

zip

Combina varios Observables para crear otro Observable cuyos valores se calculen a partir de las emisiones, en orden, de cada uno de sus Observables de entrada

► Signatura

Descripción

Si el último parámetro es una función, esta se utiliza para computar el valor creado a partir de los valores de entrada. Si no, se retorna un array de los valores de entrada.

Ejemplos

Esperar a que dos Observables emitan un valor, y emitir ambos valores en un array

StackBlitz

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

const hello$ = timer(2000).pipe(mapTo("Hello"));
const world$ = timer(4000).pipe(mapTo("World"));

zip(hello$, world$).subscribe(console.log);
// Salida: ['Hello', 'World']
```

[Copy](#)

Al combinarlo con interval (o timer), zip puede utilizarse para emitir los valores de un Observable cada cierto tiempo. Ej: Emitir una cadena cada vez que interval emite (cada segundo)

[StackBlitz](#)

```
import { from, interval, zip } from "rxjs";
import { map } from "rxjs/operators";

const framework$ = zip(from(["Angular", "React", "Vue"]), interval(1000)).pipe(
  map(([framework]) => framework)
);

framework$.subscribe(console.log);
// Salida: (1s) Angular (1s) React (1s) Vue
```

Ejemplo de la documentación oficial

Combinar la edad y el nombre de distintas fuentes