



01



◦1-7. *Flow Terminal operator*



Coroutine

Terminal operator

- Flow 를 이용하기 위한 함수를 operator 라고 통칭
- Flow 의 데이터를 구독해 최종 이용하는 Terminal operator
- Flow 에 어떤 내용을 추가해 다시 Flow 를 만들어 주는 Intermediator operator
- Terminal operator 는 Flow 의 확장함수로 제공되는 함수들이며 Flow 데이터를 구독하는 역할을 하는 함수
- 가장 대표적인 Terminal operator 가 collect()

first()

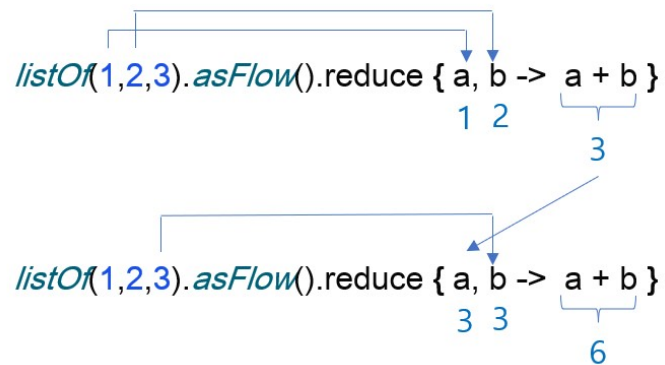
- first() 는 Flow 가 발행하는 첫번째 데이터를 구독하기 위한 operator
- 구독한 데이터를 획득만 하면 되면 first() 를 이용, 조건을 명시해 조건에 만족하는 첫번째 데이터를 획득하고 싶다면 first { } 를 이용
- first() 혹은 first { } 에 의해 획득되는 데이터가 없다면 예외가 발생
- firstOrNull() 혹은 firstOrNull { } 을 이용하면 획득되는 데이터가 없는 경우 null 을 리턴

single()

- single() 은 하나의 값만 구독하기 위한 Terminal operator
- single() 은 flow 가 종료될 때 까지 대기했다가 flow 가 종료되면 발행한 값을 리턴
- Flow 에서 발행하는 값이 없거나 하나 이상의 값을 발행하게 되면 예외가 발생

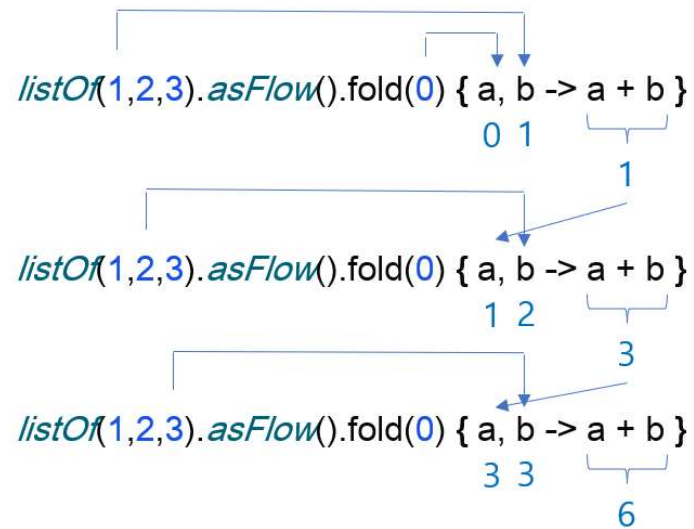
reduce(), fold()

- `reduce()` 와 `fold()` 는 Flow 발행에 의해 실행된 이전 결과 값을 새로운 데이터 발행에 의해 실행될 때 전달되어 참조할 수 있게 해주는 operator



reduce(), fold()

- fold() 는 reduce() 와 동일하게 동작하며 초기값 지정이 가능



toList(), toSet()

- toList(), toSet() 은 Flow 에 의해 발행되는 데이터를 MutableList, MutableSet 에 담아 반환시켜 주는 operator
- 모든 발행이 완료된 후 발행된 데이터를 모아 한꺼번에 처리하고자 할 때 유용

collectIndexed(), collectLatest()

- collectIndexed() 에 지정되는 함수는 매개변수를 두개 가지고 있으며 Flow 가 발행한 데이터 이외에 호출한 Index 값까지 전달
- collectLatest() 는 Flow 의 데이터 발행에 의해 collectLatest() 부분이 실행되는 도중 Flow 가 새로운 데이터를 발행하게 되면 collectLatest() 실행이 멈추고 새로운 데이터로 다시 실행

count()

- count() 는 Flow 에서 발행한 데이터의 개수를 반환시켜 주는 operator

launchIn()

- Flow 의 데이터 수신은 Blocking 으로 실행
- collect() 를 Non-Blocking 으로 실행하고 싶다면 코루틴을 만들고 그 코루틴에서 collect() 하게 해주어야 한다.
- launchIn() 은 코루틴에서 Flow 데이터를 이용되게 해주는 operator
- launchIn() 매개변수에 코루틴 스코프를 지정하면 자동으로 지정된 스코프내에서 코루틴이 실행
- launchIn() 으로 코루틴을 구동시킨 것임으로 리턴타입은 Job



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare