Combining Operators

Đề cương

1. Tiền tố và nối

2. Hợp nhất

3. Kết hợp các yếu tố

4. Kích hoạt

5. Kết hợp phần tử trong một chuỗi

1. Tiền tố và nối

1. startWith ()

2. concat ()

3. concatMap ()

1.1 startWith ()

❖ startWith (\_ :) cho phép tạo một phần tử ban đầu cho một Observable. Giá trị này phải cùng loại với các phần tử khác trong Observable.

1.2 concat ()

❖ Toán tử này nối các Quan sát một cách tuần tự thành một Quan sát nguồn.

❖ Nguồn Observable sẽ hoàn thành chỉ khi tất cả các Observables con hoàn thành.

❖ Nếu một trong các Observable con phát ra lỗi, nguồn Observable sẽ phát ra lỗi ngay lập tức và chấm dứt.

1.3 concatMap ()

❖ Toán tử này ánh xạ các giá trị đến các giá trị có thể quan sát bên trong, đăng ký và phát ra theo thứ tự.

❖ Sự khác biệt duy nhất giữa flatMapLatest và concatMap là nó duy trì thứ tự của các mục.

❖ Nó có một luồng, không tốt cho các hoạt động không đồng bộ vì chờ người có thể quan sát để hoàn thành tất cả các mục theo trình tự.

2. Hợp nhất

1. merge ()

2.1. hợp nhất ()

Toán tử merge (\_ :) cho phép kết hợp nhiều Observable bằng cách hợp nhất các phát xạ của chúng thành một nguồn có thể Observable.

❖ Nguồn có thể quan sát được sẽ chỉ hoàn thành nếu tất cả các Chuỗi bên trong và nguồn có thể quan sát được đều hoàn thành.

❖ Tất cả các Chuỗi bên trong hoạt động độc lập.

❖ Nếu bất kỳ Chuỗi bên trong nào phát ra Lỗi, nguồn Observable sẽ phát ra Lỗi ngay lập tức và kết thúc.

3. Kết hợp các yếu tố

1. connectLatest (\_: \_: resultSelector :)

2. zip (\_: \_: resultSelector :)

3.1. connectLatest (\_: \_: resultSelector :)

❖ Mỗi khi một trong các chuỗi bên trong (kết hợp) phát ra một giá trị, nó sẽ gọi một lần đóng mà bạn cung cấp. Bạn nhận được giá trị cuối cùng được phát ra bởi mỗi chuỗi bên trong.

❖ Giống như toán tử map (\_ :), connectLatest (\_: \_: resultSelector :) tạo ra một kiểu có thể quan sát được là kiểu trả về bao đóng.

3.2. zip (\_: \_: resultSelector :)

❖ Toán tử này kết hợp các phần tử có cùng chỉ số trong Observable thành một phần tử mới và trả về cho người đăng ký.

❖ Nếu một trong các Observables đầu vào không có phần tử thích hợp tại chỉ mục trong các Observables khác, zip (\_: \_: resultSelector :) sẽ không phát ra phần tử.

4. Kích hoạt

1. withLatestFrom (\_ :)

4.1 withLatestFrom (\_ :)

❖ Toán tử này nhận một Observable làm đầu vào và biến đổi một trình kích hoạt thành sự kiện mới nhất được phát ra từ đầu vào Observable.

5. Kết hợp phần tử trong một chuỗi

1. giảm (\_: \_ :)

2. quét (\_: \_ :)

5.1 giảm (\_: \_ :)

❖ Toán tử này áp dụng một hàm cho mỗi mục được phát ra bởi một giá trị có thể Quan sát, tuần tự và phát ra giá trị cuối cùng.

5.2 quét (\_: \_ :)

❖ Toán tử này áp dụng một hàm cho mỗi mục được phát ra bởi một Giá trị quan sát được, tuần tự và phát ra từng giá trị liên tiếp.