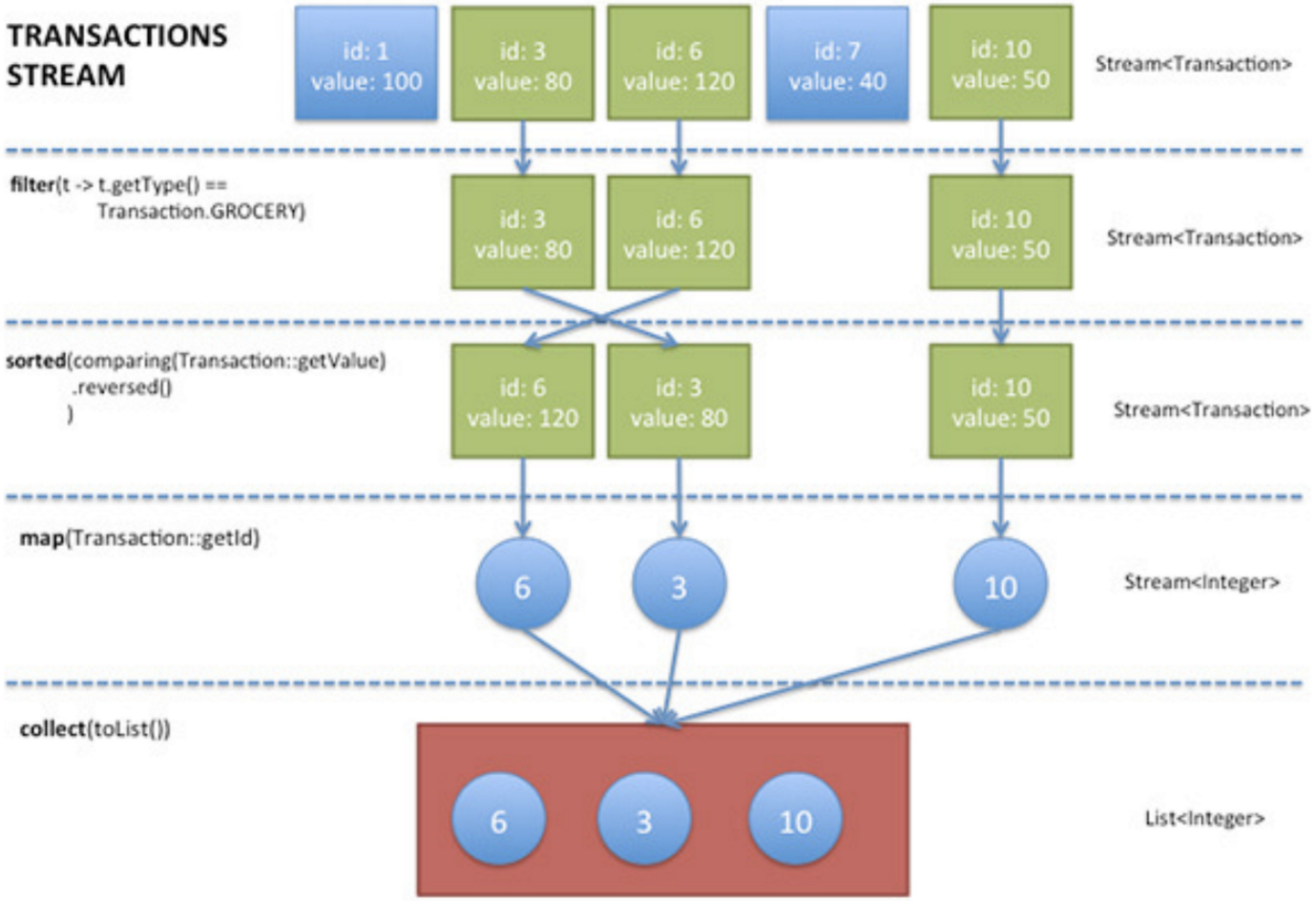
Java SE 8 Stream API

2017년 12월 28일 목요일 오전 11:28

Stream API

- Java SE 8이 출시되면서 Stream이라는 새로운 API가 공개되었고, 이를 통해 쿼리를 작성하듯 데이터를 명시적인 방법으로 처리할 수 있게 되었다.
- Stream은 멀티 스레드 관련 코드도 별도로 작성할 필요없이 멀티 코어를 지원한다.
- Stream() 메스드를 통해 Stream을 가져오고, 그 다음 여러가지 기능을 체인처럼 엮어 데이터를 처리한다.
    - Stream()
    - filter, sorted, map, collect
- Stream의 정의는 "집계 연산을 지원하는 요소의 순서 (a Sequence of Elements form a Source that supports Aggregate of Operations)"이다.
    - Sequence of Elements: Stream은 정의된 엘리먼트의 속성에 따라서 처리할 수 있는 인터페이스를 제공하지만 실제 엘리먼트들을 저장하지 않고 계산하는데에만 쓰인다.
    - Source: Stream은 컬렉션, 배열, I/O 리소스 등에서 제공받은 데이터를 가지고 작업을 처리한다.
    - Aggregate Operations: Stream은 SQL 같은 처리를 지원하고, 함수형 같은 처리 방법도 지원한다.
        - filter, map, reduce, find, match, sorted ...
- Stream은 기존 컬렉션 처리 방법과 구분되는 두 가지 특징이 있다.
    - Pipelining: 많은 Stream 기능들은 Stream 자기 자신을 리턴한다. 이 방식은 처리 작업이 체인처럼 연결되어 큰 파이프라인처럼 동작한다. 이를 통해 laziness와 short-circuiting과 같은 최적화 작업이 가능하다.
    - Internal Iteration: 명시적으로 반복 작업을 수행해야 하는 Collection과 비교하면 Stream 작업은 내부에서 처리한다.

처리 흐름



- 처음 Stream은 List에서 거래 내역에 대한 목록을 stream() 메서를 사용하여 Stream으로 가져온다.
- 다음 몇 가지 집계와 관련된 작업을 진행한다.
    - filter: 주어진 속성에 맞게 필터링
    - sorted: 주어진 Comparator를 통해 정렬
    - map: 데이터 추출
- 위의 메서드들은 Stream을 반환하기 때문에, 서로 연결되어 파이프라인 형태로 처리된다.
- 마지막으로 collect() 메서드를 통해 값을 반환하며, 해당 메서드의 파라미터를 가지고 반호나 값 타입을 결정한다.
- collect() 메서드를 제외한 모든 작업들은 collect() 메서드가 호출되기 전까지 실제로 실행되지 않는다. collect()는 파이프라인에 연결된 집계 작업을 시작하고 완료되면 결과를 반환한다.

Stream Vs Collection

- Collection은 데이터와 관련된 것이며, Stream은 데이터를 계산하는 것이다.
- Collection은 메모리에 저장되는 데이터 구조이며, Stream은 이를 계산해서 새로운 데이터를 만들어낸다.
- Collection 인터페이스를 사용하기 위해서는 사용자에 의해 정의된 반복문은 외부 반복을 사용하며, Stream은 내부 반복을 사용하고 사용자는 결과 값을 가지고 무엇을 할 것인지만 정의하면 된다.

- [ ] 나머지는 다음에
  http://starplatina.tistory.com/entry/%EC%9E%90%EB%B0%94-%EC%8A%A4%ED%8A%B8%EB%A6%BC-API