

2025년 상반기 K-디지털 트레이닝

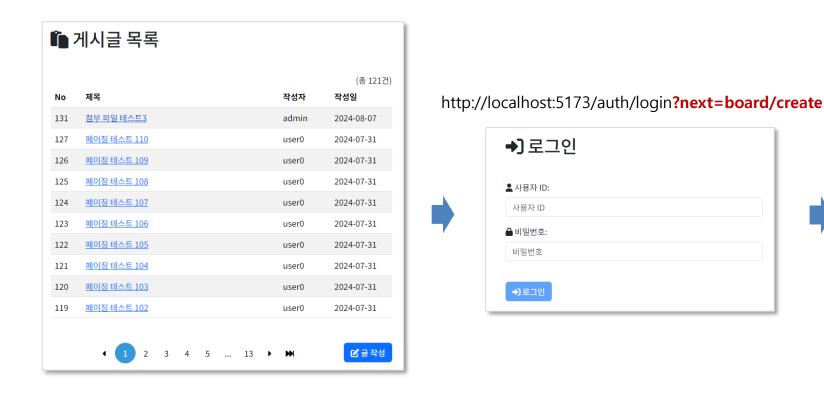
인증 가드

[KB] IT's Your Life



인증 가드

☑ 로그인하지 않은 상태에서 글쓰기를 누른 경우 다음과 같이 페이지가 이동하도록 인증 가드를 설정하세요.



제목			
첨부파일 파일 선택	선택된 파일 없음		
내용			
내용			

○ 라우팅 가드 처리

인증 가드

util/guards.js

```
import { useAuthStore } from '@/stores/auth';

export const isAuthenticated = (to, from) => {
  const auth = useAuthStore();

  if (!auth.isLogin) {
    console.log('로그인 필요.....');
    return { name: 'login', query: { next: to.name } };
  }
  console.log('로그인 인증');
};
```

router/board.js

```
import { isAuthenticated } from '@/util/guards';
export default [
    path: '/board/create',
   name: 'board/create',
    component: () => import('../pages/board/BoardCreatePage.vue'),
    beforeEnter: isAuthenticated,
 },
   path: '/board/update/:no',
   name: 'board/update',
    component: () => import('../pages/board/BoardUpdatePage.vue'),
    beforeEnter: isAuthenticated,
 },
```

pages/auth/LoginPage.vue

```
<script setup>
import { computed, reactive, ref } from 'vue';
import { useAuthStore } from '@/stores/auth';
import { useRoute, useRouter } from 'vue-router';
const cr = useRoute();
const router = useRouter();
const auth = useAuthStore();
const login = async () => {
 try {
   await auth.login(member);
   if (cr.query.next) { // 로그인 후 이동할 페이지가 있는 경우
      router.push({ name: cr.query.next });
   } else { // 일반 로그인
     router.push('/');
 } catch (e) {
   error.value = e.response.data;
</script>
```



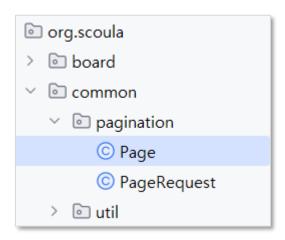
2025년 상반기 K-디지털 트레이닝

페이지네이션(백엔드)

[KB] IT's Your Life



- ☑ 페이지네이션 DTO 객체를 만드시오.
 - PageRequest.java: 페이지네이션 요청 저장
 - Page.java: 페이지네이션 결과 저장



PageRequest.java

```
package org.scoula.common.pagination;
@Data
@AllArgsConstructor(access = AccessLevel.PRIVATE)
public class PageRequest {
    private int page; // 요청 페이지
   private int amount; // 한 페이지 당 데이터 건수
    public PageRequest() {
       page = 1;
       amount = 10;
    public static PageRequest of(int page, int amount) {
       return new PageRequest(page, amount);
    public int getOffset() { // offset 값 계산
       return (page - 1) * amount;
```

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 totalPage; // 전체 페이지 수
   @JsonIgnore
   private PageRequest pageRequest;
   private List<T> list; // 데이터 목록
   public static <T> Page of(PageRequest pageRequest, int totalCount, List<T> list) {
       int totalPage = (int)Math.ceil((double)totalCount / pageRequest.getAmount()); // 전체 페이지 수
계산
       return new Page(totalCount, totalPage, pageReguest, list);
    public int getPageNum() { return pageRequest.getPage(); }
    public int getAmount() { return pageRequest.getAmount(); }
```

- ☑ 페이지네이션처리를 위한 Mapper 인터페이스와 XML을 정의하세요.
 - ㅇ 추가 메서드
 - int getTotalCount();
 - List<BoardVO> getPage(PageRequest pageRequest);
- 💟 서비스 계층에 페이지네이션 처리 메서드를 추가하세요.
 - Page<BoardDTO> getPage(PageRequest pageRequest);

BoardController의 기존 목록보기 기능을 페이지네이션으로 처리되도록 수정하세요.

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

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());
```

BoardController.java

```
public class BoardController {
    private final BoardService service;

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