

[Hummingbird](#) / LogRequestsMiddleware

## Structure

# LogRequestsMiddleware

Middleware outputting to log for every call to server.

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

## Overview

Uses [Swift-Log](#) for logging the output. Swift-Log has a flexible backend, and will output to console by default. You can replace the Logging backend with other implementations. A list of implementations is available in the [swift-log repository's README](#).

## Topics

### Structures

```
struct HeaderFilter
```

Header filter

### Initializers

```
init(Logger.Level, includeHeaders: HeaderFilter, redactHeaders: [HTTPField.Name])
```

## 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 MetricsMiddleware`

Middleware recording metrics for each request

## `struct TracingMiddleware`

Middleware creating Distributed Tracing spans for each request.

---