

# merge

---

Crea un Observable de salida que emite concurrentemente los valores de todos los Observables de entrada

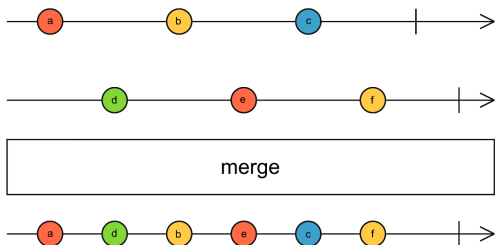
---

## ► Signatura

---

## Descripción

Une varios Observables en uno solo.



`merge` se suscribe a cada Observable de entrada de forma concurrente, y emite sus valores, sin transformarlos, en el Observable resultante. El Observable resultante solo se completa cuando todos los Observables de entrada se hayan completado.

Cualquier error lanzado por un Observable de entrada será emitido inmediatamente en el Observable resultante.

---

## Ejemplos

### Unir dos Observables ajax

[StackBlitz](#)

```
import { concat } from "rxjs";
import { ajax } from "rxjs/ajax";

const totoroFilmData$ = ajax.getJSON(
  "https://ghibliapi.herokuapp.com/films/58611129-2dbc-4a81-a72f-77ddfc1b1b4";
);

const charmanderData$ = ajax.getJSON("https://pokeapi.co/api/v2/pokemon/4");

concat(totoroFilmData$, charmanderData$).subscribe(console.log);
// Salida: {..., title: 'My Neighbor Totoro', ...}, { abilities: [], ...}
```

**merge** se suscribe de forma concurrente (a la vez) a todos los Observables de entrada, mientras que **concat** se suscribe a ellos por orden, y hasta que el primero no se complete, no se suscribe al siguiente. Comparación entre **concat** y **merge**:

StackBlitz

```
import { concat, merge, timer } from "rxjs";
import { mapTo } from "rxjs/operators";

const first$ = timer(3000).pipe(mapTo("Primero"));

const second$ = timer(1000).pipe(mapTo("Segundo"));

const third$ = timer(2000).pipe(mapTo("Tercero"));

merge(first$, second$, third$).subscribe(console.log);
// Salida: (1s) Segundo (1s) Tercero (1s) Primero

concat(first$, second$, third$).subscribe(console.log);
// Salida: (3s) Primero (1s) Segundo (2s) Tercero
```

Ejemplos de la documentación oficial

Unir 2 Observables: 1s interval y clicks

```
import { merge, fromEvent, interval } from "rxjs";
```

```
const clicks = fromEvent(document, "click");
const timer = interval(1000);
const clicksOrTimer = merge(clicks, timer);
clicksOrTimer.subscribe(x => console.log(x));

// Salida:
// timer emite valores ascendentes, uno cada segundo (1000ms)
// clicks imprime MouseEvents por console cada vez que se haga click en el
```

**Unir 3 Observables, pero solo ejecutar 2 de forma concurrente**