Documentation

API Collection

RouterMethods Implementations

Topics

Instance Methods

```
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(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

---