

## 0701

### 1. 새페이지 만들기

- kepcolms/edu/test 패키지 생성
- kepcolms/edu/test 에 testController 생성
- 같은 위치에 service, mapper, mapper.xml 생성 예) testService.java
- src/main/webapp/lms/edu 에 test 디렉토리 생성
- src/main/webapp/lms/edu/test 에 list.html 생성

### 2. 1페이지에 리스트 불러오기

url (RequestMapping) /lms/edu/test/list 로 하여 해당페이지에 lms\_edu\_contents\_play 목록 띄우기  
불러올 항목은 contents\_idx detail\_idx member\_id attempt

### 3. 2로 만든 목록에 클릭이벤트 생성하여 ajax 호출하는 스크립트(함수) 짜기

url은 lms/edu/test/listJson 필요한 파라미터는 contents\_idx

### 4. 3에서 contents\_idx 를 파라미터로 전달하여 다음내용 불러오기

3을 클릭하였을 경우 contents\_idx 를 전달해 하단에 lms\_edu\_event 와 lms\_edu\_event\_contents 목록 띄우기

-> lms\_edu\_event 에 lms\_edu\_event\_contents 를 event\_id 로 left join 해주세요

띄울 내용은 event\_idx contents\_idx event\_id event\_contents excellent\_cut good\_cuevaluation\_yn

## 0703

### 1. 이벤트 목록 불러오기(어제 만든것) /eventListJson

```
@RequestMapping(value = "eventlistJson")
```

```
@ResponseBody
```

```
public JsonResponse eventlistJson(@RequestEgovMap EgovMap req, Model model) {
```

```
    req.put("pageSize", "5");
```

```

        List<TestVo> list = testService.eventSelectList(req);

        model.addAttribute("list", list);

        model.addAttribute("req", req);


        return new JsonResponse.Builder().model(model).build();
    }

```

## 2. 이벤트 목록 페이징 /listPaging

```

@RequestMapping(value = "listPaging")

public String listPaging(@RequestEgovMap EgovMap req, Model model) throws Exception {

    String viewName = "lms/edu/result/paging";


    req.put("pageSize" , "5");

    List<TestVo> list = testService.eventSelectList(req);


    PageInfo<TestVo> pageInfo = new PageInfo<TestVo>(list, 3);


    model.addAttribute("pageInfo", pageInfo);

    model.addAttribute("req", req);

    model.addAttribute("list", list);


    return viewName;

}

```

## 3. paging.html 분리 (새로고침x, 재사용가능)

카톡첨부

#### 4. html 구조

이벤트 목록 구조 아래에 페이징 구조 추가(id명은 자유지만 html내에서 id값은 유니크해야합니다)

```
<div id="paginate_complex" th:if="${!#lists.isEmpty(list)}">
</div>
```

#### 5. 스크립트

(1) 이벤트 목록불러오는 ajax(어제만든) A함수

- A함수 파라미터에 pageNum 받는것 추가하고 상단에 if(pageNum == null) {pageNum = 1} 추가
- url 에도 pageNum=" + pageNum 추가
- 위의A함수 done 이후 마지막줄에 paging(pageNum); 추가

(2) 페이징 불러오는 ajax B함수

```
function paging(pageNum) {
    $.ajax({
        url: "/lms/edu/test/listPaging?pageNum=" + pageNum,
        type: 'GET',
        dataType: 'html',
    }).done(function (data) {
        $('#paginate_complex').empty();
        $('#paginate_complex').html(data);
    }).fail(function () {
        toast('네트워크 또는 시스템 장애입니다.', 4000);
    });
}
```

- page.html 에 페이지 클릭시 호출될 함수

```
function page(i) {
```

```
1번 A함수호출 A함수명(pageNum);  
}
```

위 내용대로 추가하고 이벤트목록 페이징이 적용되는지, 새로고침없이 내용만 바뀌는지 확인해보세요  
메모장으로 작성하고 테스트 안해봐서 오류날 수 있습니다

0704

#### 1. 이벤트 목록 불러오기(어제 만든것) /eventListJson

```
@RequestMapping(value = "eventlistJson")  
@ResponseBody  
public JsonResponse eventlistJson(@RequestEgovMap EgovMap req, Model model) {  
  
    req.put("pageSize" , "5");  
  
    List<TestVo> list = testService.eventSelectList(req);  
    model.addAttribute("list", list);  
    model.addAttribute("req", req);  
  
    return new JsonResponse.Builder().model(model).build();  
}
```

#### 2. 이벤트 목록 페이징 /listPaging

```
@RequestMapping(value = "listPaging")  
public String listPaging(@RequestEgovMap EgovMap req, Model model) throws Exception {  
  
    String viewName = "lms/edu/result/paging";
```

```

req.put("pageSize" , "5");

List<TestVo> list = testService.eventSelectList(req);

PageInfo<TestVo> pageInfo = new PageInfo<TestVo>(list, 3);

model.addAttribute("pageInfo", pageInfo);

model.addAttribute("req", req);

model.addAttribute("list", list);

return viewName;

}

```

### 3. paging.html 분리 (새로고침x, 재사용가능)

카톡첨부

### 4. html 구조

이벤트 목록 구조 아래에 페이징 구조 추가(id명은 자유지만 html내에서 id값은 유니크해야합니다)

```

<div id="paginate_complex" th:if="${!#lists.isEmpty(list)}">
</div>

```

### 5. 스크립트

(1) 이벤트 목록불러오는 ajax(어제만든) A함수

- A함수 파라미터에 pageNum 받는것 추가하고 상단에 if(pageNum == null) {pageNum = 1} 추가

- url 에도 pageNum=" + pageNum 추가

- list html body 어딘가에

```

<input type="hidden" name="contentsIdx" id="contentsIdx" value="1"/>   추가

```

수정

( - 위의 A함수 done 이후 마지막줄에 paging(pageNum);) 추가 X

-위의 A함수 done 이후 마지막줄에 추가

```
$("#contentsIdx").val(contentsIdx);
```

```
paging(contentsIdx, pageNum);
```

(2) 페이징 불러오는 ajax B함수

```
function paging(pageNum) {
```

```
    $.ajax({
```

```
        url: "/lms/edu/test/listPaging?pageNum=" + pageNum,
```

```
        type: 'GET',
```

```
        dataType: 'html',
```

```
    }).done(function (data) {
```

```
        $('#paginate_complex').empty();
```

```
        $('#paginate_complex').html(data);
```

```
    }).fail(function () {
```

```
        toast('네트워크 또는 시스템 장애입니다.', 4000);
```

```
    });
```

```
}
```

수정

- page.html 에 페이지 클릭시 호출될 함수

```
function page(i) { XXXX
```

```
    1번 A함수호출 A함수명(pageNum);
```

```
}
```

```
function page(i) {  
    var contentsIdx = $("#contentsIdx").val();  
    playclick(contentsIdx,i);  
}
```

위 내용대로 추가하고 이벤트목록 페이징이 적용되는지, 새로고침없이 내용만 바뀌는지 확인해보세요  
메모장으로 작성하고 테스트 안해봐서 오류날 수 있습니다

paging() 함수의 .html(data) 이 .append 혹은 .appendTo 대신에 사용되는데  
이게 어떤식으로 적용되는지 확인해보시면 좋습니다.

1. 마리아db 설치
2. sql gate 에서 db 백업파일 생성 (.sql)
3. 마리아db 커서 호스트생성, db생성, 권한부여 등등
4. 2에서 백업해둔 db 3에서 생성한 db에 넣기
5. application-db 로컬부분 url 로컬호스트로 수정
6. 작동확인

0715

(작업 중 대소문자 주의)

1.

사용자 페이지 메뉴 생성

관리자페이지 접속

/admin/cms/member/auth

게시판 관리 - 등록 (게시판 관리페이지를 따로 만들지 않을것이기 때문에 기존 페이지에서 등록)

사이트 - 초실감훈련센터

게시판명 - 테스트게시판

게시판 아이디 - test

콘텐츠 관리 - 메뉴관리

ROOT 클릭 후 하위메뉴 추가 클릭

메뉴명 - 테스트 게시판

메뉴종류 - 리디렉션

메뉴코드 - TESTBOARD(임의의 코드)

메뉴 주소 - /cms/test/board/post/list?site=kepco&mn=TEST&boardId=test (menu= 하위의 메뉴코드, boardId는 위에서만든 게시판 아이디 )

레이아웃 - 서브

메뉴 클릭시 이동될 페이지는 하위 페이지이므로 menu= 에 하위 코드 입력

생성된 테스트 게시판 클릭 후 하위메뉴 추가 클릭

메뉴명 - 테스트 게시판

메뉴종류 - 모듈

메뉴코드 - TEST(임의의 코드)

메뉴 주소 - /cms/test/board/post/list?site=kepco&mn=TEST&boardId=test (menu= 값은 메뉴코드 )



## 레이아웃 - 서브

2.

데이터베이스 테이블 작성

데이터베이스에 cms\_test\_post 라는 이름으로 테이블 생성, 내용은 cms\_board\_post 와 같을것

3.

컨트롤러 생성

kepcو.cms.test.board.post 패키지 생성

패키지 내용물은 아래와 같이 구성

TestPostController.java

TestPostService.java

TestPostVo.java

TestPostMapper.java

TestPostMapper.xml

기존 cms.board.post 를 복사 하여

이름을 TestPostController 와 같은식으로 변경하고

이후 작업은 해당 내용에서 필요한 내용을 제외하고 지우는 방식으로 진행

(이후 진행 중 오류가 뜬다면 이 부분에서 수정이 잘못되었을 가능성이 큼니다

오류 발생시 어느부분이 바뀌어야하는지 꼼꼼하게 체크하시길 바랍니다. 바뀌어야 하는 부분이 대개 핵심적인 부분입니다.)

4.

cms\_test\_post 목록 불러오는 메서드 작성 (selectList)

5.

TestPostController 작성

@Controller 밑의

@RequestMapping(value = "/cms/test/board/post") 해당부분 수정

public class TestPostController { } 내부에

list 메서드 빼고 나머지 전부 제거(view save delete 등등..)

list 메서드는 cms\_test\_post 테이블의 목록을 불러오는 것으로 짤 것

list 메서드에서 게시판 페이지 구성을 위해 어떤 요소들이 필요한지 확인해보세요

ex)키워드, 검색어, 페이징, 게시판정보, 권한, 설정, 스킨 등

5.

html 생성

cms/board/skin/basic/list.html 파일을

cms/test/post 폴더를 생성해서 그 안에 복사해 넣습니다

컨트롤러에서 뿌려주는 내용이 변경된 것이 없기 때문에 일단 기존 것 그대로 사용해도 에러가 없을 겁니다.

6.

사용자페이지 상단 메뉴 중 나의강의실 우편에 테스트 게시판이 생성되었는지 확인

클릭하여 정상적으로 게시판 목록페이지로 넘어가지는 지 확인

CUD 는 이 다음 예정....

오류가 난다면 콘솔창과 크롬개발자모드(F12) 를 참고하여 해결해보시고 5분이상 해결이 안된다면 시간 더 쓰지 마시고 불러주세요!

0716

CRUD 나머지

(스킨,메일 기능 주석처리로 제외할 것)

0. 관리자페이지 게시판관리 - 테스트게시판 - 수정

권한 전부다 체크

사용여부 - 첨부파일 사용 체크

부가옵션 - 페이지당 게시물수 15 , 목록의 페이지징갯수 10

1.

게시판 view페이지 만들기

컨트롤러

viewName 체크

testPostService.select 체크

PostVo -> TestPostVo 로 변경 (Vo명과 변수명의 대소문자 주의할 것)

boardPermissionService 에서 getBoardPermissionView 를 PostVo 대신 TestPostVo 를 받는 메소드로 복사

view.html 옮기기

ajax url 체크

2.

게시판 form페이지 만들기 (reply 는 답글)

컨트롤러

viewName 체크

PostVo -> TestPostVo 로 변경 (Vo명과 변수명의 대소문자 주의할 것)

boardPermissionService 에서 getBoardPermissionForm 를 PostVo 대신 TestPostVo 를 받는 메소드로 복사

html

form.html 옮기기

ajax url 체크

3.

save , file save, delete 등등 나머지

컨트롤러

@Value("\${global.file.base-path}")

private Path basePath;

체크

global.file.base-path 는 application-local 에서 자신의 실제 data 폴더 위치와 맞출 것

testPostService.insertWrite

testPostService.updateModify

testPostService.insertReply

각각 체크

4.

CRUD 기능 점검

5.

번외로 0에서 첨부파일 사용 해제시 게시글 저장 후 화면이 나가지 않는 오류가 있었습니다.

문제원인을 파악하고, 정상작동하도록 수정해보세요

=> 첨부파일 체크시에만 #fileDropzone을 주는데 체크 안해도 요구함.

=>

문제: id = fileDropzone

왜? Dropzone.js는 id나 class를 통해 드롭존 영역을 찾고 초기화한다. id="fileDropzone"가 지정된 요소가 HTML 문서에 존재하지 않으면 Dropzone.js는 해당 요소를 찾을 수 없어 초기화가 실패합니다. 이로 인해 파일 업로드 기능이 작동하지 않거나 페이지에서 오류가 발생할 수 있습니다. 따라서 Dropzone을 사용할 때는 해당 id를 가진 요소가 HTML 문서에 반드시 존재해야 합니다.

0722

목표 - 벤치마킹, 오픈소스 사용

(\*)참고 - <https://www.gjcity.go.kr/bigdata/home/programs/pictures/list?menu=395>

위 페이지는 Carousel(슬라이드 효과) panellum(사진 파노라마효과) 가 동시에 적용되어 있는 페이지

1. 테스트게시판 메뉴 밑에 새로운 페이지를 만들기 (메뉴명은 대충 파노라마 사진첩 정도)
2. 페이지 검사 및 개발자 도구를 활용하여 (\*)소스코드를 참고하기 (적용된 소스를 찾는법을 익히고 타 사이트 벤치마킹 경험)
3. 필요한 오픈소스(panellum)를 다운받아서 1. 페이지에 적용하기. 이미지는 (\*)에서 다운로드 (오픈소스 적용 경험)

(추가 목표(안해봐도 됨)) 캐러셀 까지해서 두개의 오픈소스 혼합사용

css는 해당사이트에서 긁어서 사용해도 되고, 위 기능만 있어도 상관없습니다.

사용했을때

사용 X