ShoeHub Exporter Documentation

Overview

This document provides a comprehensive overview of all metrics exported by the ShoeHub application. The project is instrumented with the prometheus-net.AspNetCore library, which automatically exposes a set of standard HTTP metrics in addition to the custom business metrics defined in the application code.

Custom Business Metrics

These metrics are defined in the Program.cs file and are designed for a deep understanding of business-specific performance.

shoehub_sales (Counter)

A cumulative counter that tracks the total number of shoes sold.

- **Labels:** ShoeType
  - Loafers
  - Boots
  - HighHeels

PromQL Queries:

- Total sales for each shoe type:
  shoehub_sales
- Total sales for "Loafers" only:
  shoehub_sales{ShoeType="Loafers"}
- Sum of all sales across all shoe types:
  sum(shoehub_sales)

shoehub_payments (Gauge)

A gauge that records a random, instantaneous payment value. The value can go up or down.

- **Labels:** CountryCode and PaymentMethod
  - CountryCode: AU, US, IN
  - PaymentMethod: Card, Cash, Paypal

PromQL Queries:

- Current payment value for each combination of labels:
  shoehub_payments
- Payments via "Paypal" in "AU":
  shoehub_payments{CountryCode="AU", PaymentMethod="Paypal"}
- Average payments by country:
  avg by (CountryCode) (shoehub_payments)

- Total sum of all current payments:
  sum(shoehub_payments)

Automatic HTTP Metrics

These are standard metrics provided by the prometheus-net.AspNetCore library's app.UseHttpMetrics() middleware. You can use these to monitor the performance of your web API. The metric names listed below are the most common standard, but may vary slightly depending on the library version. You can confirm the exact names by visiting http://localhost:8080/metrics.

HTTP Request Counter

A counter that tracks the total number of HTTP requests processed by your application.

- **Metric Name:** http_requests_total
- **Labels:** method, path, status_code

PromQL Queries:

- Overall HTTP request rate over 5 minutes:
  rate(http_requests_total[5m])
- Error rate (5xx status codes) over 5 minutes:
  rate(http_requests_total{status_code=~"5.."}[5m])

HTTP Request Duration (Histogram)

A histogram that records the duration of HTTP requests in seconds. This provides detailed latency information.

- **Metric Names:**
  - http_request_duration_seconds_count (Total number of requests)
  - http_request_duration_seconds_sum (Sum of all request durations)
  - http_request_duration_seconds_bucket (Request counts by time bucket)

PromQL Queries:

- 99th Percentile Latency over 5 minutes:
  histogram_quantile(0.99, rate(http_request_duration_seconds_bucket[5m]))
- Average latency over 5 minutes:
  rate(http_request_duration_seconds_sum[5m]) /
  rate(http_request_duration_seconds_count[5m])

Other Automatically Exposed Metrics

The prometheus-net library also exposes a range of standard process metrics that give you insight into the health of the application's host environment. These metrics are available by default and can be a valuable part of your monitoring.

- process_cpu_seconds_total: Cumulative CPU time spent by the process.
- process_virtual_memory_bytes: Virtual memory size in bytes.

- process_resident_memory_bytes: Resident memory size in bytes.
- dotnet_threadpool_threads_count: Number of threads in the .NET thread pool.
- dotnet_gc_collections_total: Total number of garbage collections.