提供巨集可用來定義:

巨集(Macros)
函式(function)
產生新 Metric

1. 產生新 Metric 的內容
2. 觸發的 HTTP Body(JSON) 的內容

- 以 `{<macro_name>}` 的語法代表要產生的巨集
- 以 `@func()` 的語法代表要使用的函式

EBNF

```
expr = {term};

term = text template
  | macro
  | function object

macro = "{", macro content , "}";

macro content = "tag", '[', { tag name } , ']' | alpha, {letter with dot};

function object = "@", function name, "(", [ function args ] , ")";

function name = alpha, { letter };

function args = argv, { ",", argv };

argv = expr | number | '"', text , '"';

number = ["+" | "-"], digit, {digit}, [ ".", digit, {digit} ];

letter with dot = "." | letter;
letter = "_" | alpha | digit;
alpha = 'a-zA-Z'; (* regexp *)
digit = '0-9'; (* regexp *)

tag name = [^]]; (* regular expression *)

text = "\\" | '\"' | any character;
text template =  "\{" | "\}" | "\@" | "\\" | any character
```

巨集(Macros)

在 產生新 Metric 可用的巨集

Macro	Description	Example
`{endpoint}`	The value of endpoint	`{endpoint}.ext1`
`{metric}`	The value of metric	`{metric}.ext1`
`{type}`	The value of type	`{type}.ext1`
`{step}`	The value of step	`{endpoint}.step.{step}`
`{value}`	The value of metric	`{value}`
`{tag[<tag_name>]}`	The value of tag	`{tag[cpu.seq]}`

在 觸發條件(橫向聚合) 可用的巨集

Macro	Description	Example
`{var[<var_name>]}`	The value of variable	`{var[recent.3]}.ext1`
`{filter.metric.name}`	The name of metric filter	`{trigger.name}.ext1`
`{trigger.name}`	The name of trigger	`{trigger.name}.ext1`

Macro	Description	Example
`{trigger.result}`	The value of result of trigger(be **true** or **false** value)	`{trigger.result}.ext1`
`{endpoint}`	The value of endpoint(last element)	`{endpoint}.ext1`
`{metric}`	The value of metric(last element)	`{metric}.ext1`
`{type}`	The value of type(last element)	`{type}.ext1`
`{step}`	The value of step(last element)	`{endpoint}.step.{step}`
`{value}`	The value of metric(last element)	`{value}`
`{tag[<tag_name>]}`	The value of tag(last element)	`{tag[cpu.seq]}`
`{tags}`	The value of tags(as JSON object)(last element)	`{tags}`

見 觸發動作定義

函式(function)

Function	Description	Example
`@if(<v>, <true value>, <false value>)`	Gives `<true value>` if `<v>` is viable, otherwise gives `<false value>`	`@if({tag[disk.hba]}, "disk", "non-disk")` - Gives **disk** if tag value is viable on "disk.hba". Otherwise gives "non-disk"
`@if2(<v>, <false value>)`	Gives `<v>` if `<v>` is viable, otherwise gives `<false value>`	`@if2({tag[disk.hba]}, "non-disk")` - Gives **value of tag[disk.hba]** if tag value is viable on "disk.hba". Otherwise gives "non-disk"
`@replace(<v>, <search>, <replacing>)`	Replaces all of the `<search>` string with `<replaceing>` on `<v>`	`@replace("cpu.idc.v1", ".idc.", ".intel.")` - Gives **"cpu.intel.v1"**
`@strippprefix(<v>, <prefix>)`	Strips the prefix of `<prefix>` on `<v>`	`@strippprefix("apple-key", "apple-")` - Gives **"key"**
`@stripsuffix(<v>, <suffix>)`	Strips the suffix of `<suffix>` on `<v>`	`@stripsuffix("apple-key", "-key")` - Gives **"apple"**

產生新 Metric

由單一 Metric 產生

透過 過濾語言 所符合的條件，符合的 metrics 會產生新的 metric

由 聚合語言 產生

透過 聚合語言 所符合的條件，符合的 metrics 會產生新的 metric

Last modified on 2017-07-24T16:02:11+08:00