Communicating with the Server Using HTTP, Observables, and Rx

Joe Eames
WEB DEVELOPER

@josepheames www.joeeames.me

Agenda

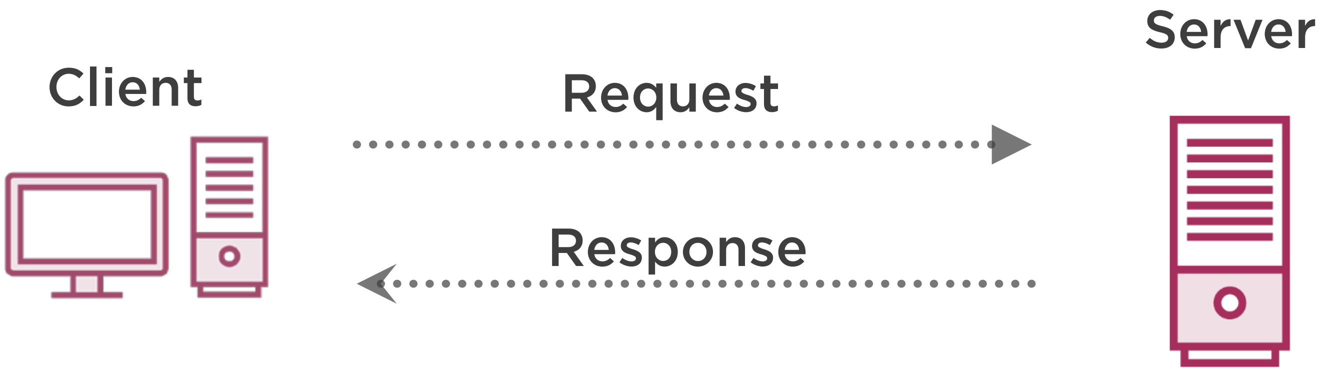
Introduction to RxJS

Moving Data Storage to the Server

- Events Service
- Voter Service

Authentication

HTTP Communication



Callbacks

```
Server.request(requestData, function(responseData) {
   // asynchronously handle the data.
});

// this will execute before the callback
doMoreThings()
```

Promises

```
var promise = http.get(url, data);
promse.then(function(responseData) {
   // handle response
})

// this will execute before the then function
doMoreThings()
return promise;
```

Observables

```
var obs = http.get(url, data);
// manipulate the observable if desired
obs.subscribe(function(responseData) {
   // handle response
});
doMoreThings()
return obs;
```

Promises vs Observables

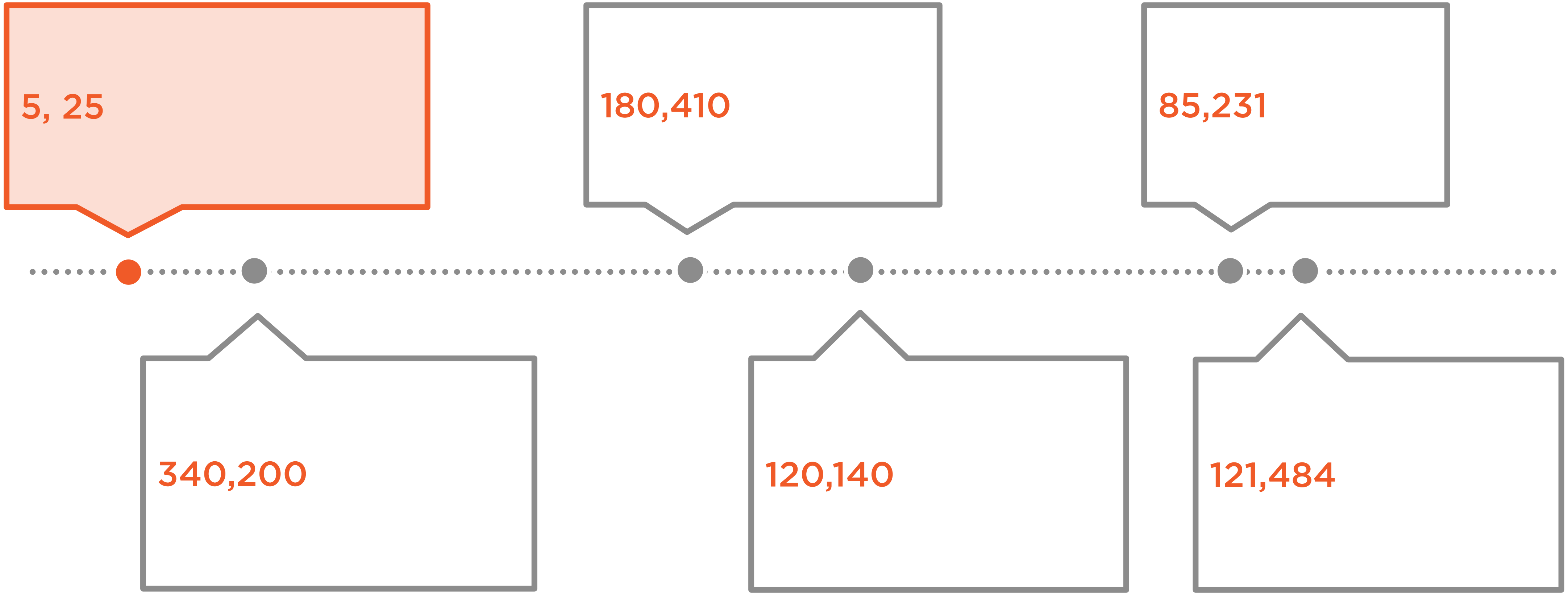
Promises

Represent a single value in the future

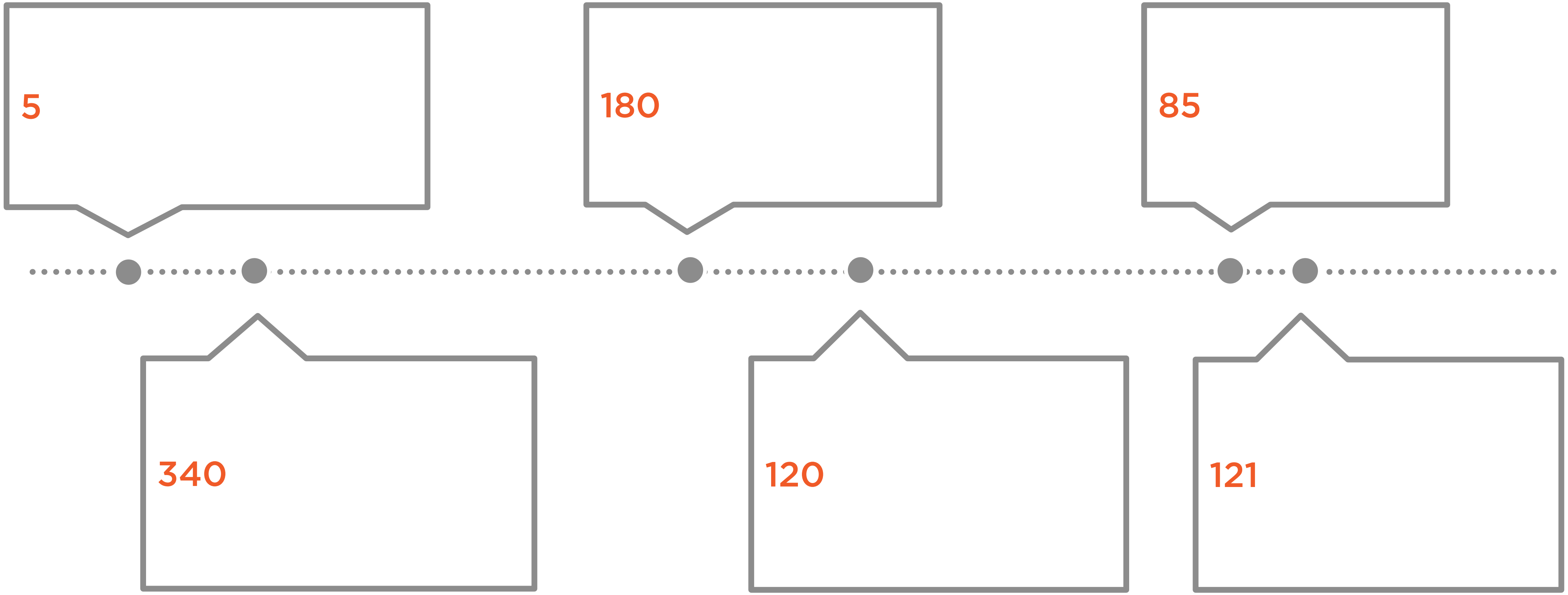
Observables

Represent 0 or more values now or in the future

Timeline of Events



Timeline of Events



Other Observable Features

Can Be Synchronous

Improved Error Handling

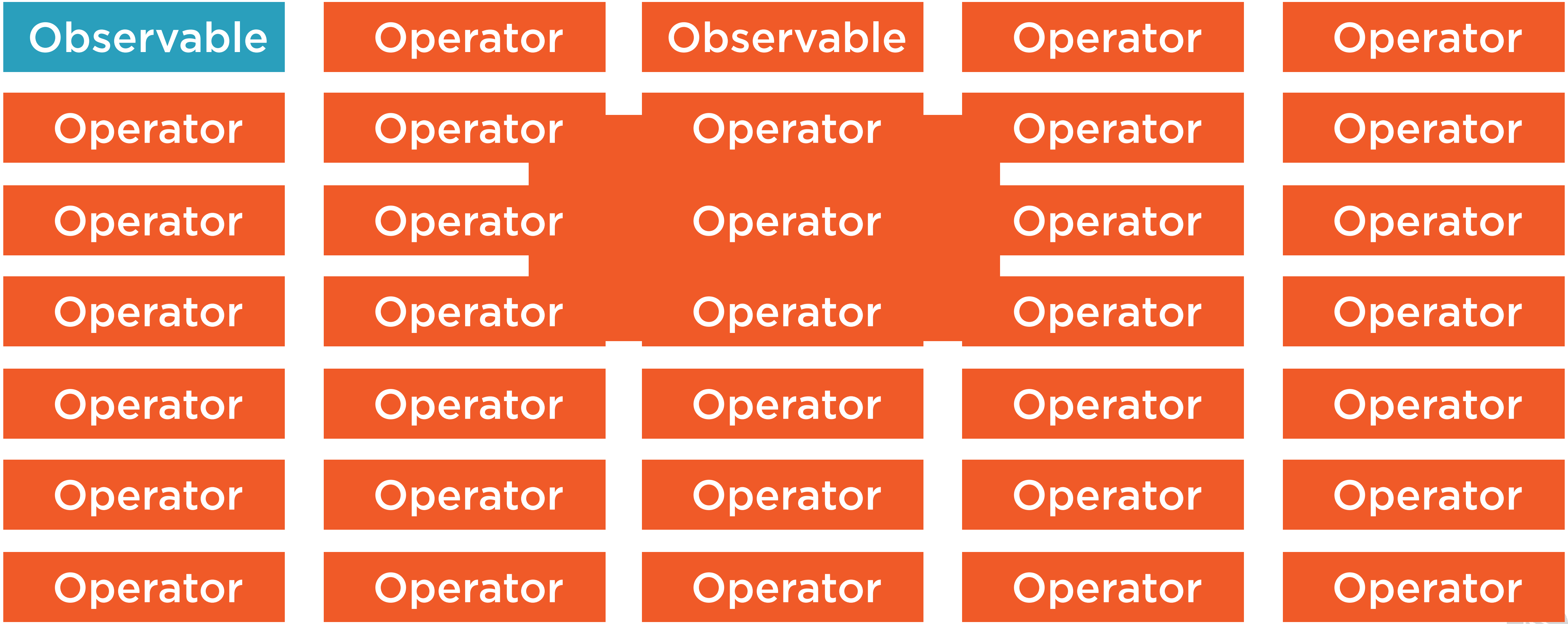
Can be Closed Independently of Returning a Value

Can Deal with Time

Advanced Operations

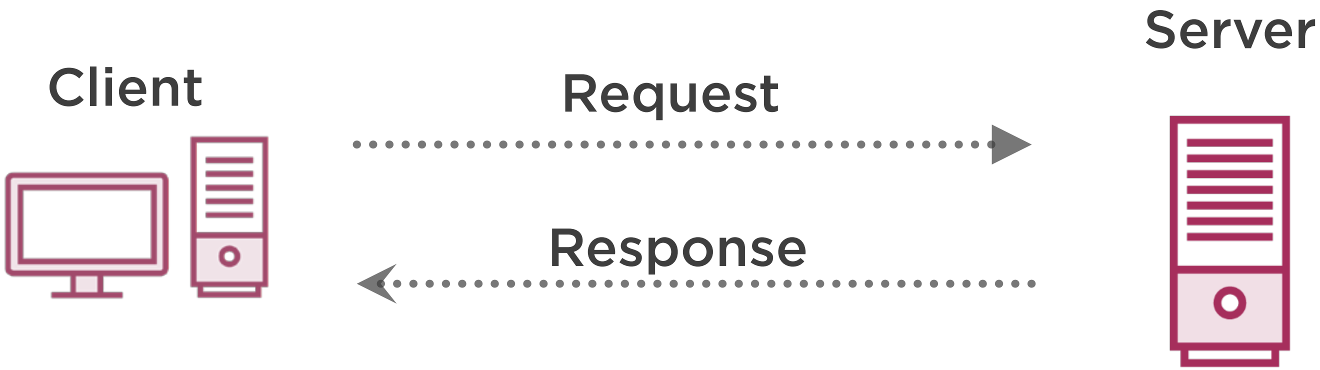
- Mathmetical Aggregation
- Buffering
- Debounce
- Distinct
- Filtering
- Combining Observables
- Retry

RxJS Library Requests



Observable	Operator	Observable	Operator	Operator
Operator	Operator	Operator	Operator	Operator
Operator	Operator	Operator	Operator	Operator
Operator	Operator	Operator	Operator	Operator
Operator	Operator	Operator	Operator	Operator
Operator	Operator	Operator	Operator	Operator
Operator	Operator	Operator	Operator	Operator

HTTP Communication



Client

Request

Response

Server

One Key Feature

`toPromise()`

Choose What’s Best For You

Summary

Observables

Data Storage on Server

Authentication