

[Hummingbird](#) / RouterPath

Structure

RouterPath

Split router path into components

```
struct RouterPath
```

Topics

Structures

```
struct Element
```

Initializers

```
init(String)
```

Initialize RouterPath from URI string

```
init(stringLiteral: String)
```

Initialize RouterPath from String literal

Instance Properties

```
let components: [Element]
```

Array of RouterPath elements

```
let description: String
```

A textual representation of the RouterPath

Instance Methods

```
func appendingPath(RouterPath) -> RouterPath
```

Combine two RouterPaths

```
func lowercased() -> RouterPath
```

Return lowercased version of RouterPath

Default Implementations

- ☰ Collection Implementations
 - ☰ ExpressibleByExtendedGraphemeClusterLiteral Implementations
 - ☰ ExpressibleByStringInterpolation Implementations
 - ☰ ExpressibleByStringLiteral Implementations
 - ☰ Sequence Implementations
-

Relationships

Conforms To

Swift.Collection

Swift.Copyable

Swift.CustomStringConvertible

Swift.ExpressibleByExtendedGraphemeClusterLiteral

Swift.ExpressibleByStringInterpolation

Swift.ExpressibleByStringLiteral

Swift.ExpressibleByUnicodeScalarLiteral

Swift.Sendable

Swift.Sequence

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

`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 RequestID`

Generate Unique ID for each request
