

Package `mil.army.swf.micrometer.metrics.helper`

# Class MetricHelper

`java.lang.Object`  
`mil.army.swf.micrometer.metrics.helper.MetricHelper`

`@Service`  
`public class MetricHelper`  
`extends Object`



Helper class for creating and updating metrics that will be scrapped by Prometheus

## Method Summary

All Methods    Static Methods    Concrete Methods

Modifier and Type	Method	Description
static void	<code><b>createMultiTagCounter</b>(<code>String</code> name, <code>String</code> description, <code>String</code> [] tagNames, <code>String</code> [] tagValues)</code>	Creates a multi-tag micrometer counter
static void	<code><b>createMultiTagGauge</b>(<code>String</code> name, <code>String</code> description, <code>String</code> [] tagNames, <code>String</code> [] tagValues)</code>	Creates a multi-tagged tagged micrometer gauge
static void	<code><b>createMultiTagHistogramExpectedValueS</b>(<code>String</code> name, <code>String</code> description, <code>TimerTypes.TimerType</code> type, <code>Duration</code> slosla, <code>Duration</code> min, <code>Duration</code> max, <code>String</code> [] tagNames, <code>String</code> [] tagValues)</code>	Create a multi-tagged slo/sla timer
static void	<code><b>createMultiTagHistogramExpectedValueT</b>(<code>String</code> name, <code>String</code> description,</code>	Create a multi-tagged expected value timer

	<b>TimerTypes.TimerType</b> type, <b>Duration</b> min, <b>Duration</b> max, <b>String</b> [] tagNames, <b>String</b> [] tagValues)	
static void	<b>createMultiTagHistogramTimer(String</b> name, <b>String</b> description, <b>String</b> [] tagNames, <b>String</b> [] tagValues)	Create a multi-tagged histogram timer
static void	<b>createMultiTagTimer(String</b> name, <b>String</b> description, <b>String</b> [] tagNames, <b>String</b> [] tagValues)	Create multi-tagged timer
static void	<b>createTagCounter(String</b> name, <b>String</b> description, <b>String</b> tagName, <b>String</b> tagValue)	Creates a tagged micrometer counter
static void	<b>createTagGauge(String</b> name, <b>String</b> description, <b>String</b> tagName, <b>String</b> tagValue)	Creates a tagged micrometer gauge
static void	<b>createTagHistogramExpectedValueSloSla</b> ( <b>String</b> name, <b>String</b> description, <b>TimerTypes.TimerType</b> type, <b>Duration</b> slosla, <b>Duration</b> min, <b>Duration</b> max, <b>String</b> tagName, <b>String</b> tagValue)	Creates a histogram timer with expected value and service level objective or service level agreement
static void	<b>createTagHistogramExpectedValueTimer</b> ( <b>String</b> name, <b>String</b> description, <b>TimerTypes.TimerType</b> type, <b>Duration</b> min, <b>Duration</b> max, <b>String</b> tagName, <b>String</b> tagValue)	Creates a histogram timer with expected values
static void	<b>createTagHistogramTimer(String</b> name, <b>String</b> description, <b>String</b> tagName, <b>String</b> tagValue)	Creates a histogram timer

static void	<b>createTagTimer</b> ( <b>String</b> name, <b>String</b> description, <b>String</b> tagName, <b>String</b> tagValue)	Creates a tagged micrometer timer
static io.micrometer.prometheus.Prometh	<b>getRegistry</b> ()	Get the meter registry
		
static void	<b>incrementMultiTagCounter</b> ( <b>String</b> name)	Increments multi-tag counter
static void	<b>incrementMultiTagCounter</b> ( <b>String</b> name, long increment)	Increments a multi-tag counter with an increment value
static void	<b>incrementTagCounter</b> ( <b>String</b> name)	Increments a defined tag counter
static void	<b>incrementTagCounter</b> ( <b>String</b> name, long increment)	Increments a defined tagged counter with supplied incremental value
static void	<b>recordMultiTagTime</b> ( <b>String</b> name, <b>Duration</b> duration)	Records the time for a multi-tagged timer
static void	<b>recordMultiTagTime</b> ( <b>String</b> name, <b>TimeUnit</b> timeUnit, long timeValue)	Records the time for a multi-tagged timer
static void	<b>recordTagTime</b> ( <b>String</b> name, <b>Duration</b> duration)	Records the time for a tagged timer
static void	<b>recordTagTime</b> ( <b>String</b> name, <b>TimeUnit</b> timeUnit, long timeValue)	Records the time for a tagged timer
static void	<b>setMultiTagGauge</b> ( <b>String</b> name, double value)	Set the value of a defined multi-tagged gauge
static void	<b>setRegistry</b> (io.micrometer.prometheus.PrometheusM	Sets the prometheus registry
		
static void	<b>setTagGauge</b> ( <b>String</b> name, double value)	Set the value of a defined tagged gauge

## Methods inherited from class java.lang.Object

`clone` , `equals` , `finalize` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

## Method Details

### getRegistry

```
public static io.micrometer.prometheus.PrometheusMeterRegistry getRegistry()
```

Get the meter registry

**Returns:**

meter registry

### setRegistry

```
public static void setRegistry(io.micrometer.prometheus.PrometheusMeterRegistry registry1)
```

Sets the prometheus registry

**Parameters:**

registry1 - prometheus registry

### createTagCounter

```
public static void createTagCounter(String name,  
                                   String description,  
                                   String tagName,  
                                   String tagValue)
```

Creates a tagged micrometer counter

**Parameters:**

name - meter name

description - meter description

tagName - tag name associated with timer

tagValue - tag value associated with timer

### incrementTagCounter

```
public static void incrementTagCounter(String name,
                                       long increment)
```

Increments a defined tagged counter with supplied incremental value

**Parameters:**

name - meter name

increment - increment value

### incrementTagCounter

```
public static void incrementTagCounter(String name)
```

Increments a defined tag counter

**Parameters:**

name - meter name

### createMultiTagCounter

```
public static void createMultiTagCounter(String name,
                                          String description,
                                          String [] tagNames,
                                          String [] tagValues)
```

Creates a multi-tag micrometer counter

**Parameters:**

name - meter name

description - meter description

tagNames - string array of tag names associated with timer

tagValues - string array of tag values associated with timer

### incrementMultiTagCounter

```
public static void incrementMultiTagCounter(String name,  
                                             long increment)
```

Increments a multi-tag counter with an increment value

**Parameters:**

name - meter name

increment - increment number

### incrementMultiTagCounter

```
public static void incrementMultiTagCounter(String name)
```

Increments multi-tag counter

**Parameters:**

name - meter name

### createTagGauge

```
public static void createTagGauge(String name,  
                                String description,  
                                String tagName,  
                                String tagValue)
```

Creates a tagged micrometer gauge

**Parameters:**

name - meter name

description - meter description

tagName - tag name associated with timer

tagValue - tag value associated with timer

## setTagGauge

```
public static void setTagGauge(String name,  
                               double value)
```

Set the value of a defined tagged gauge

**Parameters:**

name - meter name

value - the value to set the gauge

## createMultiTagGauge

```
public static void createMultiTagGauge(String name,  
                                       String description,  
                                       String [] tagNames,  
                                       String [] tagValues)  
    throws IllegalArgumentException
```

Creates a multi-tagged tagged micrometer gauge

**Parameters:**

`name` - meter name

`description` - meter description

`tagNames` - string array of tag names associated with timer

`tagValues` - string array of tag values associated with timer

**Throws:**

`IllegalArgumentException` - `tagName` and `tagValue` array lengths are unequal

**setMultiTagGauge**

```
public static void setMultiTagGauge(String name,
                                     double value)
```

Set the value of a defined multi-tagged gauge

**Parameters:**

`name` - meter name

`value` - the value to set the gauge

**createTagTimer**

```
public static void createTagTimer(String name,
                                   String description,
                                   String tagName,
                                   String tagValue)
```

Creates a tagged micrometer timer

**Parameters:**

`name` - meter name

`description` - meter description

`tagName` - tag name associated with timer



tagValue - tag value associated with timer

### createTagHistogramTimer

```
public static void createTagHistogramTimer(String name,  
                                           String description,  
                                           String tagName,  
                                           String tagValue)
```

Creates a histogram timer

**Parameters:**

name - meter name

description - meter description

tagName - tag name associated with timer

tagValue - tag value associated with timer

### createTagHistogramExpectedValueTimer

```
public static void createTagHistogramExpectedValueTimer(String name,  
                                                         String description,  
                                                         TimerTypes.TimerType type,  
                                                         Duration min,  
                                                         Duration max,  
                                                         String tagName,  
                                                         String tagValue)
```

Creates a histogram timer with expected values

**Parameters:**

name - meter name

description - meter description

type - timer type

`min` - minimum expected value

`max` - maximum expected value

`tagName` - tag name associated with timer

`tagValue` - tag value associated with timer

### **createTagHistogramExpectedValueSloSlaTimer**

```
public static void createTagHistogramExpectedValueSloSlaTimer(String name,
                                                             String description,
                                                             TimerTypes.TimerType type,
                                                             Duration slosla,
                                                             Duration min,
                                                             Duration max,
                                                             String tagName,
                                                             String tagValue)
    throws InvalidOptionType
```

Creates a histogram timer with expected value and service level objective or service level agreement

#### **Parameters:**

`name` - meter name

`description` - meter description

`type` - timer type

`slosla` - duration for SLA/SLO

`min` - minimum expected value

`max` - maximum expected value

`tagName` - tag name associated with timer

`tagValue` - tag value associated with timer

#### **Throws:**

`InvalidOptionType` - invalid timer type specified

## recordTagTime

```
public static void recordTagTime(String name,
                                Duration duration)
```

Records the time for a tagged timer

**Parameters:**

name - meter name

duration - duration of time to record

## recordTagTime

```
public static void recordTagTime(String name,
                                TimeUnit timeUnit,
                                long timeValue)
    throws InvalidTimeUnitException
```

Records the time for a tagged timer

**Parameters:**

name - meter name

timeUnit - time unit of the time value

timeValue - time value

**Throws:**

`InvalidTimeUnitException` - invalid time unit specified

## createMultiTagTimer

```
public static void createMultiTagTimer(String name,
                                       String description,
                                       String [] tagNames,
                                       String [] tagValues)
```

## Create multi-tagged timer

### Parameters:

name - meter name

description - meter description

tagNames - string array of tag names associated with timer

tagValues - string array of tag values associated with timer

### createMultiTagHistogramTimer

```
public static void createMultiTagHistogramTimer(String name,
                                                String description,
                                                String [] tagNames,
                                                String [] tagValues)
```

## Create a multi-tagged histogram timer

### Parameters:

name - meter name

description - meter description

tagNames - string array of tag names associated with timer

tagValues - string array of tag values associated with timer

### createMultiTagHistogramExpectedValueTimer

```
public static void createMultiTagHistogramExpectedValueTimer(String name,
                                                             String description,
                                                             TimerTypes.TimerType type,
                                                             Duration min,
                                                             Duration max,
                                                             String [] tagNames,
                                                             String [] tagValues)
```

Create a multi-tagged expected value timer

**Parameters:**

name - meter name

description - meter description

type - timer type

min - minimum expected value

max - maximum expected value

tagNames - string array of tag names associated with timer

tagValues - tring array of tag values associated with timer

### createMultiTagHistogramExpectedValueSloSlaTimer

```
public static void createMultiTagHistogramExpectedValueSloSlaTimer(String name,
                                                                    String description,
                                                                    TimerTypes.TimerType type,
                                                                    Duration slosla,
                                                                    Duration min,
                                                                    Duration max,
                                                                    String [] tagNames,
                                                                    String [] tagValues)
    throws InvalidOptionType
```

Create a multi-tagged slo/sla timer

**Parameters:**

name - meter name

description - meter description

type - timer type

slosla - duration for SLA/SLO

min - minimum expected value

`max` - maximum expected value

`tagNames` - string array of tag names associated with timer

`tagValues` - string array of tag values associated with timer

**Throws:**

`InvalidOptionType` - timer `tagName` and `tagValue` array are unequal

## recordMultiTagTime

```
public static void recordMultiTagTime(String name,
                                     Duration duration)
```

Records the time for a multi-tagged timer

**Parameters:**

`name` - meter name

`duration` - duration of time to record

## recordMultiTagTime

```
public static void recordMultiTagTime(String name,
                                     TimeUnit timeUnit,
                                     long timeValue)
    throws InvalidTimeUnitException
```

Records the time for a multi-tagged timer

**Parameters:**

`name` - meter name

`timeUnit` - time unit of the time value

`timeValue` - time value

**Throws:**

`InvalidTimeUnitException` - - invalid time unit specified

---

Copyright © 2023. All rights reserved.