

노인과 바다

Infinite Scroll

목차

1. Slice VS Page

2. Offset VS No - Offset

3. 구현 코드

4. 구현 영상



Slice

Slice



```
public interface Slice<T> extends Streamable<T>
```

Slice

Modifier and Type	Method	Description
List<T>	getContent()	페이지 내용을 리스트로 반환한다.
int	getNumber()	현재 페이지의 번호를 반환한다.
int	getNumberOfElements()	현재 페이지의 데이터 개수를 반환한다.
int	getSize()	슬라이스의 size를 반환한다.
boolean	hasContent()	슬라이스에 데이터가 존재하는지 확인한다.
boolean	hasNext()	다음 슬라이스가 존재하는지 확인한다.
boolean	hasPrevious()	이전 슬라이스가 존재하는지 확인한다.
default Pageable	nextOrLastPageable()	다음 Pageable을 반환한다. 단, 현재가 마지막 슬라이스면 현재 Pageable을 반환한다.
Pageable	nextPageable()	다음 Pageable을 반환한다.
default Pageable	previousOrFirstPageable()	이전 Pageable을 반환한다. 단, 현재가 첫 슬라이스면 현재 Pageable을 반환한다.
Pageable	previousPageable()	이전 Pageable을 반환한다.



Page

Page



```
public interface Page<T> extends Slice<T>
```




Page

Modifier and Type	Method	Method
long	<code>getTotalElements()</code>	전체 데이터의 개수를 반환한다.
<code>int</code>	<code>getTotalPages()</code>	전체 페이지 수를 반환한다.



Offset



Offset

(현재 페이지수-1)*limit



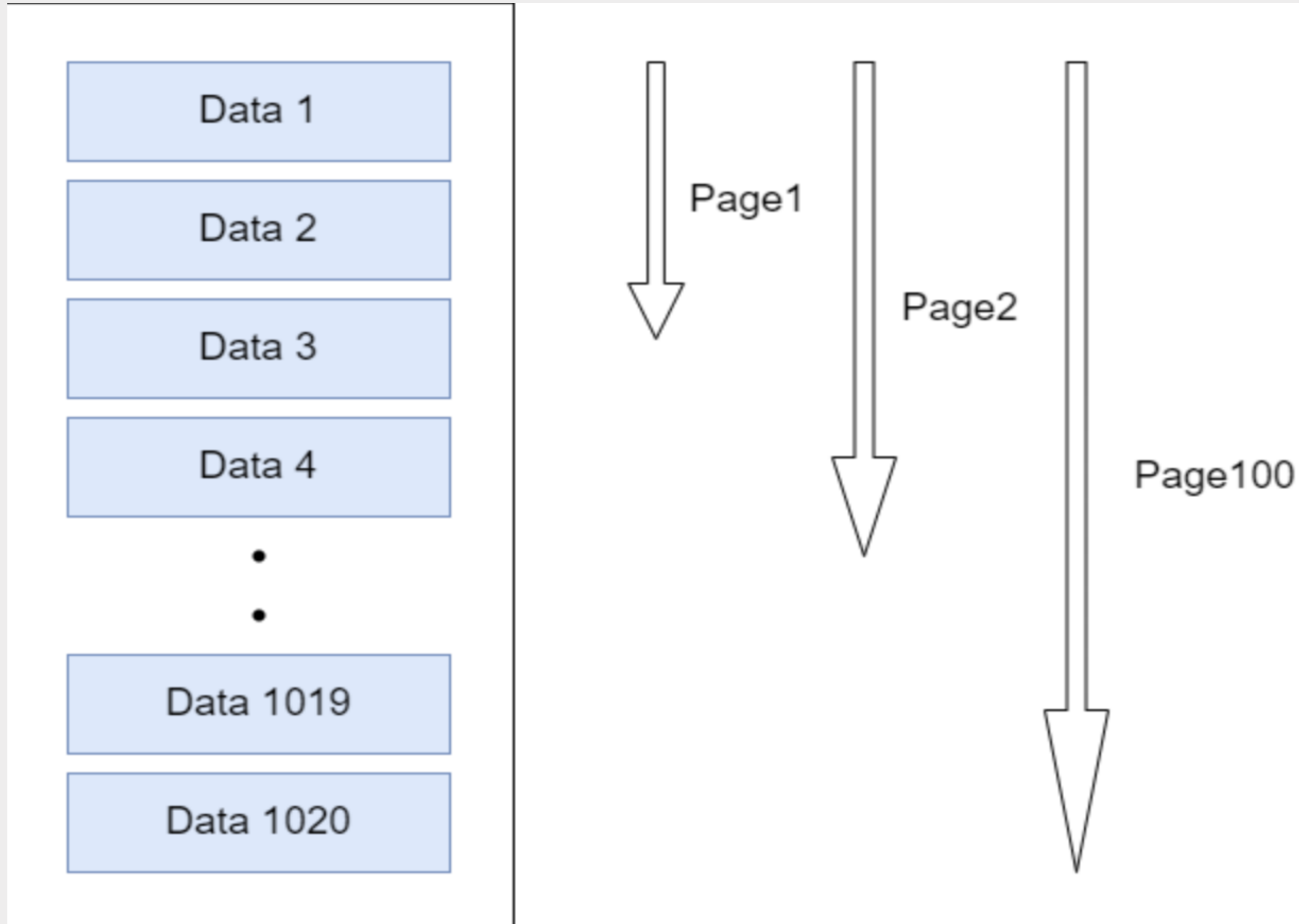
Offset

**Offset부터 limit
만큼만 조회가 될까?**



Offset 단점

Offset 단점





No - Offset

No - Offset

```
public Slice<Board> searchAllBySlice(Long lastBoardId, Pageable pageable)
{
    List<Board> results = query
        .selectFrom(QBoard.board)
        .where(ltBoardId(lastBoardId))
        .orderBy(QBoard.board.boardId.desc()) //최신글부터 보여지기
        .limit(pageable.getPageSize() + 1)
        .fetch();
}
```


No - Offset

```
boolean hasNext = results.size() > pageable.getPageSize();
    if (hasNext) {
        results.remove(pageable.getPageSize());
    }

    return new SliceImpl<>(results, pageable, hasNext);
}

private BooleanExpression ltBoardId(Long boardId) {
    if (boardId == null) {
        return null;
    }
    return QBoard.board.boardId.lt(boardId);
}
}
```