Đề cương

1. Bắt đầu

2. Bỏ qua các toán tử

3. Bỏ qua toán tử

4. Lấy các nhà khai thác

5. Các toán tử riêng biệt

1. Bắt đầu

❖ Các toán tử lọc cho phép bạn áp dụng các ràng buộc có điều kiện cho các sự kiện .next, để người đăng ký chỉ nhận các phần tử mà nó muốn xử lý.

❖ Nếu bạn đã từng sử dụng phương thức filter (\_ :) trong thư viện chuẩn Swift, bạn đã đi được nửa chặng đường

2. Bỏ qua các toán tử

1. ignoreElements ()

2. elementAt ()

3. bộ lọc ()

2.1. lờ đi ()

❖ Như được mô tả trong sơ đồ cẩm thạch sau đây, ignoreElements () sẽ làm điều đó: bỏ qua các phần tử sự kiện .next. Tuy nhiên, nó sẽ cho phép các sự kiện dừng thông qua, chẳng hạn như các sự kiện .completed hoặc .error.

2.2. elementAt ()

❖ Khi bạn chỉ muốn xử lý phần tử thứ n (thứ tự) được phát ra bởi một phần tử có thể quan sát được, bạn có thể sử dụng elementAt () lấy chỉ mục của phần tử bạn muốn nhận và nó bỏ qua mọi thứ khác.

2.3. lọc()

❖ ignoreElements () và elementAt () là các phần tử lọc được phát ra bởi một phần tử có thể quan sát được. Khi nhu cầu lọc của bạn vượt ra ngoài tất cả hoặc một, sẽ có bộ lọc. Nó có một bao đóng vị từ, nó áp dụng cho mọi phần tử được phát ra, chỉ cho phép thông qua những phần tử mà vị từ phân giải thành true.

3. Bỏ qua toán tử

1. bỏ qua ()

2. bỏ quaWhile ()

3. bỏ quaUntil ()

3.1. nhảy()

❖ Toán tử bỏ qua cho phép bạn bỏ qua từ số 1 đến số bạn chuyển làm tham số của nó.

3.2. bỏ quaWhile ()

❖ Giống như bộ lọc (\_ :), ignoreWhile (\_ :) cho phép bạn bao gồm một vị từ để xác định những gì nên được bỏ qua.

❖ Tuy nhiên, không giống như bộ lọc (\_ :), sẽ lọc các phần tử cho vòng đời của đăng ký, bỏ quaWhile sẽ chỉ bỏ qua cho đến khi nội dung nào đó không bị bỏ qua và sau đó sẽ cho phép mọi thứ khác đi qua kể từ thời điểm đó.

❖ Và với skipsWhile (\_ :), trả về true sẽ khiến phần tử bị bỏ qua và trả về false sẽ cho qua. Nó ngược lại với bộ lọc.

3.3. bỏ qua cho đến khi ()

❖ Toán tử này sẽ tiếp tục bỏ qua các phần tử từ nguồn có thể quan sát được (nguồn mà bạn đang đăng ký) cho đến khi một số kích hoạt khác có thể quan sát được phát ra.

4. Lấy các nhà khai thác

1. lấy ()

2. takeWhile ()

3. takeUntil ()

4.1. cầm lấy()

❖ Toán tử này chỉ phát ra phần tử đầu tiên trong số phần tử bạn đã chỉ định được phát ra bởi nguồn Observable.

4.2. takeWhile ()

❖ takeWhile hoạt động tương tự như bỏ quaWhile, ngoại trừ việc bạn đang sử dụng thay vì bỏ qua.

4.3. takeUntil ()

❖ Phát ra các giá trị do nguồn có thể quan sát được cho đến khi nguồn có thể quan sát khác phát ra giá trị.

5. Các toán tử riêng biệt

1. diffUntilChanged ()

5.1. Riêng biệtUntilChanged ()

❖ differenceUntilChanged () ngăn các mục liền kề trùng lặp đi qua.