
prometheus-cpp Documentation

Gregor Jasny

Oct 25, 2018

Contents:

1	Indices and tables	1
2	Doxygen Index	3

CHAPTER 1

Indices and tables

- `genindex`
- `modindex`
- `search`

struct Bucket

Public Members

std::uint64_t **cumulative_count** = 0

double **upper_bound** = 0.0

class CKMSQuantiles

Public Functions

CKMSQuantiles (**const** std::vector<[Quantile](#)> &*quantiles*)

void **insert** (double *value*)

double **get** (double *q*)

void **reset** ()

Private Functions

double **allowableError** (int *rank*)

bool **insertBatch** ()

void **compress** ()

Private Members

```
const std::reference_wrapper<const std::vector<Quantile>> quantiles_  
std::size_t count_  
std::vector<Item> sample_  
std::array<double, 500> buffer_  
std::size_t buffer_count_
```

```
struct ClientMetric
```

Public Members

```
std::vector<Label> label  
Counter counter  
Gauge gauge  
Summary summary  
Histogram histogram  
Untyped untyped  
std::int64_t timestamp_ms = 0
```

```
class Collectable
```

Subclassed by *prometheus::Family< T >*, *prometheus::Registry*

Public Functions

```
virtual ~Collectable ()  
virtual std::vector<MetricFamily> Collect () = 0
```

```
class Counter
```

Public Functions

```
void Increment ()  
void Increment (double)  
double Value () const  
ClientMetric Collect ()
```

Public Static Attributes

```
const MetricType metric_type = MetricType::Counter
```


Private Members

Gauge **gauge_**

struct Counter

Public Members

double **value** = 0.0

class CounterBuilder

Public Functions

CounterBuilder &**Labels** (**const** std::map<std::string, std::string> &*labels*)

CounterBuilder &**Name** (**const** std::string&)

CounterBuilder &**Help** (**const** std::string&)

Family<Counter> &**Register** (*Registry*&)

Private Members

std::map<std::string, std::string> **labels_**

std::string **name_**

std::string **help_**

class Exposer

Public Functions

Exposer (**const** std::string &*bind_address*, **const** std::string &*uri* = std::string("/metrics"))

~Exposer ()

void **RegisterCollectable** (**const** std::weak_ptr<*Collectable*> &*collectable*)

Private Members

std::unique_ptr<CivetServer> **server_**

std::vector<std::weak_ptr<*Collectable*>> **collectables_**

std::shared_ptr<*Registry*> **exposer_registry_**

std::unique_ptr<*detail::MetricsHandler*> **metrics_handler_**

std::string **uri_**

template <typename *T*>

class Family: **public** *prometheus::Collectable*

Public Functions

Family (**const** std::string &name, **const** std::string &help, **const** std::map<std::string, std::string> &constant_labels)

template <typename... Args>

T &Add (**const** std::map<std::string, std::string> &labels, Args&&... args)

void **Remove** (T *metric)

std::vector<MetricFamily> **Collect** ()

Private Functions

ClientMetric **CollectMetric** (std::size_t hash, T *metric)

Private Members

std::unordered_map<std::size_t, std::unique_ptr<T>> **metrics_**

std::unordered_map<std::size_t, std::map<std::string, std::string>> **labels_**

std::unordered_map<T *, std::size_t> **labels_reverse_lookup_**

const std::string **name_**

const std::string **help_**

const std::map<std::string, std::string> **constant_labels_**

std::mutex **mutex_**

Private Static Functions

std::size_t **hash_labels** (**const** std::map<std::string, std::string> &labels)

class Gateway

Public Types

using Labels = std::map<std::string, std::string>

Public Functions

Gateway (**const** std::string &uri, **const** std::string jobname, **const** Labels &labels = {}, **const** std::string username = {}, **const** std::string password = {})

void **RegisterCollectable** (**const** std::weak_ptr<Collectable> &collectable, **const** Labels *labels = nullptr)

int **Push** ()

std::future<int> **AsyncPush** ()

int **PushAdd** ()

```
std::future<int> AsyncPushAdd ()
```

```
int Delete ()
```

```
std::future<int> AsyncDelete ()
```

Public Static Functions

```
static const Labels GetInstanceLabel (std::string hostname)
```

Private Types

```
enum PushMode
```

Values:

```
    Add
```

```
    Replace
```

Private Functions

```
int push (PushMode mode)
```

```
std::future<int> async_push (PushMode mode)
```

Private Members

```
std::string jobUri_
```

```
std::string labels_
```

```
std::string username_
```

```
std::string password_
```

```
std::vector<std::pair<std::weak_ptr<Collectable>, std::string>> collectables_
```

```
struct Gauge
```

Public Members

```
double value = 0.0
```

```
class Gauge
```

Public Functions

```
Gauge ()
```

```
Gauge (double)
```

```
void Increment ()
```

```
void Increment (double)
```

```
void Decrement ()  
void Decrement (double)  
void Set (double)  
void SetToCurrentTime ()  
double Value () const  
ClientMetric Collect ()
```

Public Static Attributes

```
const MetricType metric_type = MetricType::Gauge
```

Private Functions

```
void Change (double)
```

Private Members

```
std::atomic<double> value_
```

```
class GaugeBuilder
```

Public Functions

```
GaugeBuilder &Labels (const std::map<std::string, std::string> &labels)
```

```
GaugeBuilder &Name (const std::string&)
```

```
GaugeBuilder &Help (const std::string&)
```

```
Family<Gauge> &Register (Registry&)
```

Private Members

```
std::map<std::string, std::string> labels_
```

```
std::string name_
```

```
std::string help_
```

```
struct Histogram
```

Public Members

```
std::uint64_t sample_count = 0
```

```
double sample_sum = 0.0
```

```
std::vector<Bucket> bucket
```

```
class Histogram
```

Public Types

```
using BucketBoundaries = std::vector<double>
```

Public Functions

```
Histogram (const BucketBoundaries &buckets)
```

```
void Observe (double value)
```

```
ClientMetric Collect ()
```

Public Static Attributes

```
const MetricType metric_type = MetricType::Histogram
```

Private Members

```
const BucketBoundaries bucket_boundaries_
```

```
std::vector<Counter> bucket_counts_
```

```
Counter sum_
```

```
class HistogramBuilder
```

Public Functions

```
HistogramBuilder &Labels (const std::map<std::string, std::string> &labels)
```

```
HistogramBuilder &Name (const std::string&)
```

```
HistogramBuilder &Help (const std::string&)
```

```
Family<Histogram> &Register (Registry&)
```

Private Members

```
std::map<std::string, std::string> labels_
```

```
std::string name_
```

```
std::string help_
```

```
struct Item
```

Public Functions

```
Item (double value, int lower_delta, int delta)
```

Public Members

double **value**

int **g**

int **delta**

struct Label

Public Functions

bool **operator<** (const *Label* &*rhs*) const

bool **operator==** (const *Label* &*rhs*) const

Public Members

std::string **name**

std::string **value**

struct MetricFamily

Public Members

std::string **name**

std::string **help**

MetricType **type** = MetricType::Untyped

std::vector<*ClientMetric*> **metric**

struct Quantile

Public Members

double **quantile** = 0.0

double **value** = 0.0

struct Quantile

Public Functions

Quantile (double *quantile*, double *error*)

Public Members

const double **quantile**

const double **error**

const double **u**

const double **v**

```
class Registry : public prometheus::Collectable
```

Public Functions

```
std::vector<MetricFamily> Collect ()
```

Private Functions

```
Family<Counter> &AddCounter (const std::string &name, const std::string &help, const
                           std::map<std::string, std::string> &labels)
```

```
Family<Gauge> &AddGauge (const std::string &name, const std::string &help, const
                        std::map<std::string, std::string> &labels)
```

```
Family<Histogram> &AddHistogram (const std::string &name, const std::string &help, const
                                std::map<std::string, std::string> &labels)
```

```
Family<Summary> &AddSummary (const std::string &name, const std::string &help, const
                             std::map<std::string, std::string> &labels)
```

Private Members

```
std::vector<std::unique_ptr<Collectable>> collectables_
```

```
std::mutex mutex_
```

Friends

```
friend prometheus::Registry::detail::CounterBuilder
```

```
friend prometheus::Registry::detail::GaugeBuilder
```

```
friend prometheus::Registry::detail::HistogramBuilder
```

```
friend prometheus::Registry::detail::SummaryBuilder
```

```
class Serializer
```

```
Subclassed by prometheus::TextSerializer
```

Public Functions

```
virtual ~Serializer ()
```

```
virtual std::string Serialize (const std::vector<MetricFamily>&) const
```

```
virtual void Serialize (std::ostream &out, const std::vector<MetricFamily> &metrics) const
                    = 0
```

```
class Summary
```

Public Types

```
using Quantiles = std::vector<detail::CKMSQuantiles::Quantile>
```

Public Functions

```
Summary (const Quantiles &quantiles, std::chrono::milliseconds max_age_seconds =  
std::chrono::seconds(60), int age_buckets = 5)
```

```
void Observe (double value)
```

```
ClientMetric Collect ()
```

Public Static Attributes

```
const MetricType metric_type = MetricType::Summary
```

Private Members

```
const Quantiles quantiles_
```

```
std::mutex mutex_
```

```
double count_
```

```
double sum_
```

```
detail::TimeWindowQuantiles quantile_values_
```

```
struct Summary
```

Public Members

```
std::uint64_t sample_count = 0
```

```
double sample_sum = 0.0
```

```
std::vector<Quantile> quantile
```

```
class SummaryBuilder
```

Public Functions

```
SummaryBuilder &Labels (const std::map<std::string, std::string> &labels)
```

```
SummaryBuilder &Name (const std::string&)
```

```
SummaryBuilder &Help (const std::string&)
```

```
Family<Summary> &Register (Registry&)
```

Private Members

```
std::map<std::string, std::string> labels_
```

```
std::string name_
```

```
std::string help_
```

```
class TextSerializer : public prometheus::Serializer
```


Public Functions

```
void Serialize (std::ostream &out, const std::vector<MetricFamily> &metrics) const  
class TimeWindowQuantiles
```

Public Functions

```
TimeWindowQuantiles (const          std::vector<CKMSQuantiles::Quantile>      &quantiles,  
                      Clock::duration max_age_seconds, int age_buckets)
```

```
double get (double q)
```

```
void insert (double value)
```

Private Types

```
using Clock = std::chrono::steady_clock
```

Private Functions

```
CKMSQuantiles &rotate ()
```

Private Members

```
const std::vector<CKMSQuantiles::Quantile> &quantiles_
```

```
std::vector<CKMSQuantiles> ckms_quantiles_
```

```
std::size_t current_bucket_
```

```
Clock::time_point last_rotation_
```

```
const Clock::duration rotation_interval_
```

```
struct Untyped
```

Public Members

```
double value = 0
```

```
namespace prometheus
```

Enums

```
enum MetricType
```

```
    Values:
```

```
    Counter
```

```
    Gauge
```

```
    Summary
```

```
    Untyped
```

Histogram

Functions

`bool CheckMetricName (const std::string &name)`

`bool CheckLabelName (const std::string &name)`

`detail::CounterBuilder BuildCounter ()`

`detail::GaugeBuilder BuildGauge ()`

`detail::HistogramBuilder BuildHistogram ()`

`detail::SummaryBuilder BuildSummary ()`

`namespace detail`

`file check_names.h`

`#include <string>`

`file client_metric.h`

`#include <cstdint>#include <string>#include <tuple>#include <vector>#include
"prometheus/metric_type.h"`

`file collectable.h`

`#include <vector>`

`file counter.h`

`#include <atomic>#include "prometheus/client_metric.h"#include "prometheus/gauge.h"`

`file counter_builder.h`

`#include <map>#include <string>`

`file family.h`

`#include <algorithm>#include <cassert>#include <functional>#include <map>#include <memory>#include <mutex>#include <numeric>#include <string>#include <unordered_map>#include
"check_names.h"#include "collectable.h"#include "metric_family.h"`

`file gauge.h`

`#include <atomic>#include "prometheus/client_metric.h"#include "prometheus/collectable.h"`

`file gauge_builder.h`

`#include <map>#include <string>`

`file histogram.h`

`#include <vector>#include "prometheus/client_metric.h"#include "prometheus/counter.h"`

`file histogram_builder.h`

`#include <map>#include <string>#include <vector>`

`file metric_family.h`

`#include <string>#include <vector>#include "prometheus/client_metric.h"#include
"prometheus/metric_type.h"`

`file metric_type.h`

`file registry.h`

`#include <map>#include <memory>#include <mutex>#include "prometheus/client_metric.h"#include
"prometheus/collectable.h"#include "prometheus/counter_builder.h"#include
"prometheus/family.h"#include "prometheus/gauge_builder.h"#include "prometheus/histogram.h"#include`

```
"prometheus/histogram_builder.h"#include
"prometheus/summary_builder.h"
```

```
"prometheus/summary.h"#include
```

```
file serializer.h
```

```
#include <string>#include <vector>#include "prometheus/client_metric.h"#include
"prometheus/metric_family.h"
```

```
file summary.h
```

```
#include <array>#include <atomic>#include <chrono>#include <functional>#include <list>#include <mutex>#include <vector>#include "prometheus/client_metric.h"
```

```
file summary_builder.h
```

```
#include <map>#include <string>#include <vector>
```

```
file text_serializer.h
```

```
#include <string>#include <vector>#include "prometheus/metric_family.h"#include
"prometheus/serializer.h"
```

```
file exposer.h
```

```
#include <atomic>#include <stdint>#include <memory>#include <string>#include
"prometheus/histogram.h"#include "prometheus/registry.h"
```

```
file gateway.h
```

```
#include <future>#include <map>#include <memory>#include <sstream>#include
"prometheus/histogram.h"#include "prometheus/registry.h"
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```

```
dir /home/docs/checkouts/readthedocs.org/user_builds/prometheus-cpp/checkouts/read-the-docs/
```


P

- prometheus (C++ type), 13
- prometheus::BuildCounter (C++ function), 14
- prometheus::BuildGauge (C++ function), 14
- prometheus::BuildHistogram (C++ function), 14
- prometheus::BuildSummary (C++ function), 14
- prometheus::CheckLabelName (C++ function), 14
- prometheus::CheckMetricName (C++ function), 14
- prometheus::ClientMetric (C++ class), 4
- prometheus::ClientMetric::Bucket (C++ class), 3
- prometheus::ClientMetric::Bucket::cumulative_count (C++ member), 3
- prometheus::ClientMetric::Bucket::upper_bound (C++ member), 3
- prometheus::ClientMetric::Counter (C++ class), 5
- prometheus::ClientMetric::counter (C++ member), 4
- prometheus::ClientMetric::Counter::value (C++ member), 5
- prometheus::ClientMetric::Gauge (C++ class), 7
- prometheus::ClientMetric::gauge (C++ member), 4
- prometheus::ClientMetric::Gauge::value (C++ member), 7
- prometheus::ClientMetric::Histogram (C++ class), 8
- prometheus::ClientMetric::histogram (C++ member), 4
- prometheus::ClientMetric::Histogram::bucket (C++ member), 8
- prometheus::ClientMetric::Histogram::sample_count (C++ member), 8
- prometheus::ClientMetric::Histogram::sample_sum (C++ member), 8
- prometheus::ClientMetric::Label (C++ class), 10
- prometheus::ClientMetric::label (C++ member), 4
- prometheus::ClientMetric::Label::name (C++ member), 10
- prometheus::ClientMetric::Label::operator== (C++ function), 10
- prometheus::ClientMetric::Label::operator< (C++ function), 10
- prometheus::ClientMetric::Label::value (C++ member), 10
- prometheus::ClientMetric::Quantile (C++ class), 10
- prometheus::ClientMetric::Quantile::quantile (C++ member), 10
- prometheus::ClientMetric::Quantile::value (C++ member), 10
- prometheus::ClientMetric::Summary (C++ class), 12
- prometheus::ClientMetric::summary (C++ member), 4
- prometheus::ClientMetric::Summary::quantile (C++ member), 12
- prometheus::ClientMetric::Summary::sample_count (C++ member), 12
- prometheus::ClientMetric::Summary::sample_sum (C++ member), 12
- prometheus::ClientMetric::timestamp_ms (C++ member), 4
- prometheus::ClientMetric::Untyped (C++ class), 13
- prometheus::ClientMetric::untyped (C++ member), 4
- prometheus::ClientMetric::Untyped::value (C++ member), 13
- prometheus::Collectable (C++ class), 4
- prometheus::Collectable::~Collectable (C++ function), 4
- prometheus::Collectable::Collect (C++ function), 4
- prometheus::Counter (C++ class), 4
- prometheus::Counter (C++ enumerator), 13
- prometheus::Counter::Collect (C++ function), 4
- prometheus::Counter::gauge_ (C++ member), 5
- prometheus::Counter::Increment (C++ function), 4
- prometheus::Counter::metric_type (C++ member), 4
- prometheus::Counter::Value (C++ function), 4
- prometheus::detail (C++ type), 14
- prometheus::detail::CKMSQuantiles (C++ class), 3
- prometheus::detail::CKMSQuantiles::allowableError (C++ function), 3
- prometheus::detail::CKMSQuantiles::buffer_ (C++ member), 4
- prometheus::detail::CKMSQuantiles::buffer_count_ (C++ member), 4
- prometheus::detail::CKMSQuantiles::CKMSQuantiles (C++ function), 3

prometheus::detail::CKMSQuantiles::compress (C++ function), 3	8
prometheus::detail::CKMSQuantiles::count_ (C++ member), 4	prometheus::detail::GaugeBuilder::help_ (C++ member), 8
prometheus::detail::CKMSQuantiles::get (C++ function), 3	prometheus::detail::GaugeBuilder::Labels (C++ function), 8
prometheus::detail::CKMSQuantiles::insert (C++ function), 3	prometheus::detail::GaugeBuilder::labels_ (C++ member), 8
prometheus::detail::CKMSQuantiles::insertBatch (C++ function), 3	prometheus::detail::GaugeBuilder::Name (C++ function), 8
prometheus::detail::CKMSQuantiles::Item (C++ class), 9	prometheus::detail::GaugeBuilder::name_ (C++ member), 8
prometheus::detail::CKMSQuantiles::Item::delta (C++ member), 10	prometheus::detail::GaugeBuilder::Register (C++ function), 8
prometheus::detail::CKMSQuantiles::Item::g (C++ member), 10	prometheus::detail::HistogramBuilder (C++ class), 9
prometheus::detail::CKMSQuantiles::Item::Item (C++ function), 9	prometheus::detail::HistogramBuilder::Help (C++ function), 9
prometheus::detail::CKMSQuantiles::Item::value (C++ member), 10	prometheus::detail::HistogramBuilder::help_ (C++ member), 9
prometheus::detail::CKMSQuantiles::Quantile (C++ class), 10	prometheus::detail::HistogramBuilder::Labels (C++ function), 9
prometheus::detail::CKMSQuantiles::Quantile::error (C++ member), 10	prometheus::detail::HistogramBuilder::labels_ (C++ member), 9
prometheus::detail::CKMSQuantiles::Quantile::Quantile (C++ function), 10	prometheus::detail::HistogramBuilder::Name (C++ function), 9
prometheus::detail::CKMSQuantiles::Quantile::quantile (C++ member), 10	prometheus::detail::HistogramBuilder::name_ (C++ member), 9
prometheus::detail::CKMSQuantiles::Quantile::u (C++ member), 10	prometheus::detail::HistogramBuilder::Register (C++ function), 9
prometheus::detail::CKMSQuantiles::Quantile::v (C++ member), 10	prometheus::detail::SummaryBuilder (C++ class), 12
prometheus::detail::CKMSQuantiles::quantiles_ (C++ member), 4	prometheus::detail::SummaryBuilder::Help (C++ function), 12
prometheus::detail::CKMSQuantiles::reset (C++ function), 3	prometheus::detail::SummaryBuilder::help_ (C++ member), 12
prometheus::detail::CKMSQuantiles::sample_ (C++ member), 4	prometheus::detail::SummaryBuilder::Labels (C++ function), 12
prometheus::detail::CounterBuilder (C++ class), 5	prometheus::detail::SummaryBuilder::labels_ (C++ member), 12
prometheus::detail::CounterBuilder::Help (C++ function), 5	prometheus::detail::SummaryBuilder::Name (C++ function), 12
prometheus::detail::CounterBuilder::help_ (C++ member), 5	prometheus::detail::SummaryBuilder::name_ (C++ member), 12
prometheus::detail::CounterBuilder::Labels (C++ function), 5	prometheus::detail::SummaryBuilder::Register (C++ function), 12
prometheus::detail::CounterBuilder::labels_ (C++ member), 5	prometheus::detail::TimeWindowQuantiles (C++ class), 13
prometheus::detail::CounterBuilder::Name (C++ function), 5	prometheus::detail::TimeWindowQuantiles::ckms_quantiles_ (C++ member), 13
prometheus::detail::CounterBuilder::name_ (C++ member), 5	prometheus::detail::TimeWindowQuantiles::Clock (C++ type), 13
prometheus::detail::CounterBuilder::Register (C++ function), 5	prometheus::detail::TimeWindowQuantiles::current_bucket_ (C++ member), 13
prometheus::detail::GaugeBuilder (C++ class), 8	prometheus::detail::TimeWindowQuantiles::get (C++ function), 13
prometheus::detail::GaugeBuilder::Help (C++ function),	prometheus::detail::TimeWindowQuantiles::insert (C++

function), 13

prometheus::detail::TimeWindowQuantiles::last_rotation_ (C++ member), 13

prometheus::detail::TimeWindowQuantiles::quantiles_ (C++ member), 13

prometheus::detail::TimeWindowQuantiles::rotate (C++ function), 13

prometheus::detail::TimeWindowQuantiles::rotation_interval_ (C++ member), 13

prometheus::detail::TimeWindowQuantiles::TimeWindowQuantiles (C++ function), 13

prometheus::Exposer (C++ class), 5

prometheus::Exposer::~Exposer (C++ function), 5

prometheus::Exposer::collectables_ (C++ member), 5

prometheus::Exposer::Exposer (C++ function), 5

prometheus::Exposer::exposer_registry_ (C++ member), 5

prometheus::Exposer::metrics_handler_ (C++ member), 5

prometheus::Exposer::RegisterCollectable (C++ function), 5

prometheus::Exposer::server_ (C++ member), 5

prometheus::Exposer::uri_ (C++ member), 5

prometheus::Family (C++ class), 5

prometheus::Family::Add (C++ function), 6

prometheus::Family::Collect (C++ function), 6

prometheus::Family::CollectMetric (C++ function), 6

prometheus::Family::constant_labels_ (C++ member), 6

prometheus::Family::Family (C++ function), 6

prometheus::Family::hash_labels (C++ function), 6

prometheus::Family::help_ (C++ member), 6

prometheus::Family::labels_ (C++ member), 6

prometheus::Family::labels_reverse_lookup_ (C++ member), 6

prometheus::Family::metrics_ (C++ member), 6

prometheus::Family::mutex_ (C++ member), 6

prometheus::Family::name_ (C++ member), 6

prometheus::Family::Remove (C++ function), 6

prometheus::Gateway (C++ class), 6

prometheus::Gateway::Add (C++ enumerator), 7

prometheus::Gateway::async_push (C++ function), 7

prometheus::Gateway::AsyncDelete (C++ function), 7

prometheus::Gateway::AsyncPush (C++ function), 6

prometheus::Gateway::AsyncPushAdd (C++ function), 6

prometheus::Gateway::collectables_ (C++ member), 7

prometheus::Gateway::Delete (C++ function), 7

prometheus::Gateway::Gateway (C++ function), 6

prometheus::Gateway::GetInstanceLabel (C++ function), 7

prometheus::Gateway::jobUri_ (C++ member), 7

prometheus::Gateway::Labels (C++ type), 6

prometheus::Gateway::labels_ (C++ member), 7

prometheus::Gateway::password_ (C++ member), 7

prometheus::Gateway::Push (C++ function), 6

prometheus::Gateway::push (C++ function), 7

prometheus::Gateway::PushAdd (C++ function), 6

prometheus::Gateway::PushMode (C++ type), 7

prometheus::Gateway::RegisterCollectable (C++ function), 6

prometheus::Gateway::Replace (C++ enumerator), 7

prometheus::Gateway::username_ (C++ member), 7

prometheus::Gauge (C++ class), 7

prometheus::Gauge (C++ enumerator), 13

prometheus::Gauge::Change (C++ function), 8

prometheus::Gauge::Collect (C++ function), 8

prometheus::Gauge::Decrement (C++ function), 7, 8

prometheus::Gauge::Gauge (C++ function), 7

prometheus::Gauge::Increment (C++ function), 7

prometheus::Gauge::metric_type (C++ member), 8

prometheus::Gauge::Set (C++ function), 8

prometheus::Gauge::SetToCurrentTime (C++ function), 8

prometheus::Gauge::Value (C++ function), 8

prometheus::Gauge::value_ (C++ member), 8

prometheus::Histogram (C++ class), 8

prometheus::Histogram (C++ enumerator), 13

prometheus::Histogram::bucket_boundaries_ (C++ member), 9

prometheus::Histogram::bucket_counts_ (C++ member), 9

prometheus::Histogram::BucketBoundaries (C++ type), 9

prometheus::Histogram::Collect (C++ function), 9

prometheus::Histogram::Histogram (C++ function), 9

prometheus::Histogram::metric_type (C++ member), 9

prometheus::Histogram::Observe (C++ function), 9

prometheus::Histogram::sum_ (C++ member), 9

prometheus::MetricFamily (C++ class), 10

prometheus::MetricFamily::help (C++ member), 10

prometheus::MetricFamily::metric (C++ member), 10

prometheus::MetricFamily::name (C++ member), 10

prometheus::MetricFamily::type (C++ member), 10

prometheus::MetricType (C++ type), 13

prometheus::Registry (C++ class), 10

prometheus::Registry::AddCounter (C++ function), 11

prometheus::Registry::AddGauge (C++ function), 11

prometheus::Registry::AddHistogram (C++ function), 11

prometheus::Registry::AddSummary (C++ function), 11

prometheus::Registry::Collect (C++ function), 11

prometheus::Registry::collectables_ (C++ member), 11

prometheus::Registry::mutex_ (C++ member), 11

prometheus::Serializer (C++ class), 11

prometheus::Serializer::~Serializer (C++ function), 11

prometheus::Serializer::Serialize (C++ function), 11

prometheus::Summary (C++ class), 11

prometheus::Summary (C++ enumerator), 13

prometheus::Summary::Collect (C++ function), 12

prometheus::Summary::count_ (C++ member), 12

prometheus::Summary::metric_type (C++ member), 12

prometheus::Summary::mutex_ (C++ member), 12

[prometheus::Summary::Observe \(C++ function\), 12](#)
[prometheus::Summary::quantile_values_ \(C++ member\), 12](#)
[prometheus::Summary::Quantiles \(C++ type\), 11](#)
[prometheus::Summary::quantiles_ \(C++ member\), 12](#)
[prometheus::Summary::sum_ \(C++ member\), 12](#)
[prometheus::Summary::Summary \(C++ function\), 12](#)
[prometheus::TextSerializer \(C++ class\), 12](#)
[prometheus::TextSerializer::Serialize \(C++ function\), 13](#)
[prometheus::Untyped \(C++ enumerator\), 13](#)