

# OpenTelemetry Metrics

OpenTelemetry Metrics OpenTelemetry APM (Uptrace Prometheus)

## Instruments

Instrument (counter, gauge, histogram)

```
instrument :  
  
  • http.server.duration  
  • Histogram  
  • milliseconds bytes  
  •
```

## Timeseries()

instrument

## Synchronous()

Synchronous instruments counter.Add(ctx, 1) trace context

| Instrument    | Properties | Aggregation       | Example |
|---------------|------------|-------------------|---------|
| Counter       |            | sum -> delta      |         |
| UpDownCounter |            | last value -> sum |         |
| Histogram     |            | histogram         |         |

## Asynchronous()

Asynchronous instruments CPU Asynchronous trace context

| Instrument Name       | Properties | Aggregation            | Example                |
|-----------------------|------------|------------------------|------------------------|
| CounterObserver       |            | sum -> delta           | CPU time               |
| UpDownCounterObserver |            | last value -> sum      | Memory usage (bytes)   |
| GaugeObserver         |            | last value -> none/avg | Memory utilization (%) |

### Content Menu

- [Instruments](#)
- 
-

```

func main() {
    initMeter()

    meter := global.Meter("ex.com/basic")
    observerLock := new(sync.RWMutex)
    observerValueToReport := new(float64)
    observerLabelsToReport := new([]attribute.KeyValue)
    cb := func(_ context.Context, result metric.Float64ObserverResult) {
        (*observerLock).RLock()
        value := *observerValueToReport
        labels := *observerLabelsToReport
        (*observerLock).RUnlock()
        result.Observe(value, labels...)
    }

    _ = metric.Must(meter).NewFloat64GaugeObserver("ex.com.one", cb,
        metric.WithDescription("A GaugeObserver set to 1.0"),
    )

    histogram := metric.Must(meter).NewFloat64Histogram("ex.com.two")
    counter := metric.Must(meter).NewFloat64Counter("ex.com.three")

    commonLabels := []attribute.KeyValue{lemonsKey.Int(10), attribute.String("A", "1")}
    ctx := context.Background()

    (*observerLock).Lock()
    *observerValueToReport = 1.0
    *observerLabelsToReport = commonLabels
    (*observerLock).Unlock()
    meter.RecordBatch(
        ctx,
        commonLabels,
        histogram.Measurement(2.0),
        counter.Measurement(12.0),
    )
}

```

```

# HELP ex_com_one A GaugeObserver set to 1.0
# TYPE ex_com_one gauge
ex_com_one 1.0
# HELP ex_com_two A HistogramObserver set to 1.0
# TYPE ex_com_two histogram
ex_com_two_bucket{le="2",ex_com_three="10",service_name="unknown_service:prometheus",telemetry_ab_language="go",telemetry_ab_name="opentelemetry",telemetry_ab_version="1.0.1"} 18
ex_com_two_bucket{le="2",ex_com_three="10",service_name="unknown_service:prometheus",telemetry_ab_language="go",telemetry_ab_name="opentelemetry",telemetry_ab_version="1.0.1"} 1
# HELP ex_com_three A CounterObserver set to 1.0
# TYPE ex_com_three counter
ex_com_three 12.0
# HELP ex_com_four A HistogramObserver set to 1.0
# TYPE ex_com_four histogram
ex_com_four_bucket{le="2",ex_com_three="10",service_name="unknown_service:prometheus",telemetry_ab_language="go",telemetry_ab_name="opentelemetry",telemetry_ab_version="1.0.1"} 12
ex_com_four_bucket{le="2",ex_com_three="10",service_name="unknown_service:prometheus",telemetry_ab_language="go",telemetry_ab_name="opentelemetry",telemetry_ab_version="1.0.1"} 1
# HELP ex_com_five A CounterObserver set to 1.0
# TYPE ex_com_five counter
ex_com_five 12.0
# HELP ex_com_six A HistogramObserver set to 1.0
# TYPE ex_com_six histogram
ex_com_six_bucket{le="2",ex_com_three="10",service_name="unknown_service:prometheus",telemetry_ab_language="go",telemetry_ab_name="opentelemetry",telemetry_ab_version="1.0.1"} 12
ex_com_six_bucket{le="2",ex_com_three="10",service_name="unknown_service:prometheus",telemetry_ab_language="go",telemetry_ab_name="opentelemetry",telemetry_ab_version="1.0.1"} 1

```

- <https://opentelemetry.io/docs/specs/otel/metrics/data-model/>
- <https://github.com/open-telemetry/opentelemetry-go/tree/main/metric>