.2. Datapoint Folder 'Location'.....	6
2.3. Datapoint Folder 'Forecast'.....	6
3. Running the "Online Weather Request" Process Interface.....	7
3.1 Access Control.....	7
3.2 Execution of Write Commands.....	7
3.3 Execution of Read Commands.....	7

## Disclaimer

Reasonable effort is made to maintain the currency and accuracy of this document and the correct implementation of the "Online Weather Request" software described here, but the use of the information and the software is the sole responsibility of the user. We do not accept any liability for the up-to-dateness, correctness, completeness, suitability or quality of both document and software.

The software is developed with great care, but in no event we warrant that the software is error free or that the customer will be able to operate the software without problems or interruptions. We do not warrant for problems due to improper use of the software or other causes, which are out of the sphere of our influence.

The user must assume the entire risk of using the software. In no event we will be liable for any lost revenue, profit, or data, or for special, direct, indirect, consequential, incidental, or punitive damages, however caused and regardless of the theory of liability arising out of the use of or inability to use the software, even if we have been advised of the possibility of such damages.

This exclusion shall not apply where e.g. under product liability law or in cases of willful misconducts there is a mandatory liability. In no event shall we or our suppliers' liability to the customer, whether in contract, tort (including negligence), or otherwise, exceed the price paid by the customer.

The "Online Weather Request" software is a sample provided with source code, that you can modify according your needs and in your responsibility.

**If you do not agree to these terms, you are not allowed to use the "Online Weather Request" software.**

## Address

### **ALBRECHT JUNG GMBH & CO. KG**

Volmestraße 1  
D-58579 Schalksmühle

Phone +49 (0) 23 55/80 60  
Fax +49 (0) 23 55/80 61 89



[mail.info@jung.de](mailto:mail.info@jung.de)

### **Copyright ©2019 Agentilo GmbH**

All Rights reserved

## Trademarks

EIB®, KNX® and ETS® are registered trademarks of the KNX Association.

LON® is a registered trademark of Echelon Corporation registered in the United States and other countries.

OPC® is a registered trademark of OPC Foundation.

Sax Basic Engine is a trademark of Sax Software Corporation.

Adobe Acrobat® is a registered trademark of Adobe Systems Incorporated.

Microsoft®, ActiveX®, DirectX®, Windows®, Windows NT®, Excel®, Visual Basic® are registered trademarks of Microsoft Corporation.

All trademarks and registered trademarks are the property of their respective owners.

The process interface example uses the restful API of the 'Free Plan' of

<https://openweathermap.org>.

## Functionality

The "Online Weather Request" process interface shows an example of how a data record formatted in JSON can be queried from a server and then evaluated by the script interface.

The script shows the queries and the evaluation of responses on the example of the API of OpenWeatherMap (<https://openweathermap.org>), a JSON/XML restful API, which is freely available at the time of creation of this process interface.

The prerequisite for using this process interface is an account at OpenWeatherMap, that allows the data request via the freely usable API (with up to 60 API calls per minute). Basis of the script is the API described in the documentation of the weather service (<https://openweathermap.org/current> and <https://openweathermap.org/forecast5>).

Furthermore, the software library "json.lua" is used, which is available for free use under the MIT license (<https://github.com/rxi/json.lua>).

The user can use this process interface to query current weather data and the forecast for up to 4 following days. The process interface provides several folders for this purpose: A configuration folder to configure the interface and several data point folders to store the received weather data.

The configuration folder contains the URL's and an automatic query interval. By changing the value of the binary datapoint "Manual update" (for example by a switch control in the visualization), the update of the weather data can be carried out independently of the automatic interval.

## 1. Configuration

In order work properly, the URL's of the weather API must be entered in the configuration folder. These URL's contain the API key that is assigned when creating the OpenWeatherMap account. This API key is visible on the website of the account.

The URL's have the following structure:

[https://api.openweathermap.org/data/2.5/<TYPE>?  
zip=<ZIP>&units=metric&lang=en&appid=<API-KEY>](https://api.openweathermap.org/data/2.5/<TYPE>?zip=<ZIP>&units=metric&lang=en&appid=<API-KEY>)

**TYPE:** The type changes depending on whether the current weather or the forecast is retrieved.  
The script supports the following types:

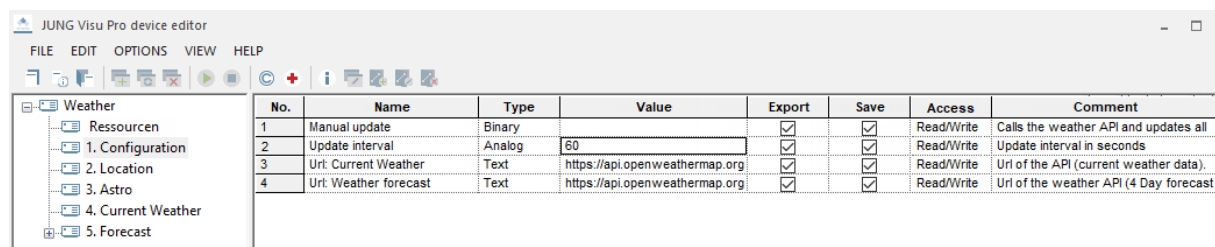
- For current weather: weather
- For forecast: forecast

**API-KEY:** Key assigned when the account is created.

**ZIP:** Postcode of the city for which the weather data should be requested.  
Also contains the country code. Example: 58579, DE

The configuration folder can be selected by clicking in the directory tree. A table appears in the corresponding window on the right. Enter the URL's in the row "Url: Current weather" and "Url: Weather forecast".

In addition, an update interval of at least 60 seconds can be set. If this value remains empty, it is initialized with 3600 seconds by default.



The screenshot shows the JUNG Visu Pro device editor window. On the left is a directory tree with 'Weather' expanded, showing sub-items: 'Ressourcen', '1. Configuration', '2. Location', '3. Astro', '4. Current Weather', and '5. Forecast'. The '1. Configuration' item is selected. On the right is a table with the following data:

No.	Name	Type	Value	Export	Save	Access	Comment
1	Manual update	Binary		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Read/Write	Calls the weather API and updates all
2	Update interval	Analog	60	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Read/Write	Update interval in seconds
3	Url: Current Weather	Text	https://api.openweathermap.org	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Read/Write	Uri of the API (current weather data).
4	Url: Weather forecast	Text	https://api.openweathermap.org	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Read/Write	Uri of the weather API (4 Day forecast)

After the process interface has been started, a request is sent to the entered URL. If you have entered the correct URL's, the process interface receives the JSON formatted weather data as response and evaluates it.

## 2. Datapoint Folders

After the process interface creation the datapoint tree provides the following folders: 'Current weather', 'Location' and the 'Forecast' for one or more days.

### 2.1. Datapoint Folder 'Current Weather'

Contains the weather data of the current day.

### 2.2. Datapoint Folder 'Location'

Contains additional information about the current location. This information (e.g. geographical longitude / latitude, time zone) will be provided by the weather service.

### 2.3. Datapoint Folder 'Forecast'

The datapoint folder 'Forecast', subfolder 'Day forecast', contains the weather data of the next days. When the process connection is created, only one day folder is already created by default. The number of days required to evaluate all information from the API response is 4 full days.

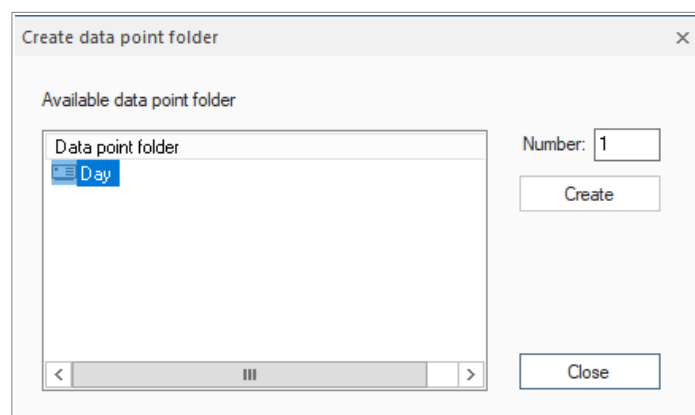
Creation of a new day folder:

- select the folder 'Day forecast' in the tree structure,
- open the context menu via right mouse click,
- select the command **Create datapoint folder**,

alternatively, choose the menu item **Edit** and click **Create datapoint folder**.

The dialog **Create datapoint folder** opens. Select the datapoint type **Day** and click the button **Create**.

The names of all datapoints and datapoint folders may be renamed at any time. This has no impact on the working of the script, because the script uses internal names which are fixed.



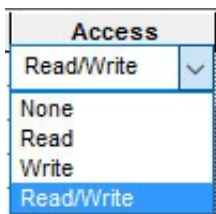
Each day contains folders for detailed weather data at 3 hour intervals and a general forecast.

### 3. Running the "Online Weather Request" Process Interface

To start the "Online Weather Request" process interface, choose the menu item **Options** and click **Start process interface** or it will be started automatically when configured and the devices project is started. In consequence, it may be started automatically from the process model.

After the initialization, the weather data will be queried automatically from the weather service. The resulting data will be evaluated and displayed in the related datapoint folders.

#### 3.1 Access Control



The script evaluates access rights. The predefined access rights should not be changed. Write access should only be assigned to the variables in the configuration folder.

#### 3.2 Execution of Write Commands

**Important:** Transferring the HTTP requests to the device editor is the last operation of the script regarding sending. Afterward, execution of the command is reported as successful to the process model. There will be no check whether the transfer has been successful and the data have been received by the weather service!

The new value sent will be reported back to the process model if the access rights **Write** or **Read/Write** have been set.

#### 3.3 Execution of Read Commands

When a "Read command" is invoked by the process model or the user, no requests will be created. Instead, the current value of the datapoint which is the value last received, will be reported back.

The current value will be reported back only if the access rights **Read** or **Read/Write** have been set.