Documentation Language: Swift

Hummingbird / RouterMethods

Protocol

RouterMethods

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

```
protocol RouterMethods<Context>
```

Topics

Associated Types

`associatedtype Context : RequestContext`

Required

Instance Methods

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

add middleware

Required

`func addMiddleware(buildMiddlewareStack: () -> some MiddlewareProtocol<Request, Response, Context>) -> Self`

Add middleware stack to router

`func addRoutes(RouteCollection<Context>, atPath: RouterPath) -> Self`

Add route collection to router

```
func delete(RouterPath, use: (Request, Context) async throws -> some
 ResponseGenerator) -> Self
```

DELETE path for async closure returning type conforming to ResponseGenerator

```
func get(RouterPath, use: (Request, Context) async throws -> some
ResponseGenerator) -> Self
```

GET path for async closure returning type conforming to ResponseGenerator

```
func group(RouterPath) -> RouterGroup<Context>
```

Return a group inside the current group

```
func group<TargetContext>(RouterPath, context: TargetContext.Type) -
> RouterGroup<TargetContext>
```

Return a group inside the current group that transforms the RequestContext

```
func group<TargetContext>(RouterPath, context: TargetContext.Type) -
> RouterGroup<TargetContext>
```

Return a group inside the current group that transforms the RequestContext

```
func head(RouterPath, use: (Request, Context) async throws -> some
ResponseGenerator) -> Self
```

HEAD path for async closure returning type conforming to ResponseGenerator

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

Add responder to call when path and method are matched

Required

```
func on(RouterPath, method: HTTPRequest.Method, use: (Request,
Context) async throws -> some ResponseGenerator) -> Self
```

Add path for async closure

```
func patch(RouterPath, use: (Request, Context) async throws -> some
ResponseGenerator) -> Self
```

PATCH path for async closure returning type conforming to ResponseGenerator

```
func post(RouterPath, use: (Request, Context) async throws -> some ResponseGenerator) -> Self
```

POST path for async closure returning type conforming to ResponseGenerator

```
func put(RouterPath, use: (Request, Context) async throws -> some ResponseGenerator) -> Self
```

PUT path for async closure returning type conforming to ResponseGenerator

```
func ws(RouterPath, shouldUpgrade: (Request, Context) async throws -> RouterShouldUpgrade, onUpgrade: WebSocketDataHandler<WebSocket RouterContext<Context>>) -> Self
```

Add path to router that support WebSocket upgrade

Relationships

Conforming Types

`RouteCollection`, `Router`, `RouterGroup`

See Also

Router

```
class Router
```

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

```
struct 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

```
class RouteCollection
```

Collection of routes

`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