







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

# Typesafe HTTP Contracts

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

- How do we enforce our incoming HTTP contract?
  - Locations: Path/Query/Header/Body/Form
  - Optionality - required or optional?
  - Marshalling + Typesafety
- What about creating outbound messages?

# Concept: Lens