

[Hummingbird](#) / TracingMiddleware

Structure

TracingMiddleware

Middleware creating Distributed Tracing spans for each request.

```
struct TracingMiddleware<Context> where Context : RequestContext
```

Overview

Creates a span for each request, including attributes such as the HTTP method.

You may opt in to recording a specific subset of HTTP request/response header values by passing a set of header names.

Uses [Swift-Distributed-Tracing](#) for recording the traces. Swift-Distributed-Tracing has a flexible backend, which will need to be initialized before any traces are recorded.

A list of implementations is available in the [swift-distributed-tracing repository's README](#).

Topics

Initializers

```
init(recordingHeaders: some Collection<HTTPField.Name>, attributes: SpanAttributes?)
```

Initialize a new TracingMiddleware.

Instance Methods

```
func handle(Request, context: Context, next: (Request, Context)  
async throws -> Response) async throws -> Response
```

Relationships

Conforms To

MiddlewareProtocol, RouterMiddleware, Swift.Sendable

See Also

Middleware

`protocol MiddlewareProtocol`

Middleware protocol with generic input, context and output types

`enum MiddlewareFixedTypeBuilder`

Middleware stack result builder

`protocol RouterMiddleware`

Version of MiddlewareProtocol whose Input is Request and output is Response.

`class MiddlewareGroup`

Group of middleware that can be used to create a responder chain. Each middleware calls the next one

`struct CORSMiddleware`

Middleware implementing Cross-Origin Resource Sharing (CORS) headers.

`struct LogRequestsMiddleware`

Middleware outputting to log for every call to server.

`struct MetricsMiddleware`

Middleware recording metrics for each request