


```
val twitterFilter = Filter { next: HttpHandler ->
  { req: Request ->
    val tweet: Request = req.body(req.bodyString().take(140))
    next(tweet)
  }
}
```

```
val tweet: HttpHandler = twitterFilter.then(handler)
```

Concept: Filter

“Provides pre and post processing on an HTTP operation”

ie. it's a function!

`Filter: (HttpHandler) -> HttpHandler`

```
val twitterFilter = Filter { next: HttpHandler ->
  { req: Request ->
    val tweet: Request = req.body(req.bodyString().take(140))
    next(tweet)
  }
}
```

```
val tweet: HttpHandler = twitterFilter.then(handler)
```

Concept: Router

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

ie. it's a function!

`Router: (Request) -> HttpHandler?`