

Programación Reactiva

RxJS

¿No era la
Programación
Funcional la última
chupá del mate?

Programación Reactiva

- Nace de:
 - Programación Funcional
 - Patrón Observador
 - Patrón Iterador

Programación Reactiva

- La idea es que todo sea un flujo de información asíncrono
- Los actores principales son los:
 - Observables
 - Observadores
- Componibles, así como los *.then* y *.catch* de las *promesas*.
- Se usa principalmente para interfaces gráficas complejas y aplicaciones en tiempo real



ReactiveX

<http://reactivex.io>

Implementaciones oficiales

- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Ruby: Rx.rb
- Python: RxPY
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift
- PHP: RxPHP

<https://github.com/ReactiveX/rxjs>

The screenshot shows the GitHub repository page for ReactiveX/rxjs. The browser address bar displays 'GitHub, Inc. github.com/ReactiveX/RxJS'. The repository name 'ReactiveX / rxjs' is at the top left. On the top right, there are buttons for 'Watch' (174), 'Star' (3,114), and 'Fork' (291). Below these, a navigation bar includes 'Code' (selected), 'Issues' (98), 'Pull requests' (8), 'Wiki', 'Pulse', and 'Graphs'. A description states: 'A reactive programming library for JavaScript <http://reactivex.io/rxjs>'. A statistics bar shows '1,907 commits', '4 branches', '29 releases', and '88 contributors'. Below this is a bar with 'Branch: master', a 'New pull request' button, a 'Find file' button, and a 'Clone or download' button. The commit history section shows a commit by 'jayphelps' titled 'Merge pull request #1939 from tklever/expect-observable-test-fix', with the latest commit '070e480' from 21 hours ago. A table of recent commits follows:

.github	fix(Rx): remove kitchenSink and DOM, let Rx export all	5 months ago
doc	feat(operator): Add repeatWhen operator	12 days ago
perf	test(sequenceEqual): add micro perf tests	28 days ago
spec	chore(typings): add missing expectSubscriptions constant	21 hours ago

Spoiler Alert

RxJS es una librería gigante y compleja

buffer
bufferCount
bufferTime
bufferToggle
bufferWhen
combineAll
combineLatest
concat
concatAll
concatMap
concatMapTo
count
debounce
debounceTime
defaultIfEmpty

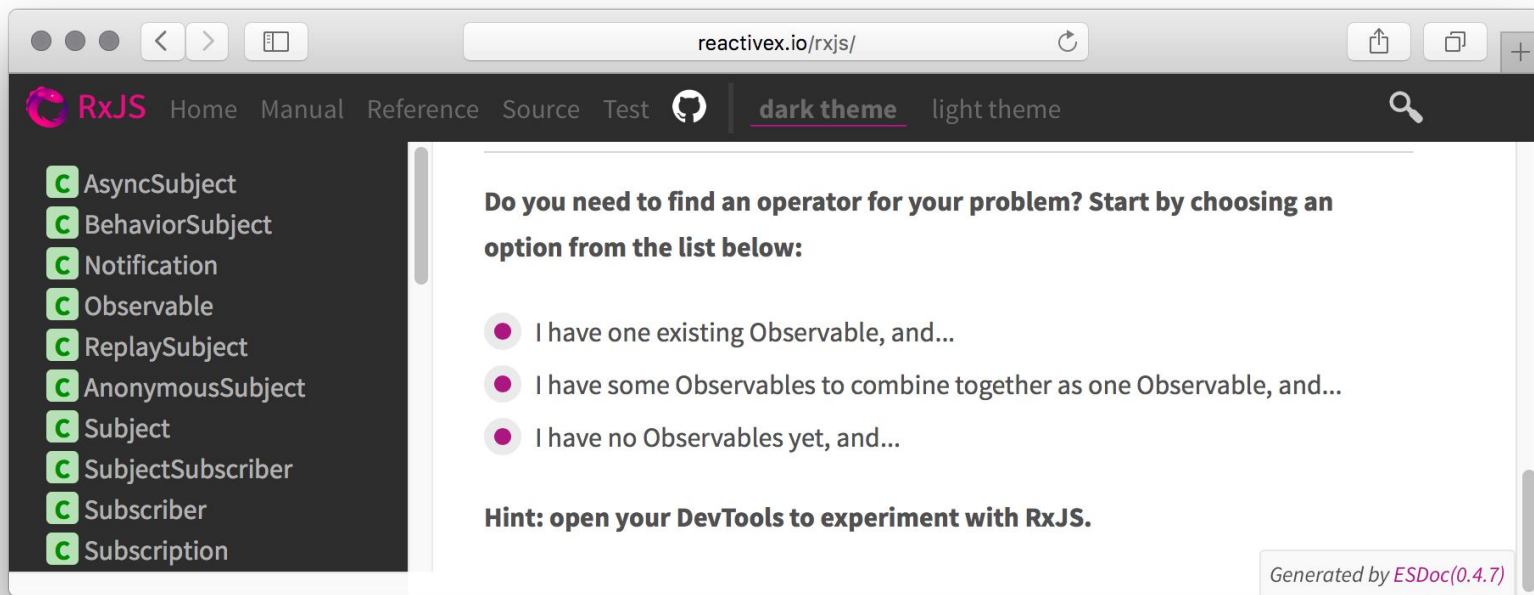
delay
delayWhen
dematerialize
distinctUntilChanged
do
every
expand
filter
first
groupBy
ignoreElements
last
let
map

mapTo
merge
mergeMap
partition
pluck
publish
race
repeat
retry
retryWhen
sample
scan
share
single

skip
skipUntil
skipWhile
startWith
switchMap
window
windowCount
windowTime
windowToggle
windowWhen
withLatestFrom
zip

Spoiler Alert

RxJS es una librería gigante y compleja



```
// From array of values  
Rx.Observable.from([1,2,3]);
```

```
// From an event  
Rx.Observable.fromEvent(document.querySelector('button'), 'click');
```

```
// From a Promise  
Rx.Observable.fromPromise(fetch('/users'));
```

```
3 import Rx from 'rxjs/Rx';
4
5 const source = Rx.Observable.of(1,2,3)
6   .map(x => x * 2);
7
8 const subscription = source
9   .subscribe(value => console.log(value));
10
11 subscription.unsubscribe();
```

```
17 const button = document.querySelector('#my-button');
18
19 Rx.Observable.fromEvent(button, 'click')
20   .throttleTime(1000 /* ms */)
21   .scan(count => count + 1, 0)
22   .subscribe(count => console.log('Total counts:', count));
23
```

```
34 const racing = Rx.Observable.race(...[
35   Rx.Observable.interval(1500).mapTo('1'),
36   Rx.Observable.interval(1000).mapTo('2'),
37   Rx.Observable.interval(2000).mapTo('3'),
38   Rx.Observable.interval(2500).mapTo('4'),
39 ]);
40
41 racing.subscribe(winner => console.log(winner));
42
```

```
15 const hello = Rx.Observable.create(observer => {
16     observer.next('Hello');
17     observer.next('World');
18 });
19
20 const subscribe = hello
21     .subscribe(val => console.log(val));
22
```

¿Quién usa esto?



Cycle.js

Referencias

- <http://reactivex.io/rxjs/manual/tutorial.html>
- <https://egghead.io/lessons/rxjs-reactive-programming-what-is-rxjs>
- <https://www.learnrxjs.io/>