

[Hummingbird](#) / RouteCollection

## Class

# RouteCollection

Collection of routes

```
final class RouteCollection<Context> where Context : RequestContext
```

## Topics

### Initializers

```
init(context: Context.Type)
```

Initialize RouteCollection

### Instance Methods

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

Add middleware to RouteCollection

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

Add responder to call when path and method are matched

### Default Implementations

# Relationships

## Conforms To

RouterMethods

---

## 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

`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

---