

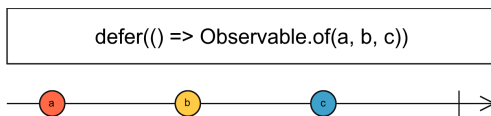
defer

Crea un Observable que, al ser suscrito, llama a una factoría Observable para crear otro Observable nuevo por cada suscriptor

► Signatura

Descripción

El Observable se crea de forma diferida, es decir, solamente se crea cuando un Observador se suscribe a él.



`defer` nos permite crear Observables únicamente cuando un Observador se suscribe, y crear un Observable nuevo para cada Observador.

Espera a que un Observador se suscriba a él, y entonces genera un Observable nuevo, normalmente mediante una función factoría Observable.

Este proceso se lleva a cabo para cada suscriptor, por lo que, aunque cada suscriptor crea que se está suscribiendo al mismo Observable, en realidad cada suscriptor recibe su propio Observable.

Ejemplos

Emitir la fecha/hora en el momento de la suscripción

[StackBlitz](#)

```
import { defer, of } from "rxjs";

const deferredTime$ = defer(() => of(new Date()));

deferredTime$.subscribe(console.log);
// Salida: La fecha en el momento de la suscripción
```

Copy

Crear un Observable que emita una fruta distinta cada vez que un observador se suscribe a él

[StackBlitz](#)

```
import { defer, of } from "rxjs";

const fruits = ["Cereza", "Fresa", "Mora", "Arándano"];

function getRandomFruit() {
  return fruits[Math.floor(Math.random() * 3)];
}

// Cada suscriptor a este Observable recibirá SIEMPRE la misma fruta, ya que
const randomFruit$ = of(getRandomFruit());

// Cada suscriptor a este Observable recibirá una fruta DISTINTA cada vez, ya que
const randomFruitForReal$ = defer(() => of(getRandomFruit()));

randomFruit$.subscribe(console.log);
// Salida: Cereza
randomFruit$.subscribe(console.log);
// Salida: Cereza

randomFruitForReal$.subscribe(console.log);
// Salida: Fresa
randomFruitForReal$.subscribe(console.log);
// Salida: Mora
```

Copy

Ejemplo de la documentación oficial

Suscribirse a un Observable de clicks o a un Observable intervalo, de forma aleatoria