Documentation Language: Swift

Hummingbird / RouterGroup

Structure

RouterGroup

Used to group together routes under a single path. Additional middleware can be added to the endpoint and each route can add a suffix to the endpoint path

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

Overview

The code below creates an `RouterGroup`with path “todos” and adds GET and PUT routes on “todos” and adds GET, PUT and DELETE routes on “todos/:id” where id is the identifier for the todo

```
app.router
.group("todos")
.get(use: todoController.list)
.put(use: todoController.create)
.get(":id", use: todoController.get)
.put(":id", use: todoController.update)
.delete(":id", use: todoController.delete)
```

Topics

Instance Methods

```
func add(middleware: any MiddlewareProtocol<Request, Response, Context>) -> RouterGroup<Context>
```

Add middleware to RouterGroup

```
func on<Responder>(RouterPath, method: HTTPRequest.Method, responder: Responder) -> RouterGroup<Context>
```

Add responder to call when path and method are matched

Default Implementations

- RouterMethods Implementations

Relationships

Conforms To

`RouterMethods`

See Also

Router

```
class Router
```

Create rules for routing requests and then create `Responder` that will follow these rules.

```
class RouteCollection
```

Collection of routes

`protocol` RouterMethods

Conform to `RouterMethods` to add standard router verb (get, post …) methods

`struct` RouterOptions

Router Options

`protocol` HTTPResponder

Protocol for object that produces a response given a request

`protocol` HTTPResponderBuilder

A type that has a single method to build a HTTPResponder

`struct` CallbackResponder

Responder that calls supplied closure

`struct` RouterResponder

`struct` EndpointPath

Endpoint path storage

`struct` RouterPath

Split router path into components

`struct` RequestID

Generate Unique ID for each request