

Lens example 1/3

- http4k provides a Lens implementation which targets our domain: HTTP messages.
- Violations result in an HTTP 400 (Bad Request).

```
val btcPath = Path.int().of("btc")
```

```
val miner: HttpHandler = CatchLensFailure.then(  
  routes(  
    "/mine/{btc}" bind POST to { r: Request ->  
      val newTotal: Int = btcPath.extract(r) + 1  
      val out = "You have $newTotal BTC"  
      val json = """"{ "value": "$out"}"""  
      Response(OK).body(json)  
    }  
  ))
```

Lens example 2/3

- We can also “map ()” the lens into domain types:

```
data class BTC(val value: Int) {  
    operator fun plus(that: BTC) = BTC(value + that.value)  
    fun toString(): value.toString()  
}
```

```
val btcPath = Path.int().map(::BTC).of("btc")
```

```
val miner: HttpHandler = CatchLensFailure.then(  
    routes(  
        "/mine/{btc}" bind POST to { r: Request ->  
            val newTotal: BTC = btcPath.extract(r) + BTC(1)  
            val message = "You have $newTotal BTC"  
            val json = """"{ "value": "$message"}""""  
            Response(OK).body(json)  
        }  
    ))
```