

JPA - Pageable(페이징)

프로젝트 : B21dJPA_Pageable

준비사항

- 의존설정 : JDBC API, Lombok, Oracle Driver, Spring Web, Spring Data JPA
- JSP 사용을 위한 설정을 한다.
- Refresh Gradle Project 를 눌러 적용한다.

Pageable

- Spring Data JPA에서 페이징 처리를 위한 인터페이스
- 대량의 데이터를 한 번에 모두 가져오지 않고, 페이지 크기만큼 데이터를 분할해서 조회
- 이를 통해 메모리 사용량을 줄이고 성능을 최적화

주요 정보

- 페이지 번호 (Page Number)
 - 현재 가져올 페이지가 몇 번째 페이지인지를 나타냄
 - 페이지 번호는 0부터 시작
- 페이지 크기 (Page Size)
 - 한 페이지에 몇 개의 데이터를 가져올지를 설정
- 정렬 정보 (Sort)
 - 페이징된 데이터를 특정 컬럼을 기준으로 정렬
 - Sort 객체를 사용하여 오름차순 또는 내림차순으로 정렬
 - 여러 필드에 대해 정렬 기준을 설정할 수도 있음

프로퍼티 : /B21dJPA_Pageable/src/main/resources/application.properties

이전 예제와 동일함.

spring.jpa.properties.hibernate.hbm2ddl.auto 항목만 적절히 설정

엔티티 : /B21dJPA_Pageable/src/main/java/com/edu/springboot/jpa/Member.java

```
1 package com.edu.springboot.jpa;
2
3*import jakarta.persistence.Entity;
12
13 @Getter
14 @AllArgsConstructor
15 @NoArgsConstructor(access = AccessLevel.PROTECTED)
16 @Builder
17 @Entity(name="JPAMEMBER03")
18 public class Member {
19     @Id
20     @SequenceGenerator (
21         name = "mySequence03",
22         sequenceName = "JPAMEMBER03_SEQ",
23         initialValue = 1,
24         allocationSize = 1
25     )
26     @GeneratedValue (generator = "mySequence03")
27     private Long id;
28     private String name;
29     private String email;
30 }
```

리포지토리 : com/edu/springboot/jpa/MemberRepository.java

```
1 package com.edu.springboot.jpa;
2
3*import org.springframework.data.domain.Page;
7
8 @Repository
9 public interface MemberRepository extends JpaRepository<Member, Long>
10 {
11     //List<Member> findByNameLike(String keyword, Pageable pageable);
12     Page<Member> findByNameLike(String keyword, Pageable pageable);
13 }
14
```

서비스 : /B21dJPA_Pageable/src/main/java/com/edu/springboot/jpa/MemberService.java

```
1 package com.edu.springboot.jpa;
2
3*import org.springframework.beans.factory.annotation.Autowired;
4
5
6
7
8 @Service
9 public class MemberService {
10
11     @Autowired
12     private MemberRepository memberRepository;
13
14     public Page<Member> selectNameLike(String search,
15         Pageable pageable){
16         Page<Member> member = memberRepository
17             .findByNameLike(search, pageable);
18         return member;
19     }
20 }
21
```

컨트롤러 : /B21dJPA_Pageable/src/main/java/com/edu/springboot/MainController.java

```
1 package com.edu.springboot;
2
3*import java.util.List;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 @Controller
20 public class MainController {
21
22     @Autowired
23     MemberService memberService;
24
25     @GetMapping("/")
26     public String main() {
27         return "main";
28     }
29
30     @GetMapping("/selectByNameLike.do")
31     public String selectByNameLike(
32         @RequestParam("name") String pname,
33         @RequestParam("page") String page,
34         Model model) {
35         System.out.println("selectMembers3(검색어) : "+ pname );
36         System.out.println("selectMembers3(페이지) : "+ page );
37
38         String name = pname + "%";
39         Sort sort = Sort.by(Sort.Order.desc("id"));
40         int pageNum = Integer.parseInt(page) - 1;
41
42     }
43 }
44
```

```

42     Pageable pageable = PageRequest.ofSize(5).withPage(pageNum)
43         .withSort(sort);
44     Page<Member> result = memberService
45         .selectNameLike(name, pageable);
46     List<Member> content = result.getContent();
47     long totalElements = result.getTotalElements();
48     int totalPages = result.getTotalPages();
49     int size = result.getSize();
50     int pageNumber = result.getNumber() + 1;
51     int numberOfElements = result.getNumberOfElements();
52
53     model.addAttribute("members", content);
54     model.addAttribute("totalElements", totalElements);
55     model.addAttribute("totalPages", totalPages);
56     model.addAttribute("size", size);
57     model.addAttribute("pageNumber", pageNumber);
58     model.addAttribute("numberOfElements", numberOfElements);
59
60     return "member_list";
61 }
62 }
63

```

뷰 : /B21dJPA_Pageable/src/main/webapp/WEB-INF/views/main.jsp

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ taglib uri="jakarta.tags.core" prefix="c" %>
4  <!DOCTYPE html>
5  <html>
6  <head>
7  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8  <title>Insert title here</title>
9  </head>
10 <body>
11     <h2>Pageable(페이징적용)</h2>
12     <ul>
13         <li><a href=/selectByNameLike.do?name=test&page=1>
14             Name Like 조회(1page)</a></li>
15         <li><a href=/selectByNameLike.do?name=test&page=2>
16             Name Like 조회(2page)</a></li>
17         <li><a href=/selectByNameLike.do?name=test&page=3>
18             Name Like 조회(3page)</a></li>
19         <li><a href=/selectByNameLike.do?name=test&page=4>
20             Name Like 조회(4page)</a></li>
21     </ul>
22 </body>
23 </body>
24 </html>

```

뷰 : /B21dJPA_Pageable/src/main/webapp/WEB-INF/views/member_list.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib uri="jakarta.tags.core" prefix="c" %>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8 <title>Insert title here</title>
9 </head>
10 <body>
11     <h2>Spring boot 프로젝트</h2>
12     <ul>
13         <li><a href="/">루트</a></li>
14     </ul>
15
16     <h2>Pageable(페이징적용)</h2>
17     총 레코드 갯수 : ${totalElements} <br>
18     전체 페이지 수 : ${totalPages} <br>
19     페이지 당 레코드 수 : ${size} <br>
20     페이지번호 : ${pageNumber} <br>
21     엘리먼트(컨텐츠) 갯수 : ${numberOfElements} <br>
22     <hr />
23
24     <h3>레코드 출력</h3>
25     <c:forEach var="member" items="${members}">
26         <p>
27             아이디 : ${member.id},
28             이름 : ${member.name},
29             이메일 : ${member.email}
30         </p>
31     </c:forEach>
32 </body>
33 </html>
34
```