

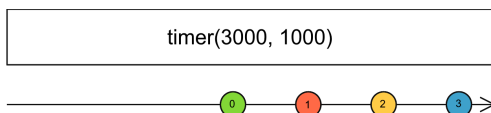
timer

Crea un Observable que comienza a emitir una secuencia ascendente de números consecutivos a intervalos, tras un periodo inicial de tiempo

► Signatura

Descripción

Es como `interval`, pero se puede especificar cuándo deben comenzar las emisiones.



`timer` retorna un Observable que emite una secuencia de números ascendentes infinita. Los valores se emiten a intervalos constantes de tiempo, según se especifique. La primera emisión ocurre tras el periodo de tiempo especificado por `dueTime`. Este periodo inicial de espera también se puede proporcionar en formato `Date`.

Por defecto, este operador utiliza el `SchedulerLike` `asyncScheduler` para proporcionar la noción del tiempo, pero se le puede proporcionar cualquier otro `SchedulerLike`.

Si no se proporciona el valor `period`, el Observable resultante emite un único valor, 0. Si se proporciona un valor `period`, se emite una secuencia infinita.

Ejemplos

Emitir un único valor, 0, tras 2 segundos de espera

[StackBlitz](#)

```
import { timer } from "rxjs";

const zero$ = timer(2000);

zero$.subscribe(console.log);
// Salida: 0
```

[Copy](#)

Emitir una secuencia ascendente de números a intervalos de 1 segundo, tras 5 segundos de espera

[StackBlitz](#)

```
import { timer } from "rxjs";

const number$ = timer(5000, 1000);

number$.subscribe((number) => console.log(number));
// Output: 0, 1, 2, 3...
```

[Copy](#)

Ejemplos de la documentación oficial

Emitir una secuencia ascendente de números, uno cada segundo (1000ms), comenzando tras 3 segundos

```
import { timer } from "rxjs";

const numbers = timer(3000, 1000);

numbers.subscribe((x) => console.log(x));
```

[Copy](#)

Emitir el número 0 tras 5 segundos de espera