**Documentation**

Language: Swift

Hummingbird / RequestContext

Protocol

# RequestContext

Protocol that all request contexts should conform to. A RequestContext is a statically typed metadata container for information that is associated with a `Request`, and is therefore instantiated alongside the request.

```
protocol RequestContext : InitializableFromSource, RequestContext
Source where Self.Source : RequestContextSource
```

## Overview

It’s passed along the whole middleware chain through to the route. This allows middleware and the route to share this metadata.

Typical use of a context includes:

- The origin that sent the request (IP address or otherwise)
- The identity, such as a user, that is associated with this request

The context is a statically typed metadata container for the duration of a single request. It’s used to store values between middleware and routes such as the user’s identity.

The lifetime of a RequestContext should not exceed that of the request.

## Topics

## Associated Types

`associatedtype Decoder : RequestDecoder = JSONDecoder`

**Required**

`associatedtype Encoder : ResponseEncoder = JSONEncoder`

**Required**

`associatedtype Source = ApplicationRequestContextSource`

**Required**

## Instance Properties

`var coreContext: CoreRequestContextStorage`

Core context

**Required**

`var endpointPath: String?`

Endpoint path

`var id: String`

Request ID, extracted from Logger

`var maxUploadSize: Int`

Maximum size of request body allowed when decoding requests. If a request body that needs decoding is greater than this size then a Content Too Large (413) response is returned. This only applies to decoding requests.

**Required** Default implementation provided.

`var parameters: Parameters`

Parameters extracted from URI

`var requestDecoder: Decoder`

Request decoder

**Required** Default implementation provided.

`var responseEncoder: Encoder`

Response encoder

**Required** Default implementation provided.

# Relationships

## Inherits From

`InitializableFromSource`, `RequestContextSource`, `Swift.Sendable`

## Inherited By

`AuthRequestContext`
`ChildRequestContext`
`LambdaRequestContext`
`RemoteAddressRequestContext`
`RouterRequestContext`
`SessionRequestContext`

## Conforming Types

`BasicAuthRequestContext`
`BasicLambdaRequestContext`
`BasicRequestContext`
`BasicRouterRequestContext`
`BasicSessionRequestContext`
`BasicWebSocketRequestContext`

# See Also

## Request context

```
protocol RequestContextSource
```

Protocol for source of request contexts

`struct ApplicationRequestContextSource`

RequestContext source for contexts created by `Application`.

`struct BasicRequestContext`

Implementation of a basic request context that supports everything the Hummingbird library needs

`protocol ChildRequestContext`

A RequestContext that can be initialized from another RequestContext.

`struct CoreRequestContextStorage`

Request context values required by Hummingbird itself.

`protocol RemoteAddressRequestContext`

Protocol for request context that stores the remote address of connected client.