

```

metrics readermetrics readerCPU

gmetricObservableCounter, ObservableUpDownCounter, ObservableGaugeObservable

package main

import (
    "context"

    "go.opentelemetry.io/otel/exporters/prometheus"

    "github.com/gogf/gf/contrib/metric/otelmetric/v2"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/os/gmetric"
)

var (
    meter = gmetric.GetGlobalProvider().Meter(gmetric.MeterOption{
        Instrument:           "github.com/gogf/gf/example/metric/basic",
        InstrumentVersion:    "v1.0",
    })
    observableCounter = meter.MustObservableCounter(
        "goframe.metric.demo.observable_counter",
        gmetric.MetricOption{
            Help: "This is a simple demo for ObservableCounter usage",
            Unit: "%",
        },
    )
    observableUpDownCounter = meter.MustObservableUpDownCounter(
        "goframe.metric.demo.observable_updown_counter",
        gmetric.MetricOption{
            Help: "This is a simple demo for ObservableUpDownCounter usage",
            Unit: "%",
        },
    )
    observableGauge = meter.MustObservableGauge(
        "goframe.metric.demo.observable_gauge",
        gmetric.MetricOption{
            Help: "This is a simple demo for ObservableGauge usage",
            Unit: "%",
        },
    )
)

func main() {
    var ctx = gctx.New()

    // Callback for observable metrics.
    meter.MustRegisterCallback(func(ctx context.Context, obs gmetric.Observer) error {
        obs.Observe(observableCounter, 10)
        obs.Observe(observableUpDownCounter, 20)
        obs.Observe(observableGauge, 30)
        return nil
    }, observableCounter, observableUpDownCounter, observableGauge)

    // Prometheus exporter to export metrics as Prometheus format.
    exporter, err := prometheus.New(

```

## Content Menu

- [Meter Callback](#)
- [Metric Callback](#)
- [Primetheus Exporter](#)

```

        prometheus.WithoutCounterSuffixes(),
        prometheus.WithoutUnits(),
    )
    if err != nil {
        g.Log().Fatal(ctx, err)
    }

    // OpenTelemetry provider.
    provider := otelmetric.MustProvider(otelmetric.WithReader
(exporter))
    provider.SetAsGlobal()
    defer provider.Shutdown(ctx)

    // HTTP Server for metrics exporting.
    otelmetric.StartPrometheusMetricsServer(8000, "/metrics")
}

```

## Meter Callback

CallbackCallbackCallbackObserveObserve

- ObservableCounter/ObservableUpDownCounterObserve
- ObservableGaugeObserveObserve

## Metric Callback

Meter CallBackMetricOptionCallback

```

observableCounter = meter.MustObservableCounter(
    "goframe.metric.demo.observable_counter",
    gmetric.MetricOption{
        Help: "This is a simple demo for ObservableCounter usage",
        Unit: "%",
        Callback: func(ctx context.Context, obs gmetric.
MetricObserver) error {
            obs.Observe(10)
            return nil
        },
    },
)

```

## Primetheus Exporter

Prometheus

```
otelmetric.StartPrometheusMetricsServer(8000, "/metrics")
```

<http://127.0.0.1:8000/metrics>

```

# HELP gopfrcache_metric_demo_observable_counter This is a simple demo for ObservabiltyCounter usage
# TYPE gopfrcache_metric_demo_observable_counter counter
gopfrcache_metric_demo_observable_counter{gitlab.com/gopfrcache/example/metric/basic",otel_scope_version="v1.1"} 10
# HELP gopfrcache_metric_demo_observable_gauge gauge
# TYPE gopfrcache_metric_demo_observable_gauge gauge
gopfrcache_metric_demo_observable_gauge{gitlab.com/gopfrcache/example/metric/basic",otel_scope_version="v1.0"} 20
# HELP gopfrcache_metric_demo_observable_spdyx_counter This is a simple demo for ObservabiltyCounter usage
# TYPE gopfrcache_metric_demo_observable_spdyx_counter counter
gopfrcache_metric_demo_observable_spdyx_counter{gitlab.com/gopfrcache/example/metric/basic",otel_scope_version="v1.0"} 20
# HELP gopfrcache_metric_demo_observable_update_counter This is a simple demo for ObservabiltyCounter usage
# TYPE gopfrcache_metric_demo_observable_update_counter counter
gopfrcache_metric_demo_observable_update_counter{gitlab.com/gopfrcache/example/metric/basic",otel_scope_version="v1.0"} 20
# HELP http_server_requests_total Total number of requests received by the server
# TYPE http_server_requests_total counter
http_server_requests_total{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 1
# HELP http_server_requests_body_size_bytes Incoming request bytes total
http_server_requests_body_size_bytes{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 4
# HELP http_server_requests_duration_seconds Duration histogram
# TYPE http_server_requests_duration_seconds histogram
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="0.01-0.1"} 1
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="0.1-1"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="1-10"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="10-100"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="100-1000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="1000-10000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="10000-100000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="100000-1000000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="1000000-10000000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="10000000-100000000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="100000000-1000000000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="1000000000-10000000000"} 2
http_server_requests_duration_seconds{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1",bin="10000000000-100000000000"} 2
# HELP http_server_requests_duration_total Total duration of all requests received by the server
# TYPE http_server_requests_duration_total counter
http_server_requests_duration_total{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 1
# HELP http_server_requests_total_total Total number of requests received by the server
# TYPE http_server_requests_total_total counter
http_server_requests_total_total{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 1
# HELP http_server_requests_body_size_bytes_total Incoming request bytes total
http_server_requests_body_size_bytes_total{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 4
# HELP http_server_requests_duration_mean Mean duration of requests received by the server
# TYPE http_server_requests_duration_mean gauge
http_server_requests_duration_mean{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 194
# HELP http_server_requests_duration_p50 P50 duration of requests received by the server
# TYPE http_server_requests_duration_p50 gauge
http_server_requests_duration_p50{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 194
# HELP http_server_requests_duration_p99 P99 duration of requests received by the server
# TYPE http_server_requests_duration_p99 gauge
http_server_requests_duration_p99{method="GET",http_code="200",server_address="127.0.0.1",server_port="8000",otel_scope_name="gitlab.com/gopfrcache/v1.1"} 194
# HELP target_info Target metadata
# TYPE target_info gauge
target_info{service_name="unknown_service",__log_build_github_com_gopfrcache_example_test__,otel_scope_name="gitlab.com/gopfrcache/v1.1",otel_scope_version="1.24.0"} 1

```