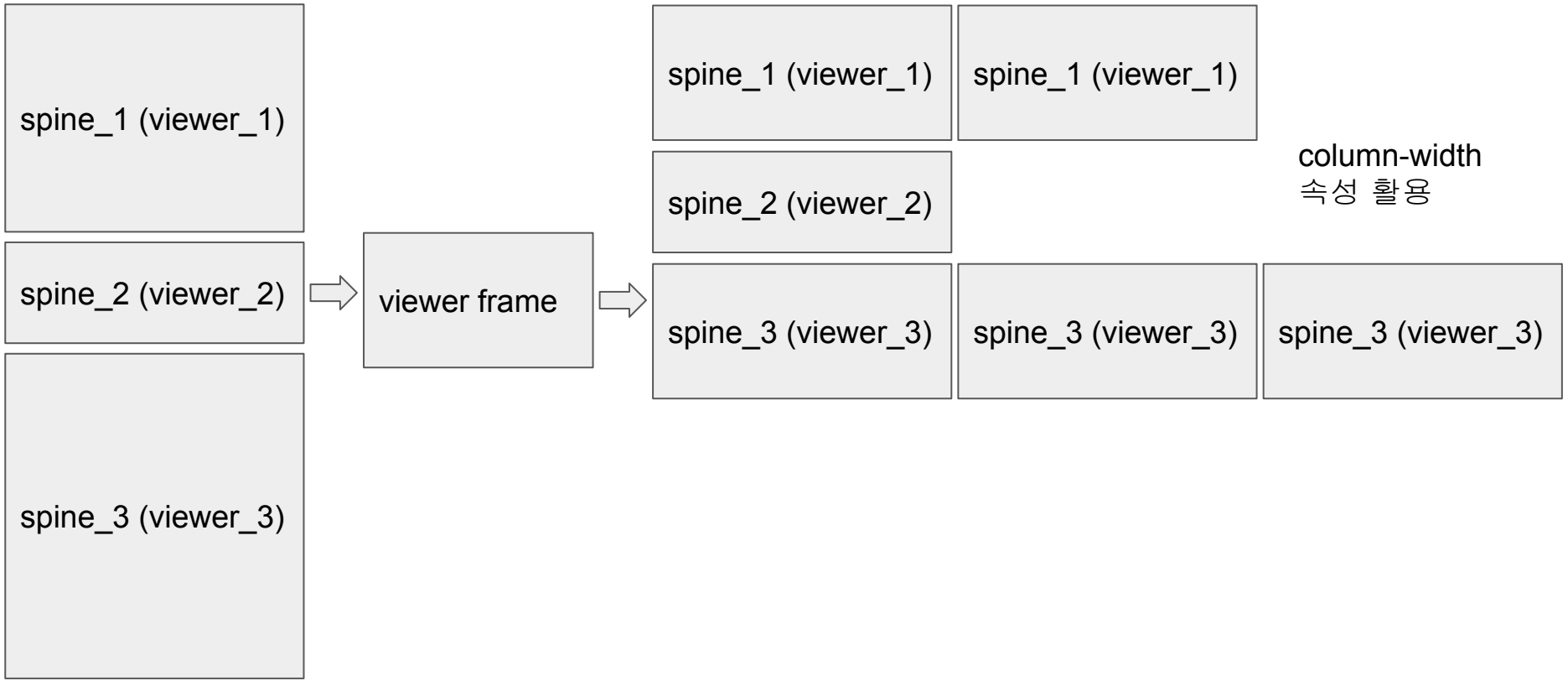
RIDI 입사자 과제
- 구현 중간 리뷰 (Viewer)

Viewer 를 어떻게 구성해야하나?

- .epub 파일을 @ridi/epub-parser 를 통하여 파싱
- public/{fileName} 으로 unZip
- parse 한 book의 spine 를 viewer(html 파일) 로 read,
- parse 한 book의 style 를 text file 로 구성

- spine 과 viewer 를 이용하여 Viewer 구성

Viewer 의 paging 처리 - 1



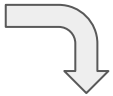
Viewer 의 paging 처리 - 2

spine_1 (viewer_1) | spine_1 (viewer_1)

spine_2 (viewer_2)

spine_3 (viewer_3) | spine_3 (viewer_3) | spine_3 (viewer_3)

column-width
속성 활용



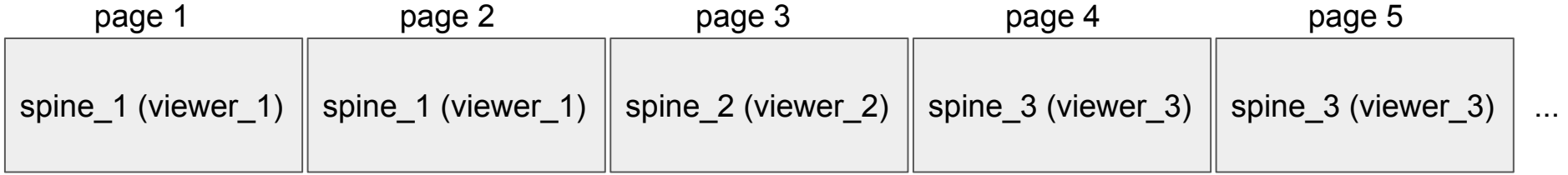
column-width 를 사용하여 viewer 한개를 가로로 펼쳐 놓고 각 viewer들을 가로 정렬 시킴

전체 스크롤 에서 viewer frame 에 width를 계산하여 전체 페이지 수를 구함

그 후 이동은 스크롤 이동을 통하여 페이징 처리

page 1	page 2	page 3	page 4	page 5	
spine_1 (viewer_1)	spine_1 (viewer_1)	spine_2 (viewer_2)	spine_3 (viewer_3)	spine_3 (viewer_3)	...

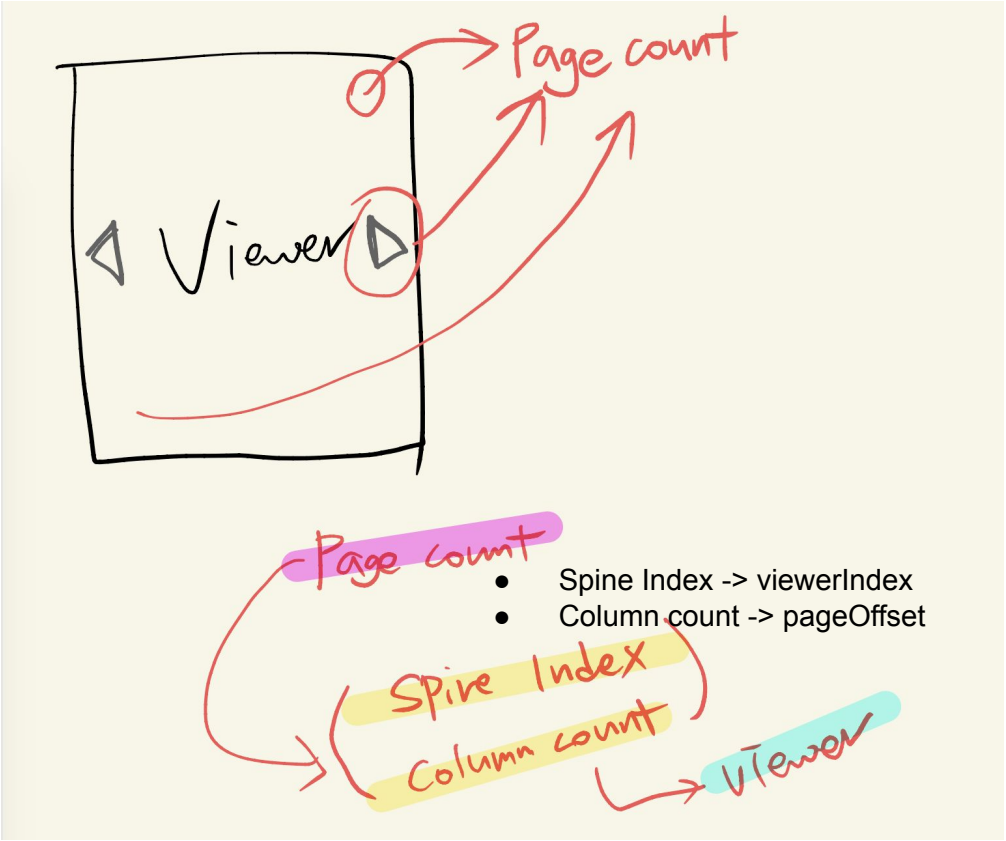
Viewer 의 paging 처리 - 3



데이터는 다음과 같이 저장되어있게 된다. (추후 필요한 데이터는 더 저장할 예정)

viewreCountList = {
    index: page1, page2, page3 와 같은 page index,
    count: 각 viewer에 column count (예: viewer_1: 2, viewer_2: 1 ...)
    spineID: 각 viewer에 spine id (목차로 이동 할 때 사용)
}

Viewer 의 paging 처리 - 4



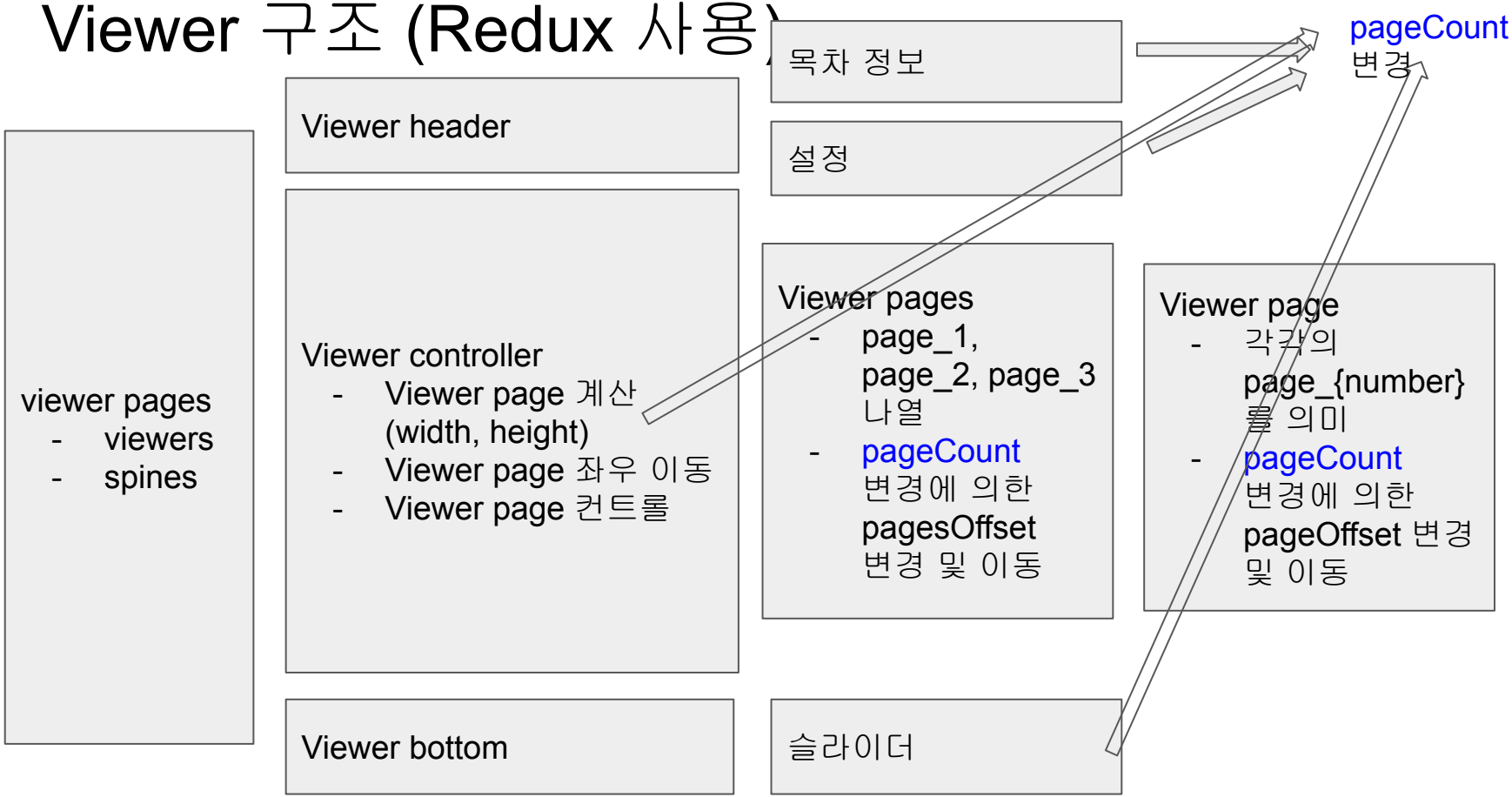
- Spine Index -> viewerIndex
- Column count -> pageOffset

pageCount 는 Viewer에 왼쪽, 오른쪽, 목차 이동, 스크롤 이동 으로 인하여 변경 가능

- pageCount 변경
- viewerCountList 로 부터 viewerIndex 검색
- viewer index 검색 후 column count(pageOffset) 검색

viewer index를 통하여 전체 스크롤 움직여 준 후, page offset을 통하여 각 viewer 에 내부 스크롤 이동 (column 이동)

Viewer 구조 (Redux 사용)



Viewer pageCount

- 목차
  - Epub 에서 주는 목차 정보를 바탕으로 목차 정보를 그린 후
  - 목차 정보 클릭시 클릭 된 spine id 를 바탕으로 그 전에 viewerCountList 안에 spine id를 검색
  - spine id 를 바탕으로 pageCount 계산
  - store 에 update
- 슬라이더
  - 슬라이더 변경되는 값에 따라 pageCount 적용

- pageCount 변화에 따라 pagesOffset, pageOffset 변경

Viewer setting

- 설정
  - 설정에서 글자 크기, 문단 너비, 줄 간격 변경
  - debounce 를 이용하여 연속 클릭시 제어
  - 스타일 변경시 보여지는 페이지에 한 해서 스타일 변경
  - 설정 모달 닫힐 시에 전체 페이지에 일괄 적용 후, 전체 페이지 수 다시 계산

추후

- 서재 Library 기능