Developer: Christopher McNeely
Source: https://github.com/professoraire/ElegantIFTTT-Crestron

Elegant IFTTT String Matching Module v1.0

GENERAL INFORMATION

SIMPL WINDOWS NAME	Elegant IFTTT String Matching Module v1.0
CATEGORY	Elegant Modules
VERSION	1.0
SUMMARY	Provides a method to parse string data from messages sent to the Crestron processor from IFTTT.
GENERAL NOTES	This module requires that an Elegant IFTTT Manager module be added to the program. This will handle parsing data received from an event with a matching EventName.
CRESTRON HARDWARE REQUIRED	3-Series Processor

PARAMETERS

ManagerID	S	The ID of the Elegant IFTTT Manager module to associate this module with.
EventName	S	The name of the event to parse information from. This module will only examine the received data if this event name matches the name received from the IFTTT service.
StringToParse (1-100)	S	A string value to look for in the incoming data. The Simpl# module converts all strings to lower-case for comparisons. Can use the following prefix qualifiers: • No prefix or = ○ Without any prefix, or with an = sign, the string has to match the entirety of at least one of the data elements. • ! ○ Placing an ! before the string will pulse the output if the string isn't equal to at least one of the data elements. • + ○ Placing a + before the string will pulse the output if the string is found anywhere in any of the data elements. This is partial matching and is like doing a Find operation in Simpl+. • - ○ Placing a – before the string will pulse the output if the string can't be found anywhere

		within any of the data elements. This is a partial matching operation. • >, <, >=, <= ○ Placing a standard mathematical comparison operator before the string will cause the module to parse the string as if it is a value and compare the value of the string to the value of the incoming data elements. The equations used are: StringToParse > DataElement StringToParse < DataElement StringToParse >= DataElement StringToParse <= DataElement The outputs are pulsed if the resulting equation evaluates to true.

CONTROL

Enable	D	When held high this allows the module to parse data. When low the module won't parse data, even if the event name matches.

FEEDBACK

EventTriggered (1-100)	D	The module evaluates each StringToParse element, checking for Prefixes. If the corresponding StringToParse element equation matches any of the data elements, the EventTriggered output is pulsed for 0.1 seconds. Multiple EventTriggered outputs can be fired by the same event.

ADDITIONAL DATA

Revision History	v1.0 – Initial Release
Additional Details	This module can fire multiple outputs for each event received from the IFTTT service. Rather than just evaluate for exactly matching string expressions, this module is also capable of checking for strings in the middle of data elements, non-existing strings, or evaluating the data as numerical and performing mathematical comparisons on them as well. The details of how to specify these are provided under the StringToParse parameter notes. If 100 outputs aren't sufficient for the quantity of string elements that need to be checked, you can place an additional String Matching Module in the program. Every Elegant IFTTT parsing module will evaluate event data when the Event's name matches their EventName parameter. SAMPLES • "=Goodnight" – The corresponding output will pulse when a data element is equal to "Goodnight". • "!Goodnight" – The corresponding output will pulse when no data element is equal to "Goodnight". • "+Goodnight" – The corresponding output will pulse when the string "goodnight" can

be found anywhere within a data element. (Such as: “Say goodnight to Crestron.”)

- “-Goodnight” – The corresponding output will pulse when the string “goodnight” cannot be found in any of the data elements. (Such as: “Say good morning to Crestron.”)
- “>10” – The corresponding output will pulse if any data element can be found that can be parsed to a numeric value and is greater than the value 10. This same mentality holds true for all other mathematical comparisons, <, >=, <=.