

지금까지는 전체 목록을 갖고오고 스크롤 문제가 발생

어떻게 각 페이지로 나눌지...

페이지 네이션 혹은 무한 스크롤(데이터가 모지랄때마다 로드시키는형식)으로 해결하는데
우리는 전자로 해결해보자

KB금융그룹

국민은행
금융파트너

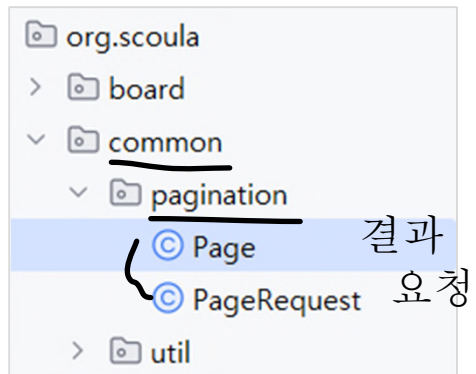
2025년 상반기 K-디지털 트레이닝

게시판 – 페이지네이션 (백엔드)

[KB] IT's Your Life

1 게시판 – 페이지네이션(백엔드)

✓ DTO 객체



PageRequest.java

```
package org.scoula.common.pagination;  
...
```

```
@Data
```

```
@AllArgsConstructor(access = AccessLevel.PRIVATE) ✓✓ 이 생성자는 private으로 만들어 달라
```

```
public class PageRequest {
```

```
    private int page; // 요청 페이지
```

```
    private int amount; // 한 페이지 당 데이터 건수
```

select ... limit @@@ 에 들어가겠따

```
    public PageRequest() {  
        page = 1;  
        amount = 10;  
    }
```

```
    public static PageRequest of(int page, int amount) {  
        return new PageRequest(page, amount);  
    }
```

AllArgs 생성자 활용

```
    public int getOffset() { // offset 값 계산  
        return (page - 1) * amount;  
    }
```

게터. offset이라는 프로퍼티 정의함. 나중에 매퍼에서 #{ offset } 가능

Page.java

```
package org.scoula.common.pagination;
...
@Data
@NoArgsConstructor
@AllArgsConstructor(access = AccessLevel.PRIVATE)
public class Page<T> {
    private int totalCount; // 전체 데이터 건수
    private int totalPages; // 전체 페이지 수
    @JsonIgnore 질렬화할때 빼라
    private PageRequest pageRequest;
    private List<T> list; // 데이터 목록

    public static <T> Page of(PageRequest pageRequest, int totalCount, List<T> list) {
        int totalPages = (int) Math.ceil((double) totalCount / pageRequest.getAmount()); // 전체 페이지 수 계산
        return new Page(totalCount, totalPages, pageRequest, list);
    }

    public int getPageNum() { return pageRequest.getPage(); }
    public int getAmount() { return pageRequest.getAmount(); }
}
```

getter/ property는 pageNum와 amount

jackson이 json을 구성할 때 뭘 보고 구성할까. 필드멤버? ㄴㄴ private 못 읽어.
property를 읽는다. 즉 getter를 읽는 getter을 보고 이를 json의 키로 구성한다

BoardMapper.java

```
package org.scoula.board.mapper;

import org.scoula.board.domain.BoardAttachmentVO;
import org.scoula.board.domain.BoardVO;
import org.scoula.common.pagination.PageRequest;

import java.util.List;

public interface BoardMapper {
    int getTotalCount(); ✓
    List<BoardVO> getPage(PageRequest pageRequest); ✓
    ...
}
```

BoardMapper.xml

```
...  
<select id="getTotalCount" resultType="java.lang.Integer">  
    select count(*) from tbl_board  
</select>  
  
<select id="getPage" resultType="BoardVO">  
    select * from tbl_board order by no desc  
    limit #{offset}, #{amount}  
</select>  
...
```

(page-1) X amount 값이 와야함
해당 수식 자체가 올 순 없다. limit절엔.
offset이라는 프로퍼티를 구해놔야함.
아까 만든 offset프로퍼티 사용

BoardService.java

```
package org.scoula.board.service;

import org.scoula.board.domain.BoardAttachmentVO;
import org.scoula.board.dto.BoardDTO;
import org.scoula.common.pagination.Page;
import org.scoula.common.pagination.PageRequest;

import java.util.List;

public interface BoardService {
    Page<BoardDTO> getPage(PageRequest pageRequest);
    ...
}
```

BoardServiceImpl.java

```
public class BoardServiceImpl implements BoardService {  
    private final static String BASE_DIR = "c:/upload/board";  
  
    private final BoardMapper mapper;
```

@Override

```
public Page<BoardDTO> getPage(PageRequest pageRequest) {
```

```
    List<BoardVO> boards = mapper.getPage(pageRequest); ✓
```

```
    int totalCount = mapper.getTotalCount(); ✓
```

```
    return Page.of(pageRequest, totalCount,  
        boards.stream().map(BoardDTO::of).toList());
```

vo객체를 DTO바꿔서
page객체 만들어 생성후 반환

```
}
```

```
...
```

```
}
```


✏ BoardController.java ✓

```
public class BoardController {
```

```
    private final BoardService service;
```

기존 getlist수정!

```
    @GetMapping("")
```

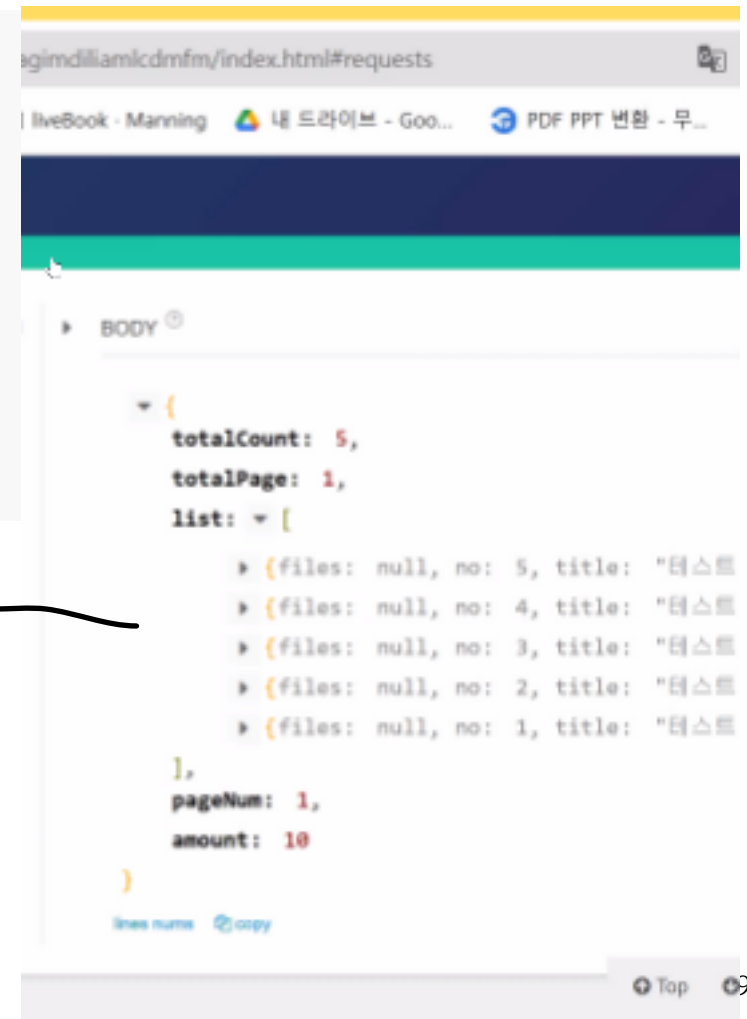
```
    public ResponseEntity<Page> getList(PageRequest pageRequest) {  
        return ResponseEntity.ok(service.getPage(pageRequest));  
    }
```

```
    ...
```

```
}
```

talend로 확인

해당정보들로
한
페이지 구성하면 돼!



테스트용 데이터가 몇백개가 필요하다면 테스트 만들어서
반복분 돌려라 넣는걸. 위래는 매퍼 서비스 컨트롤러에 기능이 추가
될때마다 단위테스트를 해야한다.

2025년 상반기 K-디지털 트레이닝

게시판 – 페이지네이션 (프론트엔드)

[KB] IT's Your Life

✓ 페이지네이션

쿼리스트링에
몇번 페이지와 한페이지당 몇개 보여줄지 지정


<http://localhost:5173/board/list?page=4&amount=10>

 **게시글 목록**

(총 120건)

No	제목	작성자	작성일
97	페이징 테스트 80	user0	2024-07-31
96	페이징 테스트 79	user0	2024-07-31
95	페이징 테스트 78	user0	2024-07-31
94	페이징 테스트 77	user0	2024-07-31
93	페이징 테스트 76	user0	2024-07-31
92	페이징 테스트 75	user0	2024-07-31
91	페이징 테스트 74	user0	2024-07-31
90	페이징 테스트 73	user0	2024-07-31
89	페이징 테스트 72	user0	2024-07-31
88	페이징 테스트 71	user0	2024-07-31



  1 2 3 **4** 5 6 ... 12  

 글 작성

이것도 만들어야함. bootstrap pagination 활용해도됨
라이브러리도 활용가능. ==>

게시판 – 페이지네이션 (프론트엔드)

✓ 페이지네이션 라이브러리

- npm i vue-awesome-paginate



main.js

```
import './assets/main.css';
import 'bootstrap/dist/css/bootstrap.css';
import 'vue-awesome-paginate/dist/style.css'; ✓

import { createApp } from 'vue';
import { createPinia } from 'pinia';
import VueAwesomePaginate from 'vue-awesome-paginate'; ✓

import App from './App.vue';
import router from './router';

const app = createApp(App);

app.use(VueAwesomePaginate); ✓
app.use(createPinia());
app.use(router);

app.mount('#app');
```

✓ 기본 사용방법

```
<vue-awesome-paginate
  :total-items="전체 데이터 건수"
  :items-per-page="한 페이지당 데이터 건수"
  :max-pages-shown="제시할 페이지 개수"
  :show-ending-buttons="처음과 마지막 페이지 이동 출력 여부"
  v-model="현재 페이지"
  @click="페이지 번호 클릭 핸들러"
/>
```

✓ 페이지네이션 스타일

○ 기본 스타일



○ 커스텀마이징 스타일



assets/main.css

```
...
/* 페이지 네이션 */
.pagination-container {
  display: flex;
  /* column-gap: 10px; */
}

.paginate-buttons {
  height: 40px;
  width: 40px;
  /* border-radius: 20px; */
  cursor: pointer;
  background-color: white; /*rgb(242, 242, 242);*/
  border: 1px solid white; /* rgb(217, 217, 217);*/
  color: black;
}

.paginate-buttons:hover {
  border-radius: 20px;
  background-color: #d8d8d8;
}
```


assets/main.css

```
.active-page {  
  border-radius: 20px;  
  background-color: #3498db;  
  border: 1px solid #3498db;  
  color: white;  
}  
  
.active-page:hover {  
  background-color: #2988c8;  
}
```

게시판 – 페이지네이션 (프론트엔드)

✓ 페이지네이션 ~

○ 쿼리 운영

- 서버의 PageRequest 객체와 동일하게 운영

pagerequestDTO에 맞춰가지고
쿼리스트링을 운영하면 돼

○ 페이지 응답 ✓

- 서버의 Page<T>와 동일하게 운영

○ 실제 페이지 로드하는 시점 !! ✓

- 마운트될 때 : 첫 번째 페이지 로드 쿼리가 없으니 디폴트로
- 페이지네이션 페이지 번호 클릭시: 라우트로 쿼리 재구성후 push
- 뒤로가기, 앞으로가기 클릭시: 현재 라우트의 쿼리 변화 감지하여 해당 페이지 로드

○ 글작성, 상세보기(수정, 삭제 포함)에 쿼리 전달

- 목록으로 돌아가기 할 때 쿼리 정보를 복원해야 해당 페이지로 정확하게 이동
- 쿼리 정보가 없다면 목록으로 돌아가기 할 때 항상 첫 페이지로 이동하게 됨

BoardListPage.vue

```
<script setup>
import api from '@api/boardApi';
import { ref, reactive, computed, watch } from 'vue';
import moment from 'moment';
import { useRoute, useRouter } from 'vue-router';

const cr = useRoute();
const router = useRouter();

const page = ref({});

const articles = computed(() => page.value.list);

const pageRequest = reactive({
  page: parseInt(cr.query.page) || 1,
  amount: parseInt(cr.query.amount) || 10,
});
```

BoardListPage.vue

// 페이지네이션 페이지 변경

```
const handlePageChange = async (pageNum) => {
```

// url 변경 --> query 파트만 변경되므로 컴포넌트가 다시 마운트되지 않음

// watch를 통해 cr이 변경됨을 감지하여 페이지 로드해야 함

```
{  
  router.push({  
    query: { page: pageNum, amount: pageRequest.amount },  
  });  
};
```

// pageRequest의 값 변경된 경우 호출

```
watch(cr, async (newValue) => { CR변경되면 호출됨
```

```
  console.log('WATCH', cr.query.page);
```

```
  pageRequest.page = parseInt(cr.query.page);
```

```
  pageRequest.amount = parseInt(cr.query.amount);
```

```
  await load(pageRequest);
```

```
});
```

url 변경 → 쿼리 파트만 변경 → 마운트 → 화면
✓ board/list? page = x & amount = x

currentRouter의 파라미터가 바뀜

BoardListPage.vue

```
const load = async (query) => {  
  try {  
    page.value = await api.getList(query);  
    console.log(page.value);  
  } catch {}  
};  
  
load(pageRequest);  
</script>
```

BoardListPage.vue

```
<template>
  <div>
    <h1 class="mb-3"><i class="fa-solid fa-paste"></i> 게시글 목록</h1>

    <div class="mt-5 text-end">(총 {{ page.totalCount }}건)</div>
    <table class="table table-striped">
      <thead>
        <tr>...</tr>
      </thead>
      <tbody>
        <tr v-for="article in articles" :key="article.no">
          <td>{{ article.no }}</td>
          <td>
            <router-link :to="{ name: 'board/detail', params: { no: article.no }, query: cr.query }">
              {{ article.title }}
            </router-link>
          </td>
          <td>{{ article.writer }}</td>
          <td>{{ moment(article.regDate).format('YYYY-MM-DD') }}</td>
        </tr>
      </tbody>
    </table>
```

BoardListPage.vue

```
<div class="my-5 d-flex">
  <div class="flex-grow-1 text-center">
    <!-- 페이지 네이션 -->
    <vue-awesome-paginate
      :total-items="page.totalCount"
      :items-per-page="pageRequest.amount"
      :max-pages-shown="5"
      :show-ending-buttons="true"
      v-model="pageRequest.page"
      @click="handlePageChange"
    >
      <template #first-page-button><i class="fa-solid fa-backward-fast"></i></template>
      <template #prev-button><i class="fa-solid fa-caret-left"></i></template>
      <template #next-button><i class="fa-solid fa-caret-right"></i></template>
      <template #last-page-button><i class="fa-solid fa-forward-fast"></i></template>
    </vue-awesome-paginate>
  </div>
```

css activate배정

named slot

BoardListPage.vue

```
<div>
  <router-link :to="{ name: 'board/create', query: cr.query }" class="btn btn-primary">
    <i class="fa-solid fa-pen-to-square"> </i> 글 작성 </router-link>
  </div>
</div>
</div>
</template>
```


BoardCreatePage.vue

```
<script setup>
...
import { useRoute, useRouter } from 'vue-router';

const cr = useRoute();
...
</script>

<template>
  <h1> <i class="fa-regular fa-pen-to-square"> </i> 글 작성 </h1>

  <form @submit.prevent="submit">
    ...
    <div class="my-5 text-center">
      ...
      <router-link class="btn btn-primary" :to="{ name: 'board/list', query: cr.query }">
        <i class="fa-solid fa-list"> </i>
        목록
      </router-link>
    </div>
  </form>
</template>
```

목록으로 돌아가기 할때만
쿼리파트를 복원해주면 됨

BoardDetailPage.vue

```
<script setup>
...
const cr = useRoute();
const router = useRouter();
...

const back = () => {
  router.push({ name: 'board/list', query: cr.query });
};

const update = () => {
  router.push({ name: 'board/update', params: { no: no }, query: cr.query });
};
...
const remove = async () => {
  if (!confirm('삭제할까요?')) return;
  await api.delete(no);
  router.push({ name: 'board/list', query: cr.query });
};
...
</script>
...
```

BoardUpdatePage.vue

```
<script setup>
...
const cr = useRoute();
const router = useRouter();
...

const back = () => {
  router.push({ name: 'board/detail', params: { no }, query: cr.query });
};
...

const submit = async () => {
  if (!confirm('수정할까요?')) return;

  if (files.value.files.length > 0) {
    article.files = files.value.files;
  }

  await boardApi.update(article);
  router.push({ name: 'board/detail', params: { no }, query: cr.query });
};
...
```