Documentation Language: Swift

Hummingbird / CORSMiddleware

Structure

CORSMiddleware

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

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

Overview

If request has “origin” header then generate CORS headers. If method is OPTIONS then return an empty body with all the standard CORS headers otherwise send request onto the next handler and when you receive the response add a “access-control-allow-origin” header

Topics

Initializers

`init(allowOrigin: AllowOrigin, allowHeaders: [HTTPField.Name], allowMethods: [HTTPRequest.Method], allowCredentials: Bool, exposedHeaders: [String]?, maxAge: TimeAmount?)`

Initialize CORS middleware

Instance Methods

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

apply CORS middleware

Enumerations

`enum AllowOrigin`

Defines what origins are allowed

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 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.