**Documentation** Language: Swift

Hummingbird / ApplicationProtocol / test(_:_:)

Instance Method

# test(_:_:)

Test `Application`

```
func test<Value>(
    _ testingSetup: TestingSetup,
    _ test: (any TestClientProtocol) async throws -> Value
) async throws -> Value
```

---

## Parameters

**`testingSetup`**
Indicates which type of testing framework we want

**`test`**
Test function

## Discussion

You use `test` and `execute` to test applications. You can either test using the `.router` test framework which sends requests directly to the router for testing your code or the `.live` or `.ahc` frameworks which both run live servers to pass requests to, but provide a single connection HTTP client or AsyncHTTPClient as a client respectively . The `.router` test framework is quicker and doesn't require setting up a full server but will only test code run from request generation onwards.

The example below is using the `.router` framework to test

```
let router = Router()
router.get("/hello") { _ in
    return "hello"
}
let app = Application(router: router)
app.test(.router) { client in
    // does my app return "hello" in the body for this route
    client.execute(uri: "/hello", method: .GET) { response in
        XCTAssertEqual(String(buffer: response.body), "hello")
    }
}
```

# See Also

## Test Setup

`struct` **TestingSetup**

Type of test framework

`enum` **TestHTTPScheme**

HTTP Scheme to use with AsyncHTTPClient test framework