

RxJS en Angular: Subject, BehaviorSubject y ReplaySubject

Introducción

En Angular, RxJS ofrece tres herramientas clave para manejar flujos de datos: Subject, BehaviorSubject y ReplaySubject. Cada uno tiene comportamientos diferentes y se adapta a distintos casos de uso en la programación real.

1. Subject

Un Subject es un emisor que no guarda valores pasados. Solo entrega a cada suscriptor los valores emitidos después de su suscripción.

Caso real:

En un chat en vivo, si un usuario entra tarde, no ve los mensajes anteriores, solo los nuevos que se envían desde que se conecta.

Ejemplo:

```
const mensaje$ = new Subject<string>();
mensaje$.subscribe(m => console.log('Usuario A:', m));
mensaje$.next('Hola a todos');
mensaje$.next('Nuevo mensaje');
```

En este caso, solo los suscriptores actuales reciben los mensajes en tiempo real.

2. BehaviorSubject

Un BehaviorSubject guarda el último valor emitido y lo entrega a los nuevos suscriptores. Necesita un valor inicial.

Caso real:

En un sistema de autenticación, queremos que cualquier componente que pregunte por el usuario actual reciba la última información, incluso si se suscribe más tarde.

Ejemplo:

```
const usuario$ = new BehaviorSubject<Usuario | null>(null);
usuario$.next({ nombre: 'Rafael', rol: 'admin' });
usuario$.subscribe(u => console.log('Usuario actual:', u));
```

Esto asegura que cualquier nuevo componente siempre reciba el usuario actual.

3. ReplaySubject

Un ReplaySubject guarda múltiples valores emitidos y los entrega a cualquier nuevo suscriptor. Podemos limitar la cantidad o el tiempo de almacenamiento.

Caso real:

En un sistema de reproducción de logs, queremos que un nuevo administrador vea las últimas 5 acciones realizadas en la app.

RxJS en Angular: Subject, BehaviorSubject y ReplaySubject

Ejemplo:

```
const acciones$ = new ReplaySubject<string>(5);
acciones$.next('Login exitoso');
acciones$.next('Producto creado');
acciones$.next('Producto editado');
acciones$.subscribe(a => console.log('Historial:', a));
```

Así, un suscriptor nuevo recibe las últimas acciones aunque se haya suscrito tarde.

Resumen

- Subject: Para eventos en tiempo real sin historial.
- BehaviorSubject: Para mantener y compartir el último estado.
- ReplaySubject: Para compartir varios valores pasados (historial).

En la vida real, la elección depende de si necesitas historial, el último estado o solo eventos actuales.