//Getting an event l_oSubDevice.GetEvent(l_nEventID); </syntaxhighlight> |attentions = }}

SubDevice.AddCondition (works via a function)

{{API Function/ru |description=Add a condition. A condition is an obligatory part of rules, but it can be used when it is necessary to make more comparisons before an action is done. This condition works via a function |constructor= SubDevice.AddCondition(Name, Token, ScriptCall, Method, Space, [Default], [AdvancedField]) |input=

`Name`	"Temperature is below..."	type: String Condition name
`Path`	"Drivers.Nest.CurrentTemperature"	type: String Full path to a channel, a driver feedback channel, a token. Subscription to keeping track of changes
`ScriptCall`	true	type: Boolean Feature of working via a function. Set true in this realization
`Method`	function(in_data){...}	type: Function Performed function. The coming parameter to the function can be the following: • If [Default] is used, the coming parameter is Default value in the String format set by a module devloper • If [AdvancedField] is used, the coming parameter is a value of Object or String type, depending on the [AdvancedField] format. The function must return boolean value (true, false), signaling that the condition is performed (true - condition is performed)
`Space`	this	type: Object Pointer to an object that services a function. If a pinter is not required, leave an empty object or undefined
`Default`	"23"	type: String Default value

{{JStable|AdvancedField||Object, Array|It is used to work with complicated conditions. When AdvancedField is used, put null in Default parameter. One object or an array of objects can be sent in AdvancedField.