









```
val routes: HttpHandler = routes(  
  "/echo" bind POST to echo,  
  "/twitter" bind routes(  
    "/tweet" bind POST to tweet  
  )  
)
```

# Concept: Router

**“Matches an `HttpHandler` against a `Request`”**

ie. it's a function!

`Router: (Request) -> HttpHandler?`

Can compose multiple routers to make an `HttpHandler`

```
val routes: HttpHandler = routes(  
  "/echo" bind POST to echo,  
  "/twitter" bind routes(  
    "/tweet" bind POST to tweet  
  )  
)
```

http4k does a depth-first search on the tree, then falls back to 404

# Testing HttpHandlers

This is trivially easy, because http4k messages are immutable data classes!

```
val echo: HttpHandler = { r: Request -> Response(OK).body(r.body) }
```

```
class EchoTest {  
    @Test  
    fun `handler echoes input`() {  
        val input: Request = Request(POST, "/echo").body("hello")  
        val expected: Response = Response(OK).body("hello")
```

```
        assertThat(echo(input), equalTo(expected))
```

```
    }
```

```
}
```