



Reactive Extensions for JavaScript - Operators

Alex Thalhammer

Outline

- Example
- Transformation Operators
- Filtering Operators
- Combination Operators
- Error Handling
- Higher Order Observables
- Reference



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Example



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Example with Pipeable Operators

```
import { map } from 'rxjs/operators';

this
  .http
  .get("http://www.angular.at/api/...")
  .pipe(map(flightDateStr => new Date(flightDateStr)))
  .subscribe({
    next: (bookings) => { ... },
    error: (err) => { console.error(err); }
  });
```



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Transformation Operators



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Operators

[<http://rxmarbles.com/#map>]



`map(x => 10 * x)`



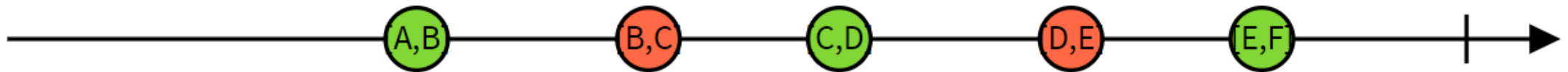


`pluck("a")`





pairwise



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

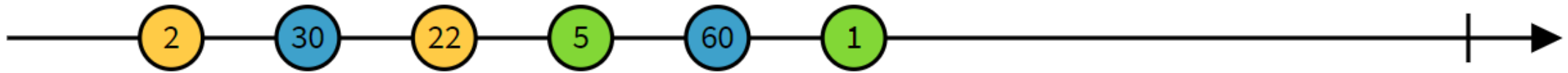
Filtering Operators



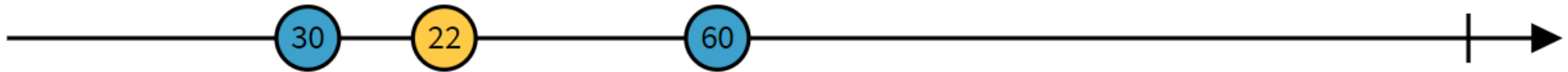
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



`filter(x => x > 10)`



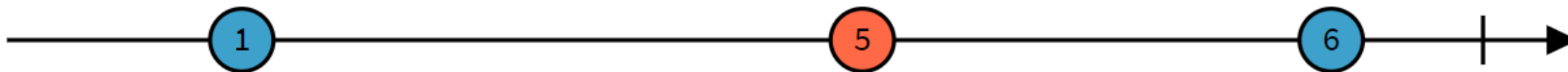
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



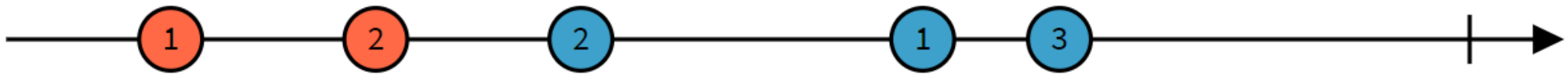
`debounceTime(10)`



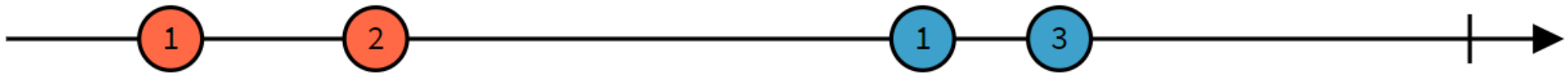
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



`distinctUntilChanged`



Demo

Simple Lookahead



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

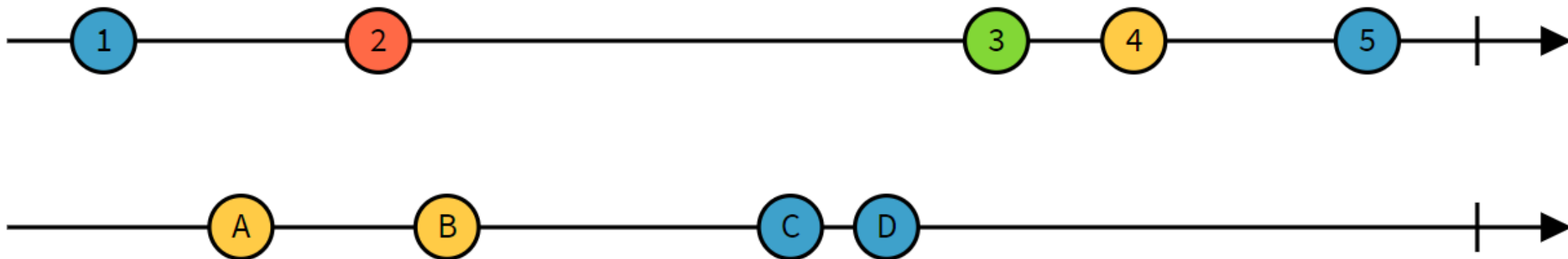
Combination Operators



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



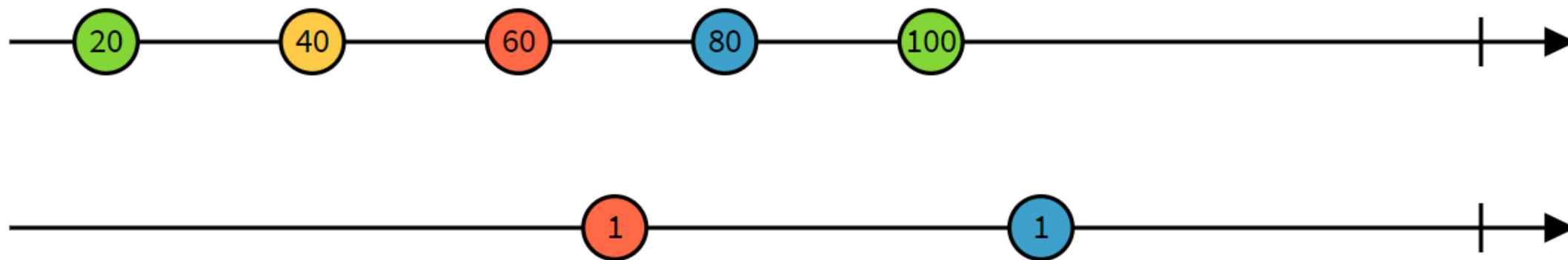
```
combineLatest((x, y) => "" + x + y)
```



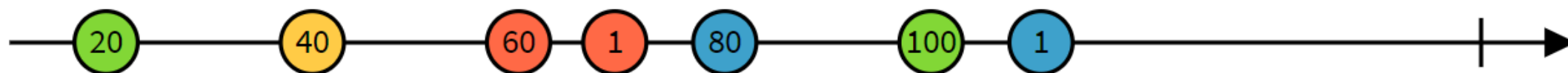
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



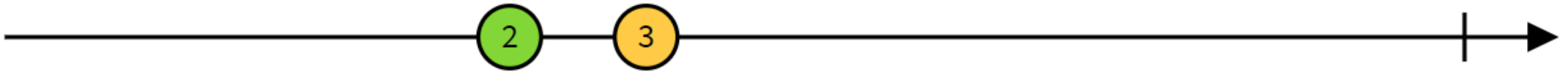
merge



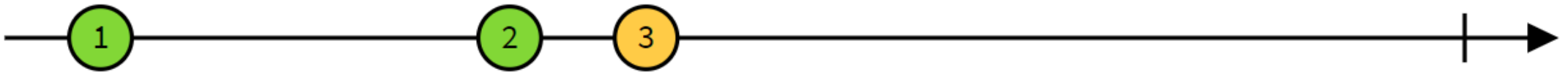
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



`startWith(1)`



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Demo

Combine Streams



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Error Handling



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Operators for Error Handling

- catchError
- retry
- retryWhen

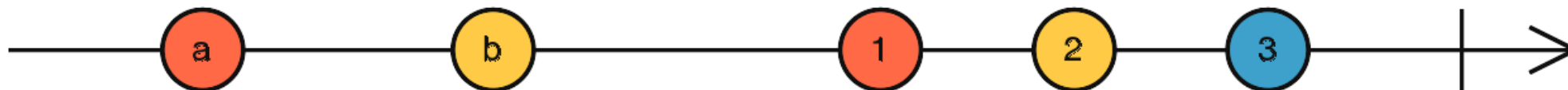
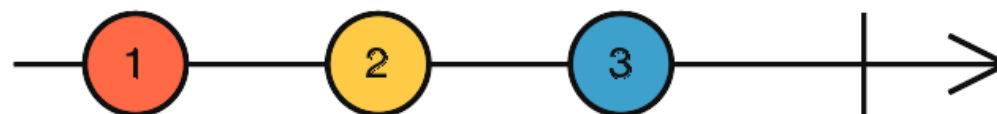
- throwError



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT



Demo

Error Handling



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Lab Time



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

One more thing 😊



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Higher Order Observables



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Operators for Higher Order Observables

- mergeMap
 - merges outer (source) and inner observables
- exhaustMap
 - outer is ignored until inner is finished
- switchMap
 - inner will be completed after next outer
- concatMap
 - outer will be sent after inner is finished



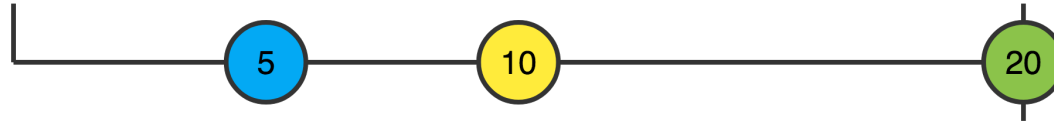
ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



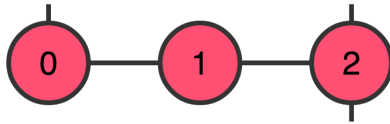
SOFTWARE
ARCHITECT



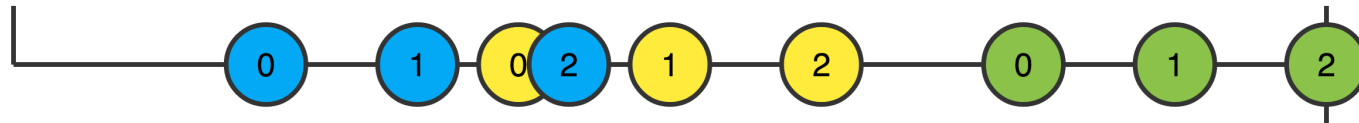
[source\$] A stream that emits at [5ms, 10ms, 20ms]



[target\$] will be mapped to a timer that emits at [N+0ms, N+3ms, N+6ms]



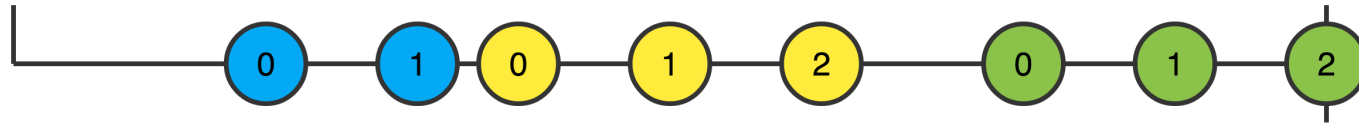
mergeMap



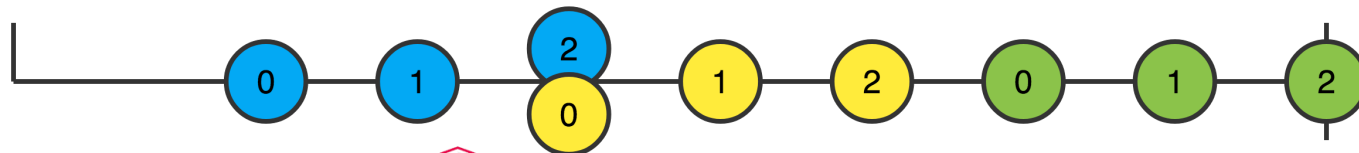
exhaustMap



switchMap



concatMap



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

Like this topic?

- Marble Diagrams
 - <http://rxmarbles.com>
- Other Links
 - <https://rxjs.dev/guide/overview>
 - <https://reactive.how/rxjs/>
 - <https://www.learnrxjs.io/>
 - <https://angular.io/guide/rx-library>
- Official documentation
 - <http://reactivex.io/rxjs/class/es6/Observable.js~Observable.html>



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT