
PORTFOLIO

WEB DEVELOPER

| | |
|------|---|
| 이름 | 유태인 |
| 생년월일 | 1990. 01. 21 |
| 이메일 | ytn42@naver.com |
| 깃허브 | https://github.com/Busterz3682 |

Project

SideWiki

About project



그동안 공부했던 내용들을 바탕으로
혼자만의 힘으로 무언가를 만들어보기 위한 개인프로젝트 입니다

PROJECT CONTENTS

01. 프로젝트 소개

02. 페이지 소개

03. ERD


04. 기능 소개

Project

01. 프로젝트 소개

Introduce project

| | |
|---------|---|
| 개발 기간 | 2022. 12. 10 ~ |
| 개발 인원 | 1명 (개인 프로젝트) |
| 프로젝트 내용 | 위키피디아와 같은 웹 백과사전 누구나 문서를 읽고, 쓰고, 수정, 삭제요청을 할 수 있습니다 문서 삭제는 관리자 권한을 가진 유저만 가능합니다 |
| | |
| 언어 | Java (JDK 1.8), HTML/CSS, JavaScript |
| 서버 | Apache Tomcat 9.0 |
| 프레임워크 | Spring Framework 5.2.9, MyBatis 3.5.11 |
| DB | Oracle 11g |
| IDE | STS, VisualStudioCode |



Wikidata

The Free Encyclopedia

다언어로
읽고 쓰기 쉬운
인터넷 용어 사전

Interaction
문서 작성
도움말
정책과 지침

Not logged in Talk Contributions Create account Log out

Read View source View history

위키백과 미러사이트 [편집]

이 문서의 마지막 수정 시점은 Fri Dec 16 21:10:14 KST 2022입니다

위키백과 또는 위키(영어)는 누구나 자유롭게 쓸 수 있는 다양한 인터넷 백과사전이다. 2001년 1월 15일 지미 웨일스와 래리 산어가 창립하였으며, 대표적인 집단의 사설로 평가받고 있다.

| | |
|-----------|--------|
| 특화 | [hide] |
| 1 역사 | |
| 2 주요 이력 | |
| 3 특징 | |
| 4 평가 | |

1.역사

위키백과 이전에도 온라인 백과사전용 운영체제는 여러 시도들이 있었으나 실패하였다. 위키백과 영향받은 전문가들이 작성했던 백과사전인 뉴미디어가들은 존재하지 않았으며 비롯되었다. 뉴미디어는 웹 및 소셜 미디어 보시마가 2000년 3월 9일에 게시되었나, 보시마의 CEO였던 지미 웨일스도 편집할 때라 생기는 뉴미디어와의 공통점을 모든 콘텐츠로서 재사용하기로 하였고 리눅스 스펀에서 주도한 GNU 자유 문서 라이선스로 제공하였다. 뉴미디어는 그리 성공적이지 않았고, 지미 웨일스와 래리 산거는 누구나 참여할 수 있는 백과사전으로 위키백과를 개발하였다. 생기는 모두의 백과사전이라는 목표를 분명히 하기 위해 이름을 위키를 넣었다. 2001년 1월 10일, 산거는 뉴미디어 메일링 리스트를 통해 뉴미디어 프로젝트를 보완하기 위해 위키를 도입했다고 밝혔다.

2.주요 이력

2007년 1월 위키백과는 처음으로 가장 인기있는 웹사이트 리스트 Top10에 이름을 올렸다. 컴스코어는 위키백과의 연간 방문자를 4,290만 명으로 집계하여 9위로 올렸고, 《뉴욕타임즈》는 10위로 올렸다. 애플은 11위로 집계하였다. 2006년의 순위가 33위였던 것에 비해 더 높은 순위를 받았다. 2015년 3월 위키백과는 5위를 기록하였다. 이 시기 순위는 일렉사 인터넷의 조사에 의한 것으로, 위키백과는 2014년 내내 네 월 8억 이상의 페이지 뷰를 기록하였다.

2012년 1월 영어 위키백과는 미국의 오픈런저 자비전 협회 공인 ISO9001의 지적 재산권 보호 법안(DIPAA)의 제정 시도에 맞아 SOPAR와 PIRSA 반대 시위의 일환으로 24시간 블랙아웃 시위를 열었다.

2014년 1월 20일 수포드 버드쇼브(Sudbury Vernalis)에 위치한 굴을 통해 위키백과가 2012년 12월부터 2013년 12월 사이에 전체적으로 페이지 뷰가 10(천)트럭에 달하는 2억 번 이상의 페이지뷰를 올렸다고 발표하였다. 주요 언어권에 따라 나눈 영어 위키백과의 페이지뷰 양순율은 12%, 독일어가 17%, 일본어는 9%였다. 바르라는 “전반 위키백과 운영자들이 통제 깊게에 오류가 있다고 주장한다”던 지난해 도입된 구글의 지식 그래프가 그 입술 다듬겨 할 것”이라고 덧붙였다.[48] 뉴욕 대학교의 북구수 플레이 세키는 지식 그래프가 다른 사이트들의 페이지뷰를 망각하고 있는 것에 대해 ‘잠여 페이지에서 당당한 질문에 대한 답을 바로 볼 수 있다는 굳이 그 사이트들 방문하려는가’라고 반론하였다

2016년 12월 위키백과는 가장 인기있는 웹사이트 리스트에 5위로 기록되었다.

3.특징

위키백과의 가장 큰 특징은 누구나 편집과 관리에 참여할 수 있다는 점이다. 인터넷을 통해 누구나 글을 고칠 수 있는 체제인 위키로 만들어져 있어 집단 지성적 특성을 가진다. 개방성은 위키백과의 가장 큰 특징 가운데 하나로, 원칙적으로 사용자는 누구나 거의 모든 문서를 새로 만들고 수정할 수 있다.

그러나 이러한 장점은 동시에 역기능인 문서의 훼손이나 부정확한 내용의 수록에 취약하다는 역점이 되고있다. 위키백과 커뮤니티는 이러한 약점을 보완하기 위해 편집 규칙을 정하고 일부 문서에 대한 생성과 편집권을 규제하고 있다.

4.평가

2005년에 영국의 데일리 저널 브리티쉬커 백과사전과 위키백과의 42가지 난해한 과학 기사를 비교한 평가 리뷰를 발표하였다. 그 결과는 위키백과의 정확도 수준이 브리티쉬커 백과사전과 근접했다는 것이었다. 다만, 비평가들은 모든 범위의 기사에 대한 무료 표본 추출을 허가하나, 사회과학 분야는 논쟁적인 시료 문제로 중량을 주 유사한 연구의 경우에서는 그렇게 광범위한 결과가 나오지 못하였을 수도 있다고 여겼다. 또 2006년 기사에서 타임지는 위키백과는 누구나 편집할 수 있는 개방 정책을 갖고 있으며, 이로써 위키백과가 세계에 서 가장 큰 백과사전으로 만들어질 수 있도록 하고, 지역 문화권의 비전문가를 활용한다고 밝혔다.

위키백과는 누구나 참여할 수 있기 때문에 편집자의 시각에 따라 누군가가 악의적인 의도 등으로 잘못된 정보를 입력할 수 있고, 이에 따라 잘못된 정보가 퍼지날 수 있다는 문제점이 제기되어 왔다. 예를 들어 2005년 영어 위키백과에서는 존 시켄슬라라는 미국의 전직 연방 판사는 존 케네디 대통령 암살에 관여했다는 잘못된 정보가 올라가 있었는데, 한 익명 사용자가 신념대로는 미국의 코피카인이 사망했다는 거짓 정보를 올리 인터넷 전반에 잘못된 소문이 퍼지기도 했다.

이 만연함의 위험, 영어판 위키백과에 종종있는 문서 위반 내용 의 60% 정도가 잘못된 사실을 담고 있다는 연구결과가 나왔다. 연구소의 조수는 이를 해결 회사를 이미지가 긍정적으로 만들기 위해 위키백과에 회사에 관련된 내용을 삽입하기 때문인 것으로 추정하고 있다.

여기는 아직 가능하지 않습니다

여기는: 독립자 · 크리몽

Percipit - Minasarchum · Molestie · Phaedrum · Luptatum constium · Habeo adipiscit - Inani zri - Foremsibus sea · Habeo adipiscit - Minimum corruptum · Regione suscept · Has et partem · Percipit - Minasarchum · Molestie · Phaedrum · Luptatum constium · Habeo adipiscit - Inani zri - Vel mit abscius · Habeo adipiscit · Minimum corruptum · Regione suscept · Percipit maenstato · Regione suscep · Percipit maenstato

Subito

Minimum corruptum | Regione suscept | Has et partem

This page was last edited on 29 Oct 2017 at 23:07 by HTML5 Templates.

Privacy policy About Terms and conditions Cookie statement Developers

페이지 소개

The image shows a screenshot of a Wikipedia page titled "페이지 소개를 위한 문서입니다 [편집]" (Page introduction document [edit]). The page content includes a table of contents, a list of links, and a "Page Not Found" message. Annotations with arrows point to specific features:

- 회원가입, 로그인** (Sign up, Log in) points to the top right navigation links.
- 로그인 안됨** (Not logged in) points to the user status link.
- 토크** (Talk) points to the "토론" (Talk) link in the top navigation bar.
- 현재 문서에 대한 토론페이지** (Discussion page for this document) points to the "토론" link.
- 삭제요청** (Request for deletion) points to the "삭제요청" link in the top navigation bar.
- 문서 검색** (Search document) points to the search bar on the right.
- 현재 문서 수정 폼** (Current document edit form) points to the "[편집]" (edit) link.
- 등록된 이미지가 없음** (No image registered) points to the "NO IMAGE" placeholder.
- 관리자 전용 페이지** (Admin-only page) points to the "삭제요청목록" (List of requests for deletion) link in the left sidebar.
- 등록되어있는 문서 중 랜덤한 문서로 이동** (Move to a random document among registered documents) points to the "대문으로" (Main page) link.
- 등록되어있는 전체 문서 목록** (List of all registered documents) points to the "임의의 문서로" (Random document) link.
- 문서 작성 폼** (Document creation form) points to the "문서 작성" (Create document) link.
- 에러 페이지** (Error page) points to the "Page Not Found" message.

The "Page Not Found" message reads: "Page Not Found. Sorry, but the page you were trying to view does not exist."

ERD

Wiki(ORACLE)

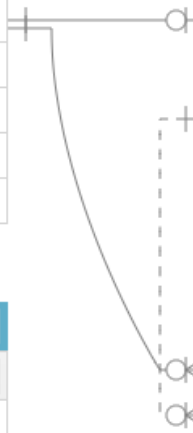
| DOC | | | | | |
|------------|----|----|------|--------------|------------------------|
| 문서에 대한 테이블 | | | | | |
| PK | AI | FK | Null | Logical Name | Name Type |
| ✓ | | | | 문서 제목 | DOC_TITLE VARCHAR2 |
| | | | | 문서 개요 | DOC_CONTENT VARCHAR2 |
| | | | | 문서 등록(수정)일자 | DOC_DATE DATE |
| | ✓ | | | 문서 번호 | DOC_ID NUMBER |
| | | | ✓ | 문서이미지 등록여부 | DOC_IMAGE NUMBER |

| DELETEREQUEST | | | | | |
|------------------|----|----|------|--------------|-----------------------|
| 삭제 요청이 들어온 문서 목록 | | | | | |
| PK | AI | FK | Null | Logical Name | Name Type |
| | | | | | DOC_TITLE VARCHAR2 |

| USERS | | | | | |
|-------|----|----|------|--------------|-----------------------|
| | | | | | |
| PK | AI | FK | Null | Logical Name | Name Type |
| ✓ | | | | | EMAIL VARCHAR2 |
| | | | ✓ | | PASSWORD VARCHAR2 |
| | | | ✓ | | GRADE VARCHAR2 |

| CHAPTER | | | | | |
|---------------|----|----|------|--------------|------------------------|
| 문서 목차에 대한 테이블 | | | | | |
| PK | AI | FK | Null | Logical Name | Name Type |
| | | ✓ | | 문서 제목 | DOC_TITLE VARCHAR2 |
| | | | ✓ | 목차 제목 | CHAPTER_TITLE VARCHAR2 |
| | | | ✓ | 목차 번호 | CHAPTER_INDEX NUMBER |

| DETAIL | | | | | |
|---------------------|----|----|------|--------------|--------------------------|
| 문서 목차별 상세내용에 대한 테이블 | | | | | |
| PK | AI | FK | Null | Logical Name | Name Type |
| | | ✓ | | 문서 제목 | DOC_TITLE VARCHAR2 |
| | | ✓ | | 목차 번호 | CHAPTER_INDEX NUMBER |
| | | | | 목차별 상세내용 | CHAPTER_CONTENT VARCHAR2 |
| | | | | 목차별 상세내용 번호 | DETAIL_INDEX NUMBER |



요청 URL

| HTTP Verb | Path | Used For |
|-----------|----------------------------|-----------|
| GET | /wiki/insertDoc | 새 문서 입력 폼 |
| POST | /wiki/insertDoc | 새 문서 입력 |
| GET | /wiki/getDoc/{docTitle} | 문서 조회 |
| GET | /wiki/updateDoc/{docTitle} | 문서 수정 폼 |
| POST | /wiki/updateDoc | 문서 수정 |
| GET | /wiki/getRandomDoc | 랜덤 문서 조회 |
| GET | /wiki/searchDoc | 문서 검색 |
| GET | /debate/detail/{docTitle} | 토론페이지 이동 |
| POST | /debate/insertreply | 댓글 입력 |
| POST | /debate/deletereply | 댓글 삭제 |
| ... | ... | ... |

회원가입

이메일

you@example.com

인증코드

이메일 인증코드 발송하기

비밀번호(*)

비밀번호

비밀번호확인(*)

비밀번호 확인

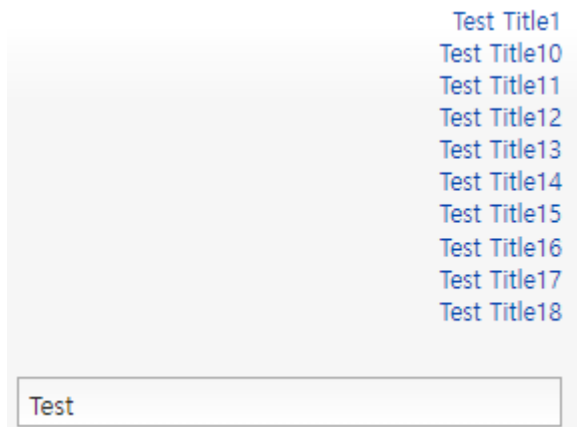
가입하기

- 회원 가입
 - 이메일 인증 시스템
 - Spring Security BcryptPasswordEncoder 사용 암호화
 - 로그인 요청시 Interceptor에서 검증

```
/*이메일 인증코드 발송*/
let code = "";
$("#sendCode").click(function() {
    if($("#email").val() == "") {
        alert('아이디(이메일)를 입력해주세요. ');
        return;
    }
    $("#sendCode").attr('disabled', true);
    const id = $("#email").val();
    $.ajax({
        type: "post",
        url: "http://localhost:8181/user/idCheck",
        data: {"email" : id},
        dataType: "text",
        success: function(result) {
            if(result == "ok"){
                $.ajax({
                    type: "post",
                    url: "http://localhost:8181/user/mailCheck",
                    data: {"email" : id},
                    dataType : "text",
                    success: function(result) {
                        if(result != 'notvalid'){
                            $("#codeChk").show();
                            code = result;
                            alert('인증번호가 발송되었습니다 이메일을 확인해주세요');
                            document.getElementById("sendCode").innerHTML = "인증번호가 발송되었습니다";
                            document.getElementById("sendCode").style.backgroundColor = "gray";
                        } else {
                            alert('메일전송실패');
                            $("#sendCode").attr('disabled', false);
                        }
                    }
                },
                error: function() {
                    alert('오류발생');
                }
            });
        } else if(result == 'exist') {
            alert('이미 가입된 이메일입니다');
            $("#sendCode").attr('disabled', false);
        }
    }
},
error: function() {
    alert('서버에러 입니다. 관리자에게 문의하세요. ');
    location.reload();
}
});
});
```

| EMAIL | PASSWORD | GRADE |
|-----------------|---|--------|
| ytn42@naver.com | \$2a\$10\$Ghtxd79zJQxjJBhNsI1TauyVqNeH0nkZbki3R0J4fqPcscyodA0U6 | normal |

문서검색



- 문서검색
 - 검색어로 시작하는 문서 검색
 - 최대 10개까지 보여줌

```
$('#searchInput').keyup(function(){
    let s = $('#searchInput').val()
    console.log(s)
    $.ajax({
        type : 'get',
        url : '/wiki/searchDoc',
        data : {"search" : s},
        success : function(result){
            console.log(result);
            $('#searchResult').empty();
            for(let i = 0; i < (result.length < 10 ? result.length : 10); i++) {
                $('#searchResult').append("<li><a href='http://localhost:8181/wiki/getDoc/'+result[i]+'>"+result[i]+"</a></li>")
            }
        },
        error : function() {
            console.log('no result');
        }
    });
});
</script>
```

```
@ResponseBody
@GetMapping("/searchDoc")
public List<String> searchList(String search){
    ArrayList<String> searchResult = (ArrayList<String>) documentService.searchList(search);
    return searchResult;
}
```

문서작성

문서 제목을 입력해주세요

위키백과, 우리 모두의 백과사전.

개요를 입력해주세요

파일 선택

선택된 파일 없음

이미지 파일을 업로드해주세요

목차를 입력해주세요

추가

추가

추가

목차를 입력해주세요

추가

추가

추가

등록하기

```
//문서 작성
@PostMapping("/insertDoc")
public String insertDoc(HttpServletRequest req, @RequestParam(name = "file", required = false) MultipartFile file) {
    Logger.info("insertDoc 요청들어옴");
    String docTitle = req.getParameter("docTitle");
    String docContent = req.getParameter("docContent");
    DocumentVO vo = new DocumentVO(docTitle, docContent, null, 0, 0);

    File uploadPath = new File("c:\\upload\\" + docTitle);
    if(!uploadPath.exists()) {
        uploadPath.mkdirs();
    }
    Logger.info("Upload File Name : " + file.getOriginalFilename());
    Logger.info("Upload File size : " + file.getSize());
    if(file.getSize() > 0) {
        File saveFile = new File(uploadPath, "image.jpg");
        vo.setDocImage(1);
        try {
            file.transferTo(saveFile);
        } catch (Exception e) {
            Logger.error(e.getMessage());
        }
    }
}

//문서(텍스트) 작성 시작
List<ChapterVO> cList = new ArrayList<ChapterVO>();
for(int i = 0; i < req.getParameterValues("chapterTitle").length; i++) {
    cList.add(new ChapterVO(docTitle, i + 1, req.getParameterValues("chapterTitle")[i]));
}
List<DetailVO> dList = new ArrayList<DetailVO>();
for(int i = 1; i <= req.getParameterValues("chapterTitle").length; i++) {
    for(int j = 0; j < req.getParameterValues("chapterContent" + i).length; j++) {
        dList.add(new DetailVO(docTitle, i, req.getParameterValues("chapterContent" + i)[j], j + 1));
    }
}
// 문서(텍스트) 작성 끝
documentService.insertDoc(vo, cList, dList);
return "redirect:/";
}

@Transactional
public void insertDoc(DocumentVO vo, List<ChapterVO> vo2, List<DetailVO> vo3) {
    documentMapper.insertDoc(vo);
    debateBoardMapper.makeDebatePage(vo.getDocTitle());
    for(ChapterVO item : vo2) {
        documentMapper.insertChapter(item);
    }
    for(DetailVO item : vo3) {
        documentMapper.insertDetail(item);
    }
}
```

문서작성

문서 제목을 입력해주세요

위키백과, 우리 모두의 백과사전.

개요를 입력해주세요

파일 선택 선택된 파일 없음

이미지 파일을 업로드해주세요

목차를 입력해주세요

추가

추가

추가

목차를 입력해주세요

추가

추가

추가

등록하기

```

<script>
let ci = 2;
function addChapter(){
    let addFrm = '<h2><input type="text" name="chapterTitle" id="chapterTitle" placeholder="목차를 입력해주세요">'
    + '<input id="addChap" type="button" onClick="addChapter()" value="추가"></h2>'
    + '<p><textarea name="chapterContent'+ci+'" id="chapterContent" cols="75" rows="5"></textarea></p>'
    + '<input id="addCont" type="button" onClick="addContent()" value="추가">'
    + '<input type="hidden" name="chapterIndex" value="'+ ci +'>';
    $('#inputWrap').append(addFrm);
    ci++;
}

function addContent(){
    let addFrm = '<p><textarea name="chapterContent'+ (ci-1) +' id="chapterContent" cols="75" rows="5"></textarea></p>'
    + '<input id="addCont" type="button" onClick="addContent()" value="추가">';
    $('#inputWrap').append(addFrm);
}
</script>
<script>
/* 유효성 체크 */
function check() {
    let title = $('#docTitle').val();
    let rtn = false;
    if(title == '') {
        alert('문서 제목을 입력해주세요');
    } else {
        $.ajax({
            type : 'get',
            url : '/wiki/check',
            data : {"docTitle" : title},
            success : function(result){
                console.log(result);
                if(result == 'ok') {
                    alert('문서가 등록되었습니다');
                    rtn = true;
                } else {
                    alert('이미 등록되어있는 문서입니다');
                }
            },
            error : function() {
                alert('오류발생')
            }
        });
    }
    return rtn;
}
</script>

```

문서삭제

삭제요청이 들어온 문서 목록

Table #6

| 문서제목 | |
|------------------|------|
| 문서삭제요청목록 테스트 172 | 삭제하기 |
| 문서삭제요청목록 테스트 173 | 삭제하기 |
| 문서삭제요청목록 테스트 174 | 삭제하기 |
| 문서삭제요청목록 테스트 175 | 삭제하기 |
| 문서삭제요청목록 테스트 176 | 삭제하기 |
| 문서삭제요청목록 테스트 177 | 삭제하기 |
| 문서삭제요청목록 테스트 178 | 삭제하기 |
| 문서삭제요청목록 테스트 179 | 삭제하기 |
| 문서삭제요청목록 테스트 180 | 삭제하기 |
| 문서삭제요청목록 테스트 181 | 삭제하기 |

1 2 3 4 5 6 7 8 9 10 다음

```
public class UserAuthHandler extends HandlerInterceptorAdapter {

    //관리자권한이 필요한 페이지 요청이 들어왔을 때 요청을 가로채 확인할 인터셉터.
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        throws Exception {

        //세션에서 로그인 데이터를 얻은 후 확인
        HttpSession session = request.getSession();
        UserVO vo = (UserVO) session.getAttribute("user");
        System.out.println(vo);
        if(vo == null) { //로그인이 안되어있으면
            response.sendRedirect(request.getContextPath() + "/user/loginPage");
            return false;
        } else if(!vo.getGrade().equals("admin")) { //관리자가 아니면
            response.setContentType("text/html; charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("<script>alert('권한이 없습니다.');" history.go(-1);</script>");
            out.flush();
            return false;
        } else {
            return true;
        }
    }
}

<!-- 권한을 체크하는 PreHandler -->
<mvc:interceptor>
    <mvc:mapping path="/wiki/delete/*"/>
    <mvc:mapping path="/wiki/getReqList"/>
    <ref bean="userAuthHandler" />
</mvc:interceptor>
```

- 문서삭제

- Interceptor로 권한을 체크하여 관리자만 접근가능
- 테이블 생성시 설정한 ON DELETE CASCADE 제약조건에 의하여 문서 삭제시 자식테이블의 데이터도 함께 삭제

전체문서목록, 랜덤문서

전체 문서 목록

현재 104개의 문서가 등록되어있습니다

최근 등록(혹은 수정)된 순서대로 보여집니다

Table #8

| 문서번호 | 문서제목 | 작성일자 |
|-------------------------|--------------|----------------------------------|
| 17 | Test Title4 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 18 | Test Title5 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 19 | Test Title6 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 20 | Test Title7 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 21 | Test Title8 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 22 | Test Title9 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 23 | Test Title10 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 24 | Test Title11 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 25 | Test Title12 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 26 | Test Title13 | Thu Nov 24 21:36:40 KST 2022상세보기 |
| 1 2 3 4 5 6 7 8 9 10 다음 | | |

```
//문서목록
@GetMapping("/getDocList")
public String getdocList(@RequestParam(required = false, defaultValue = "1", name="currpage") int currPage ,Model model) {
    logger.info("getDocList 요청 들어옴");
    int totalCount = documentService.getTotalCount();
    int pageSize = 10;
    int blockSize = 10;
    PagingVO vo = new PagingVO(currPage, totalCount, pageSize, blockSize);
    ArrayList<DocumentVO> doclist = (ArrayList<DocumentVO>) documentService.getDocList(vo);
    model.addAttribute("totalCount", totalCount);
    model.addAttribute("doclist", doclist);
    model.addAttribute("page", vo);
    return "document/docList";
}

<select id="getRandomDoc" resultMap="doc">
    select *
    from
    (select * from doc order by dbms_random.value)
    where rownum = 1
</select>
```

- 전체 문서 목록
- 전체 문서 목록 페이지징처리
- Oracle 함수를 활용한 랜덤문서 읽기

토론페이지

토론 : 페이지 소개를 위한 문서입니다

이 페이지는 문서 제목 : 페이지 소개를 위한 문서입니다 에 대한 토론 페이지입니다

작성자 : 환영합니다 작성일 : 2022-12-19 05:43:30

내용 : 여기는 토론게시판입니다

삭제

작성자 : 1 작성일 : 2022-12-19 05:42:18

내용 : 댓글 예시입니다

삭제

작성자 : 2 작성일 : 2022-12-19 05:44:21

내용 : 댓글 예시입니다

삭제

작성자 : 환영합니다 작성일 : 2022-12-19 05:41:26

내용 : 여기는 토론게시판입니다

삭제

작성자

비밀번호

작성

```
$('#insertReplyBtn').click(function() {
    let docTitle = $('#docTitle').val();
    let content = $('#content').val();
    let writer = $('#writer').val();
    let password = $('#password').val();
    if (content === '') {
        alert('내용을 입력해주세요');
    } else if (writer === '') {
        alert('작성자를 입력해주세요');
    } else if (password === '') {
        alert('비밀번호를 입력해주세요');
    } else {
        $.ajax({
            type : "POST",
            url : "http://localhost:8181/debate/insertreply",
            dataType : "json",
            data : {
                "docTitle" : docTitle,
                "content" : content,
                "writer" : writer,
                "password" : password
            },
            success : function(result) {
                if (result == 1) {
                    alert('댓글등록성공');
                    location.reload();
                } else {
                    alert('오류발생');
                }
            }
        });
    }
});
```

```
function addDeleteBtn(clickedId) {
    let docTitle = $('#docTitle').val();
    let pw = $('#pw'+clickedId).val();
    $.ajax({
        type : "POST",
        url : "http://localhost:8181/debate/deletereplay",
        dataType : "json",
        data : {
            "docTitle" : docTitle,
            "password" : pw,
            "replyNo" : clickedId
        },
        success : function(result) {
            if (result == 1) {
                alert('삭제성공');
                location.reload();
            } else {
                alert('오류발생');
            }
        }
    })
}
```

- 토론페이지

THANK YOU
