Microprofile custom metrics

1. @Metered - This annotation is used to track the rate of occurrence of a specific event, such as method invocations, and to measure the rate at which it happens over time.

Viewing Metrics

When exposed, the @Metered metrics will be available in the standard metrics format. For example:

# TYPE myMethodMeter meter

myMethodMeter\_count 123

myMethodMeter\_one\_minute\_rate 0.5

myMethodMeter\_five\_minute\_rate 0.4

myMethodMeter\_fifteen\_minute\_rate 0.3

 **\_count**: The total number of method invocations.

 **\_one\_minute\_rate**: The rate of method invocations per second over the last minute.

 **\_five\_minute\_rate**: The rate over the last five minutes.

 **\_fifteen\_minute\_rate**: The rate over the last fifteen minutes

**Best Practices**

* **Granularity:** Use @Metered for methods or code blocks that are critical for performance monitoring. Avoid overusing it on methods that are called very frequently or infrequently, as it may generate excessive metrics data.
* **Naming:** Choose clear and descriptive names for your metrics to easily understand what each one tracks.
* **Monitoring:** Integrate with monitoring tools like Prometheus and Grafana to visualize and analyze these metrics effectively.

By using @Metered, you can gain insights into how often certain operations occur in your application, which can help in performance tuning and understanding the load on your system.

1. @Timed - The @Timed annotation in MicroProfile Metrics is used to measure the duration of method executions. This can be very useful for understanding the performance characteristics of your application by tracking how long specific methods or operations take to complete.

**Metrics Captured:** It captures:

* **Elapsed Time:** How long the annotated method took to execute.
* **Count:** The number of times the method was invoked.
* **Duration Statistics:** Such as average time, maximum time, etc.

# TYPE application\_getAllOutlets\_timer\_rate\_per\_second gauge

application\_getAllOutlets\_timer\_rate\_per\_second 0.012173763739845482

# TYPE application\_getAllOutlets\_timer\_one\_min\_rate\_per\_second gauge

application\_getAllOutlets\_timer\_one\_min\_rate\_per\_second 1.2101932603692978E-5

# TYPE application\_getAllOutlets\_timer\_five\_min\_rate\_per\_second gauge

application\_getAllOutlets\_timer\_five\_min\_rate\_per\_second 0.003674604233960916

# TYPE application\_getAllOutlets\_timer\_fifteen\_min\_rate\_per\_second gauge

application\_getAllOutlets\_timer\_fifteen\_min\_rate\_per\_second 0.00419756049199671

# TYPE application\_getAllOutlets\_timer\_mean\_seconds gauge

application\_getAllOutlets\_timer\_mean\_seconds 0.3510588669847899

# TYPE application\_getAllOutlets\_timer\_max\_seconds gauge

application\_getAllOutlets\_timer\_max\_seconds 2.7942192550000002

# TYPE application\_getAllOutlets\_timer\_min\_seconds gauge

application\_getAllOutlets\_timer\_min\_seconds 0.004344781000000001

# TYPE application\_getAllOutlets\_timer\_stddev\_seconds gauge

application\_getAllOutlets\_timer\_stddev\_seconds 0.8532719210231847

# TYPE application\_getAllOutlets\_timer\_seconds summary

# HELP application\_getAllOutlets\_timer\_seconds Time taken by getgetAllOutlets.

application\_getAllOutlets\_timer\_seconds\_count 7

application\_getAllOutlets\_timer\_seconds{quantile="0.5"} 0.010334572

application\_getAllOutlets\_timer\_seconds{quantile="0.75"} 0.150438177

application\_getAllOutlets\_timer\_seconds{quantile="0.95"} 2.7942192550000002

application\_getAllOutlets\_timer\_seconds{quantile="0.98"} 2.7942192550000002

application\_getAllOutlets\_timer\_seconds{quantile="0.99"} 2.7942192550000002

application\_getAllOutlets\_timer\_seconds{quantile="0.999"} 2.7942192550000002