

2024년 상반기 K-디지털 트레이닝

ScoulaTodo – Pagination

[KB] IT's Your Life

- ✓ 페이지네이션 테스트 데이터 구축
 - 단위 테스트를 이용해서 163건의 테스트 데이터 생성
 - 소유자는 "guest"

TodoApp.java

```
package org.scoula.todo.dao;
...

class TodoDaoTest {
    TodoDao dao = new TodoDaoImpl();

    @AfterAll
    static void tearDown() {
        JDBCUtil.close();
    }

    @Test
    @DisplayName("테스트 데이터를 생성한다.")
    public void testData() throws SQLException {
        for(int i = 0; i <= 163; i++) {
            TodoVO todo = TodoVO.builder()
                .title("테스트 Todo " + i)
                .description("테스트 Todo 설명" + i)
                .userId("guest")
                .done(false)
                .build();
            dao.create(todo);
        }
    }
}
```

✓ 페이지네이션

○ 적용 대상

- 목록 보기
- 검색

○ 페이지네이션을 위한 기본 정보

- 한 페이지 당 데이터 건수 → limit 값
- 현재 페이지 → offset 계산

PageRequest.java

```
package org.scoula.todo.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class PageRequest {
    private int page; // 요청 페이지
    private int size; // 한 페이지 당 데이터 건수

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

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

TodoDao.java

```
package org.scoula.todo.dao;
...

public interface TodoDao {
    ...

    // 페이지네이션
    List<TodoVO> getPage(String userId, PageRequest request) throws SQLException;
}
```

TodoDaoImpl.java

```
@Override
public List<TodoVO> getPage(String userId, PageRequest request) throws SQLException {
    String sql = "select * from todo where userId = ? limit ?, ?";

    try (PreparedStatement stmt = conn.prepareStatement(sql)){
        stmt.setString(1, userId);
        stmt.setInt(2, request.getOffset());
        stmt.setInt(3, request.getSize());

        try(ResultSet rs = stmt.executeQuery()) {
            return mapList(rs);
        }
    }
}
```

✓ 페이지네이션 결과 구성하기

○ 기본 구성 내용

- 전체 데이터 건수
- 전체 페이지 수
- 현재 페이지 번호
- 현재 페이지의 데이터 목록

Page.java

```
package org.scoula.todo.dto;
...

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Page {
    private int totalCount; // 전체 데이터 건수
    private int totalPages; // 전체 페이지 수
    private int page;       // 현재 페이지
    private int pageSize;   // 한 페이지당 데이터 건수
    private List<TodoV0> list; // 데이터 목록

    public static Page of(PageRequest request, int totalCount, List<TodoV0> list) {
        int totalPages = (int) Math.ceil((double) totalCount / request.getSize()); // 전체 페이지 수 계산
        int page = request.getPage();
        int pageSize = request.getSize();

        return new Page(totalCount, totalPages, page, pageSize, list);
    }
}
```

☑ 페이지 출력

총 건수: 103

=====

135] 테스트 Todo 27

136] 테스트 Todo 28

137] 테스트 Todo 29

138] 테스트 Todo 30

139] 테스트 Todo 31

140] 테스트 Todo 32

141] 테스트 Todo 33

142] 테스트 Todo 34

143] 테스트 Todo 35

144] 테스트 Todo 36

=====

페이지(4/11): 1 2 3 [4] 5 6 7 8 9 10 11

페이지 번호:

TodoService.java

```
private Page getPage(int pageNum) throws SQLException {
    int count = dao.getTotalCount(userId.get());
    PageRequest request = PageRequest.of(pageNum, 10);

    List<TodoV0> list = dao.getPage(userId.get(), request);

    return Page.of(request, count, list);
}

private void printPageData(Page page, int pageNum) {
    System.out.println("총 건수: " + page.getTotalCount());
    System.out.println("=====");
    for (TodoV0 todo : page.getList()) {
        System.out.printf("%03d] %s%s\n", todo.getId(), todo.getTitle(),
            todo.getDone() ? "(완료)" : "");
    }
    System.out.println("=====");
    System.out.printf("페이지(%d/%d): ", pageNum, page.getTotalPage());
    for (int i = 1; i <= page.getTotalPage(); i++) {
        System.out.printf("%s ", i == pageNum ? "[" + i + "]" : "" + i);
    }
    System.out.println();
}
```

TodoService.java

```
public void printPage() {  
    int pageNum = 1;  
    try {  
        do {  
            Page page = getPage(pageNum);  
            printPageData(page, pageNum);  
            pageNum = Input.getInt("페이지 번호: ");  
        } while(pageNum != -1);  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

TodoApp.java

```
@Override
public void init() {
    anonymousMenu = new Menu();
    anonymousMenu.add(new MenuItem("로그인", this::login));
    anonymousMenu.add(new MenuItem("가입", accountService::join));
    anonymousMenu.add(new MenuItem("종료", this::exit));

    userMenu = new Menu();
    //    userMenu.add(new MenuItem("목록", todoService::print));
    userMenu.add(new MenuItem("목록", todoService::printPage));
    userMenu.add(new MenuItem("검색", todoService::search));
    ...
}
```