

Using WebClient

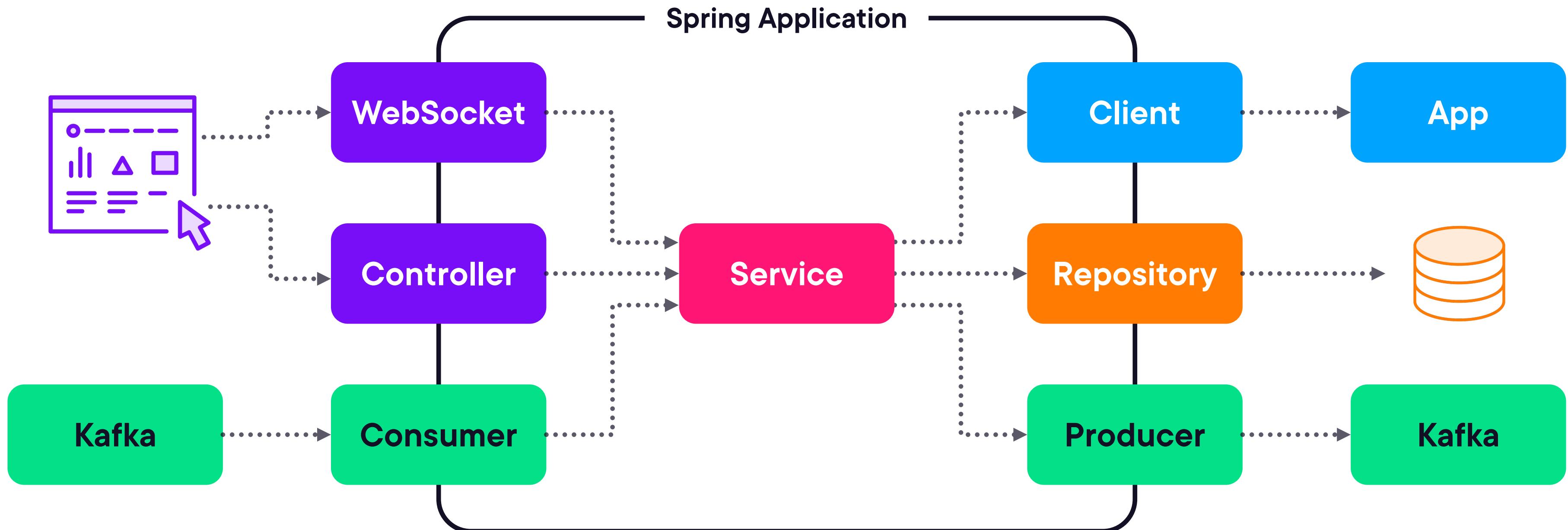


Bogdan Sucaciu

Tech Lead

@bsucaciu | bsucaciu.com

Layered Architecture



WebClient

Reactor Netty

**Jetty Reactive
HttpClient**

**Apache
HttpComponents**



```
WebClient.create()
```

```
WebClient.create(String baseUrl)
```

```
WebClient.builder()  
    .uriBuilderFactory(...)  
    .defaultUriVariables(...)  
    .defaultHeader(...)  
    .defaultCookie(...)  
    .defaultRequest(...)  
    .filter(...)  
    .exchangeStrategies(...)  
    .clientConnector(...)  
    .build()
```

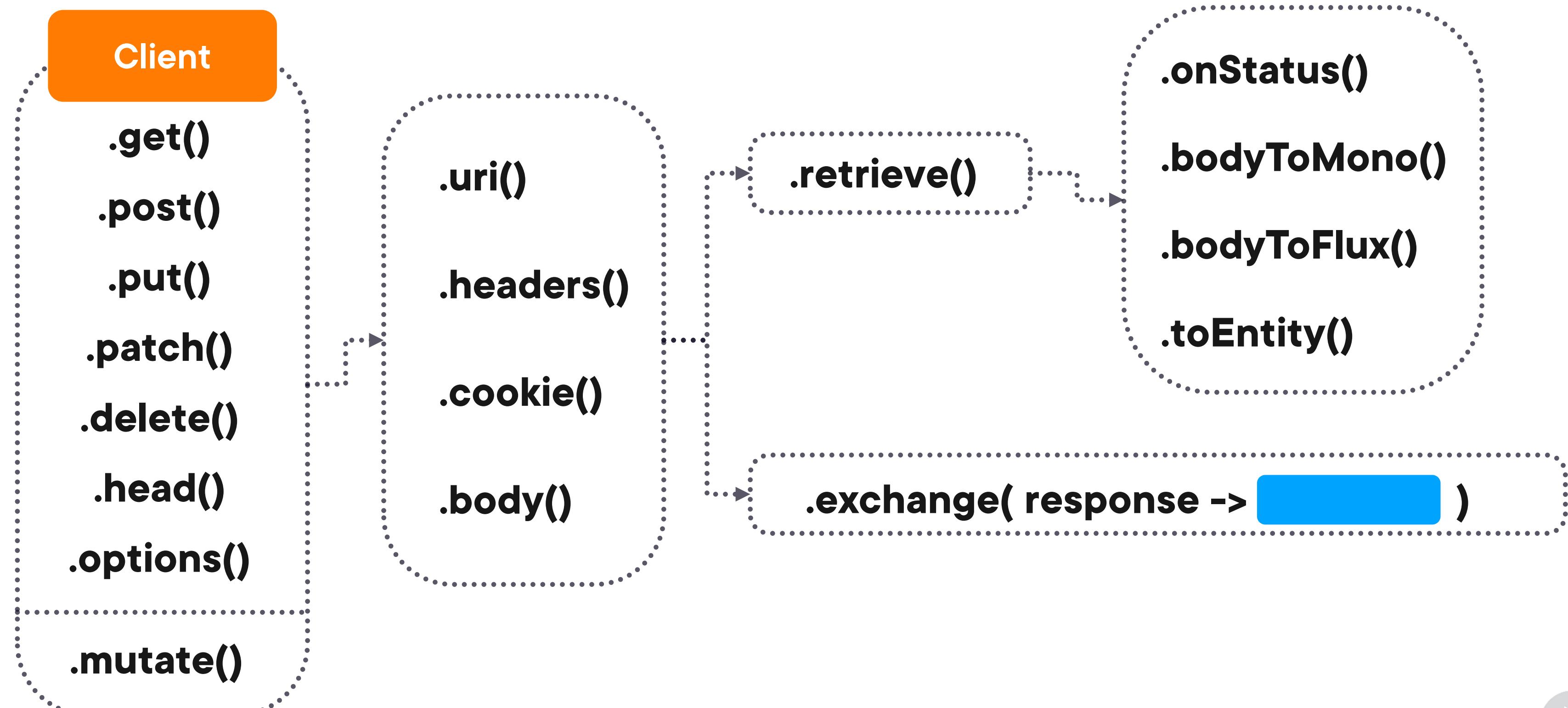
◀ **Create WebClient**

◀ **Create WebClient with base URL**

- ◀ **Builder**
- ◀ **Provide base URL**
- ◀ **Default values to use when expanding URI**
- ◀ **Provide default headers**
- ◀ **Provide default cookies**
- ◀ **Customize request**
- ◀ **Client filter**
- ◀ **HTTP reader/writer customization**
- ◀ **HTTP client library settings**
- ◀ **Finalize build**



WebClient



Demo



Build the Stock Market WebClient

Use WebClient to make GET calls

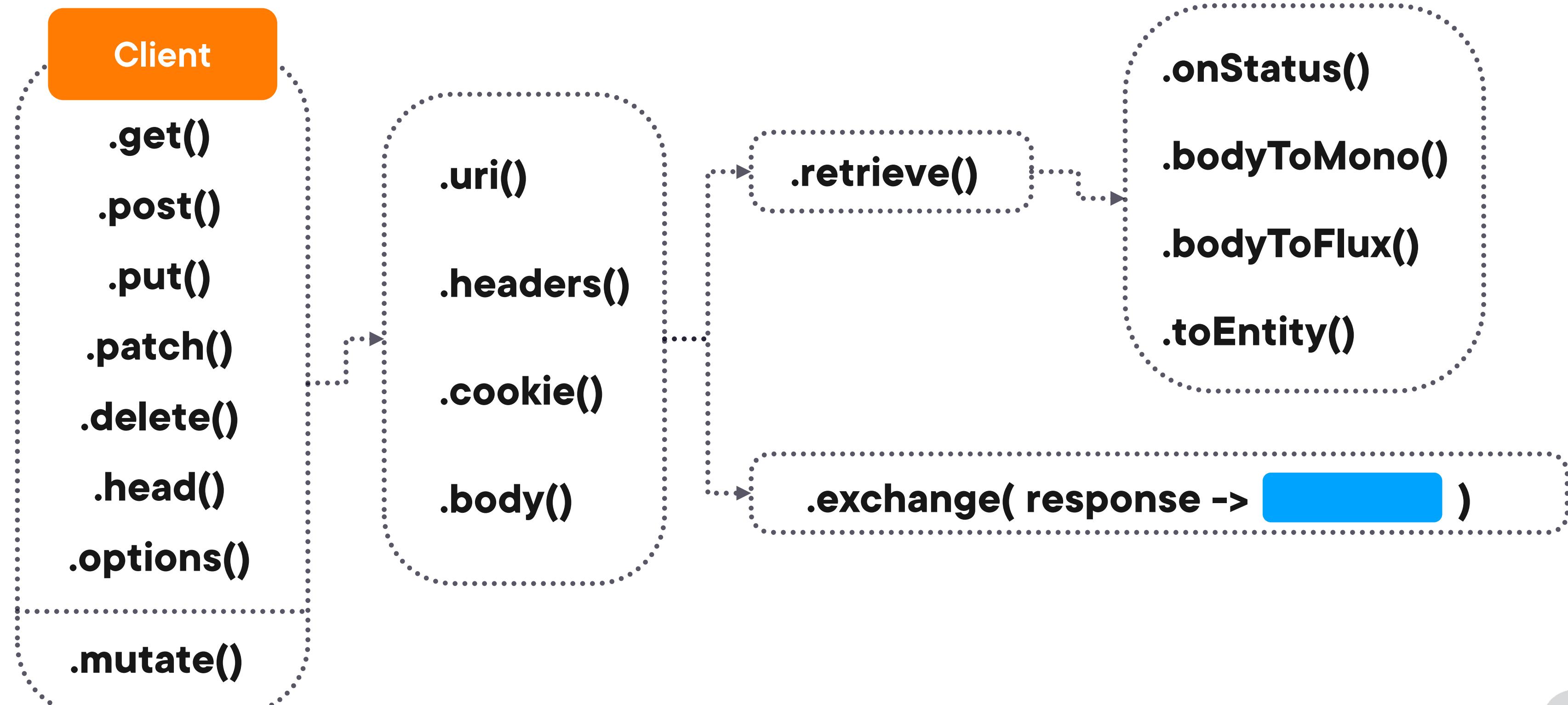
Use WebClient to make POST calls



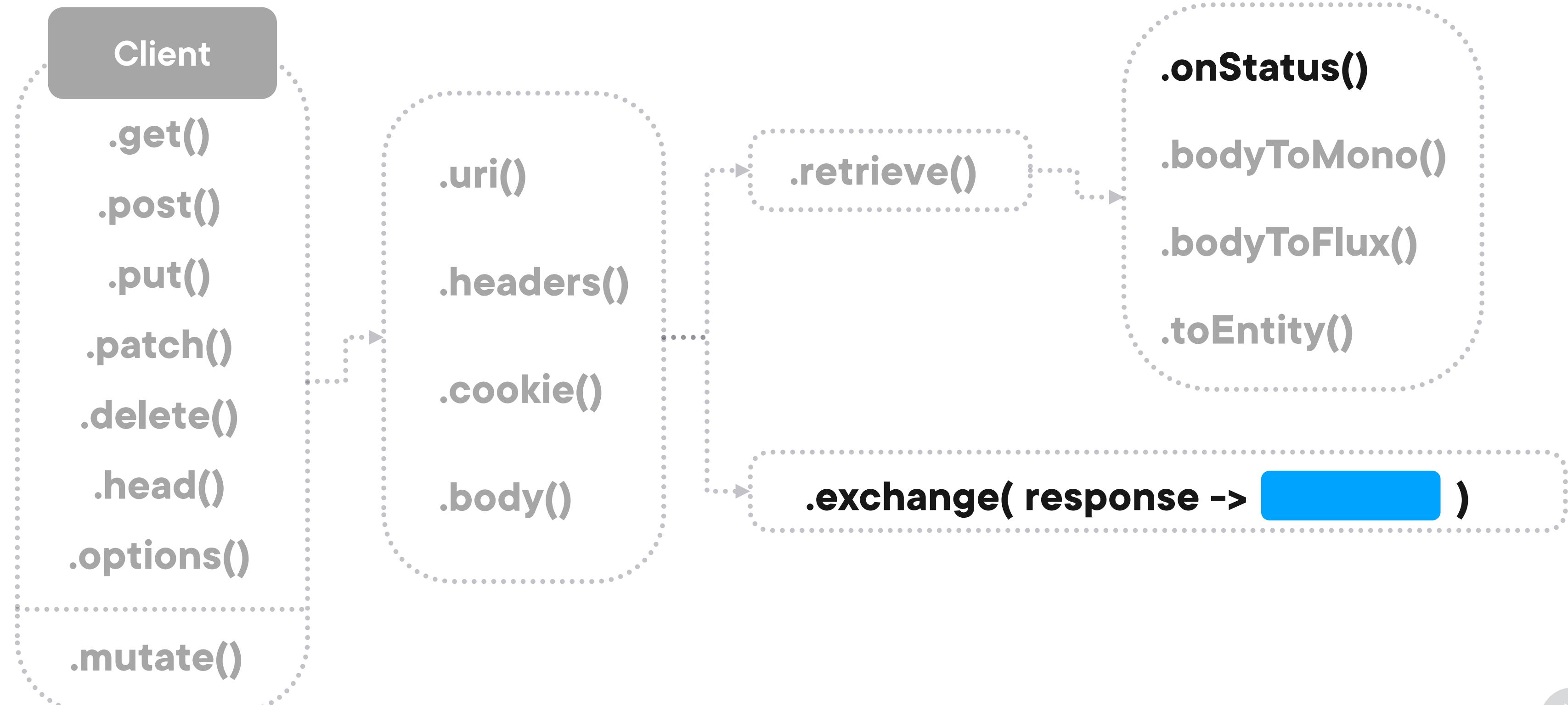
Error Handling



WebClient



WebClient



Error Handling with OnStatus

2xxResponse

```
webClient.post()  
    .retrieve()
```

```
    .bodyToMono(2xxResponse.class)
```

4xxResponse

```
    .onStatus(HttpStatus::is4xxClientError, response -> [ ] )
```

5xxResponse

```
    .onStatus(HttpStatus::is5xxServerError, response -> [ ] )
```



Error Handling with OnStatus

4xxResponse

```
webClient.post()  
    .retrieve()
```

```
    .onStatus(HttpStatus::is4xxClientError, response -> [ ] )
```

5xxResponse

```
    .onStatus(HttpStatus::is5xxServerError, response -> [ ] )
```

2xxResponse

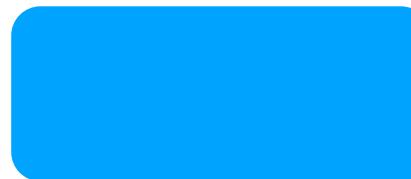
```
    .bodyToMono(2xxResponse.class)
```



Error Handling with OnStatus



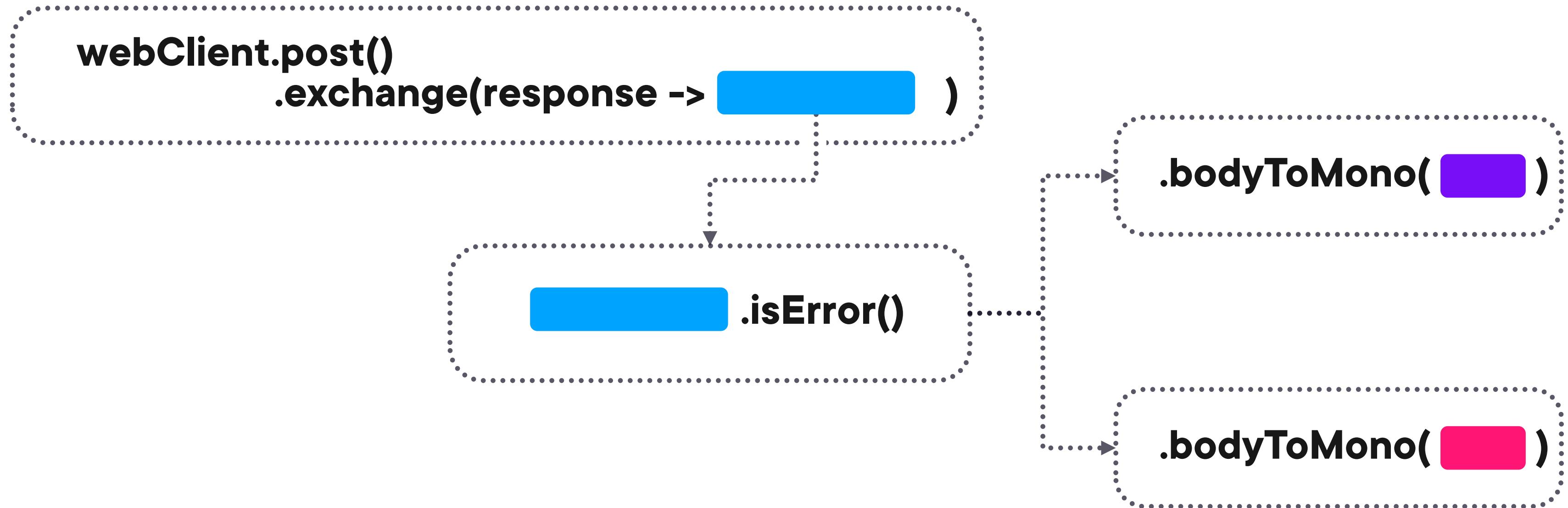
```
response.createException()  
    .flatMap(  
        webClientResponseException -> Mono.error(■))
```



```
response.bodyToMono(ProblemDetail.class)  
    .flatMap(errorResponse -> Mono.error(■))
```



Error Handling with Exchange



Demo



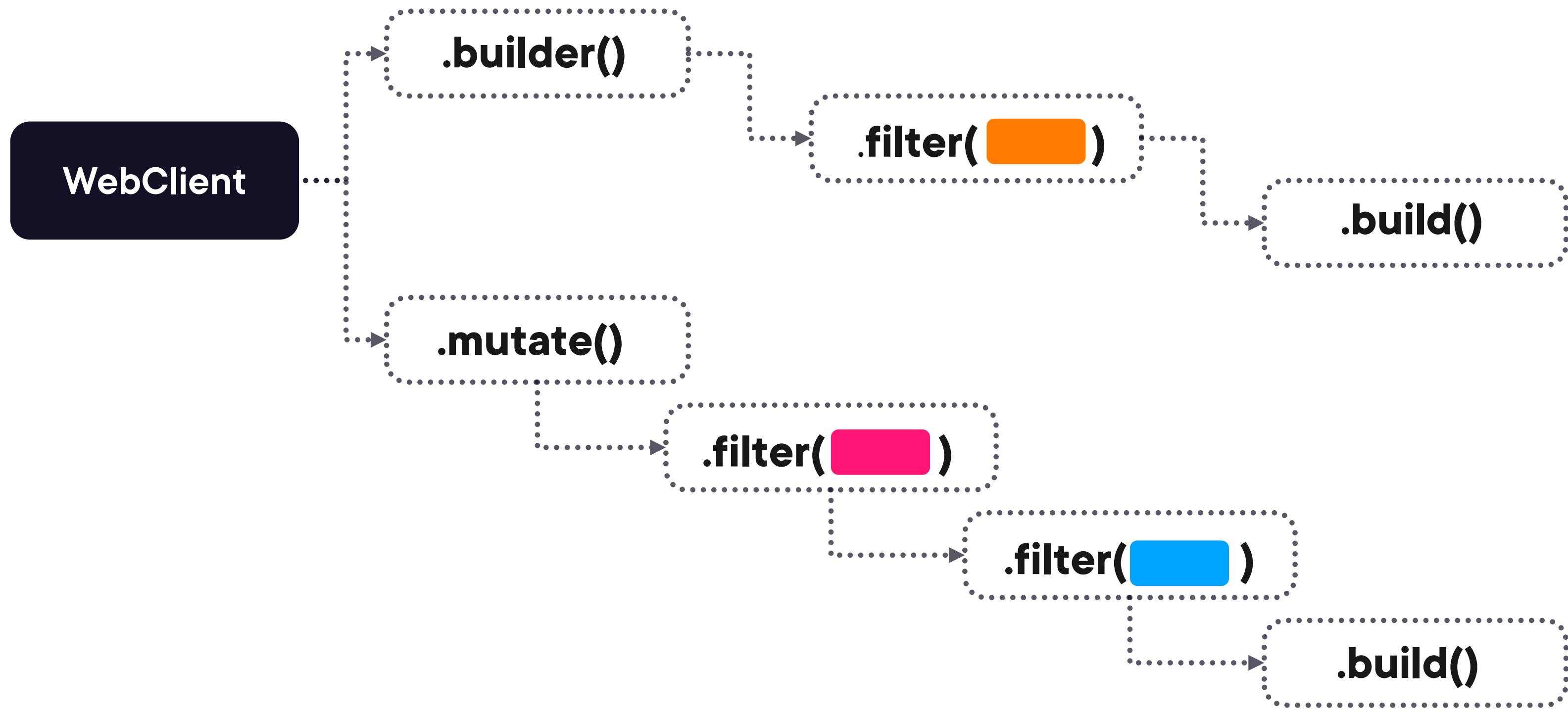
Error handling with exchange



WebClient Filters



WebClient Filters



WebClient Filters

Lambda expression
`(request, next) -> ...`

**Built-in exchange
filter functions**

**Custom exchange
filter functions**



Synchronous Calls



Demo



Create custom filter



Summary



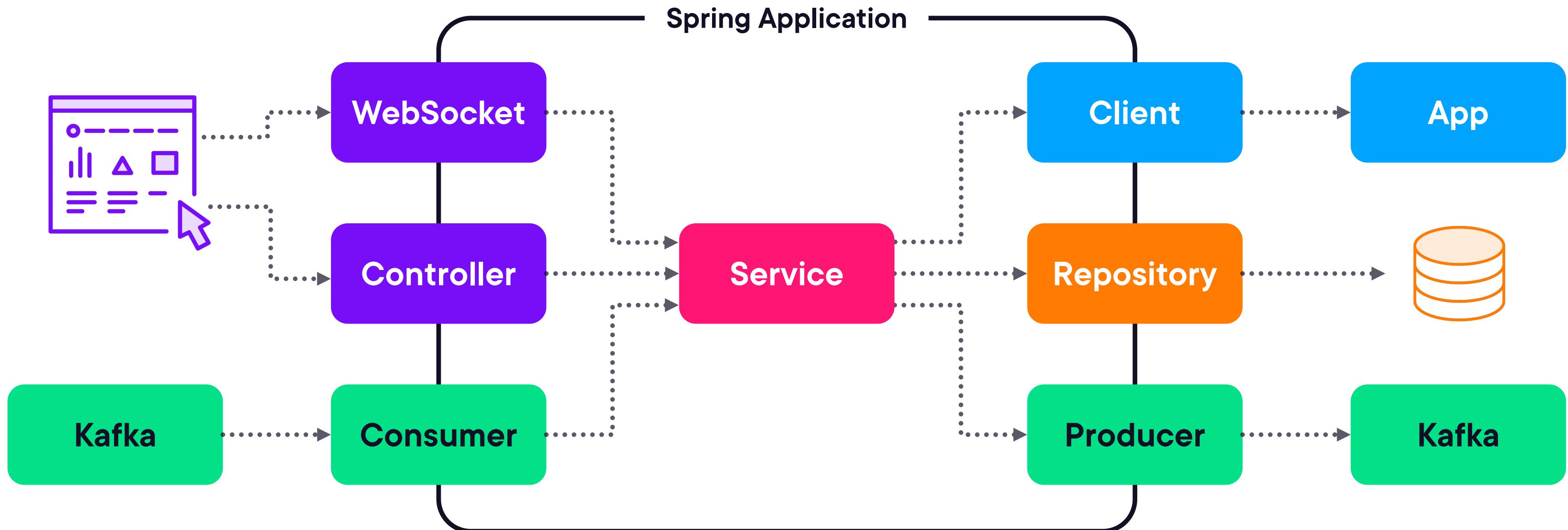
Layered architecture

Error Handling

WebClient Filters



Layered Architecture



Error Handling with OnStatus

4xxResponse

```
webClient.post()  
    .retrieve()
```

```
.onStatus(HttpStatus::is4xxClientError, response ->   )
```

5xxResponse

```
.onStatus(HttpStatus::is5xxServerError, response ->   )
```

2xxResponse

```
.bodyToMono(2xxResponse.class)
```



WebClient Filters

Lambda expression
`(request, next) -> ...`

**Built-in exchange
filter functions**

**Custom exchange
filter functions**



Up Next:

Testing Reactive Streams

