

[Hummingbird](#) / MiddlewareProtocol

Protocol

MiddlewareProtocol

Middleware protocol with generic input, context and output types

```
protocol MiddlewareProtocol<Input, Output, Context> : Sendable
```

Topics

Associated Types

associatedtype Context

Required

associatedtype Input

Required

associatedtype Output

Required

Instance Methods

```
func handle(Input, context: Context, next: (Input, Context) async throws -> Output) async throws -> Output
```

Required Default implementation provided.

Relationships

Inherits From

`Swift.Sendable`

Inherited By

`AuthenticatorMiddleware`, `RouterMiddleware`

Conforming Types

`BasicAuthenticator`
`CORSMiddleware`
`ClosureAuthenticator`
`ContextTransform`
`FileMiddleware`
`Handle`
`IsAuthenticatedMiddleware`
`LogRequestsMiddleware`
`MetricsMiddleware`
`RequestDecompressionMiddleware`
`ResponseCompressionMiddleware`
`Route`
`RouteGroup`
`RouterBuilder`
`SessionAuthenticator`
`SessionMiddleware`
`ThrowingContextTransform`
`TracingMiddleware`
`WebSocketUpgradeMiddleware`

See Also

Middleware

`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

`struct TracingMiddleware`

Middleware creating Distributed Tracing spans for each request.
