

손규진

## AI기반 자율주행 솔루션 개발 플랫폼

사용자 및 개발자들이 쉽게 접근하고 질문하는 것이 가능하고  
AI 기반 자율주행 솔루션에 관한 최신 정보와 자료를 공유하여  
개발자들 간의 지식 공유와 기술 발전을 촉진합니다.

—  
포트폴리오

# 포트폴리오

## CONTENTS

- 1** 자기소개
- 2** 프로젝트 요구사항
- 3** 웹애플리케이션



# 손규진

## 자기소개

맡은 일에 최선을 다하는 개발자를 꿈꾸는 손규진입니다.

## 개발언어

JAVA, JSP, HTML5, CSS3, JavaScript, Ajax, Python,  
C, C++, C#

## 학력

-한남대학교 정보통신공학과 4학년 졸업  
-미래융합교육원 2023.11월반 수료

**프로젝트 요구사항**

|

02

# 기능 요구사항



정보공유

자료실, 개발자 노트



커뮤니티

자유게시판, 질문게시판



AI

자율주행 체험관

# 개발환경

## Tool

구분	TOOL	Version
IDE	Spring Tool Suite	3.9.12
	Visual Studio Code	1.87.2
	Arduino IDE	2.3.2
DB	Sql developer	
	exERD	2.3.9

## Language

구분	Version	구분
JAVA	1.8	JAVA
Python	3.12.1	Python
Javascript		Javascript
HTML	5	HTML
CSS	5	CSS

## ROS

구분	용도
아두이노	ESP32-CAM
	OLED
아두이노 드라이버	ESP 2.0.11
OLED 드라이버	ESP8266, ESP32 OLED driver 4.4.0
무선 LAN	ipTIME N150UA3 (Wi-Fi 150mbps)

## Server

구분	용도	Version
Tomcat	WAS	9.0.84

## DB

구분	Version
Oracle	11g XE

## Framework

구분	이름	Version
JAVA	Spring	4.3.2
Javascript	BootStrap	
Python	flask	3.0.2

## Library

구분	이름	용도
Javascript	Toast UI	게시글 에디터
	Full Calendar	캘린더 구현
Python	PyTorch	애플리케이션
	OpenCV	카메라 영상 처리
	Keras	자율주행 모델
	TensorFlow	머신러닝
딥러닝 모델	YOLOv5	객체 검출 모델

컴퓨터 비전



딥러닝 학습

 PyTorch



TensorFlow

YOLO  v5

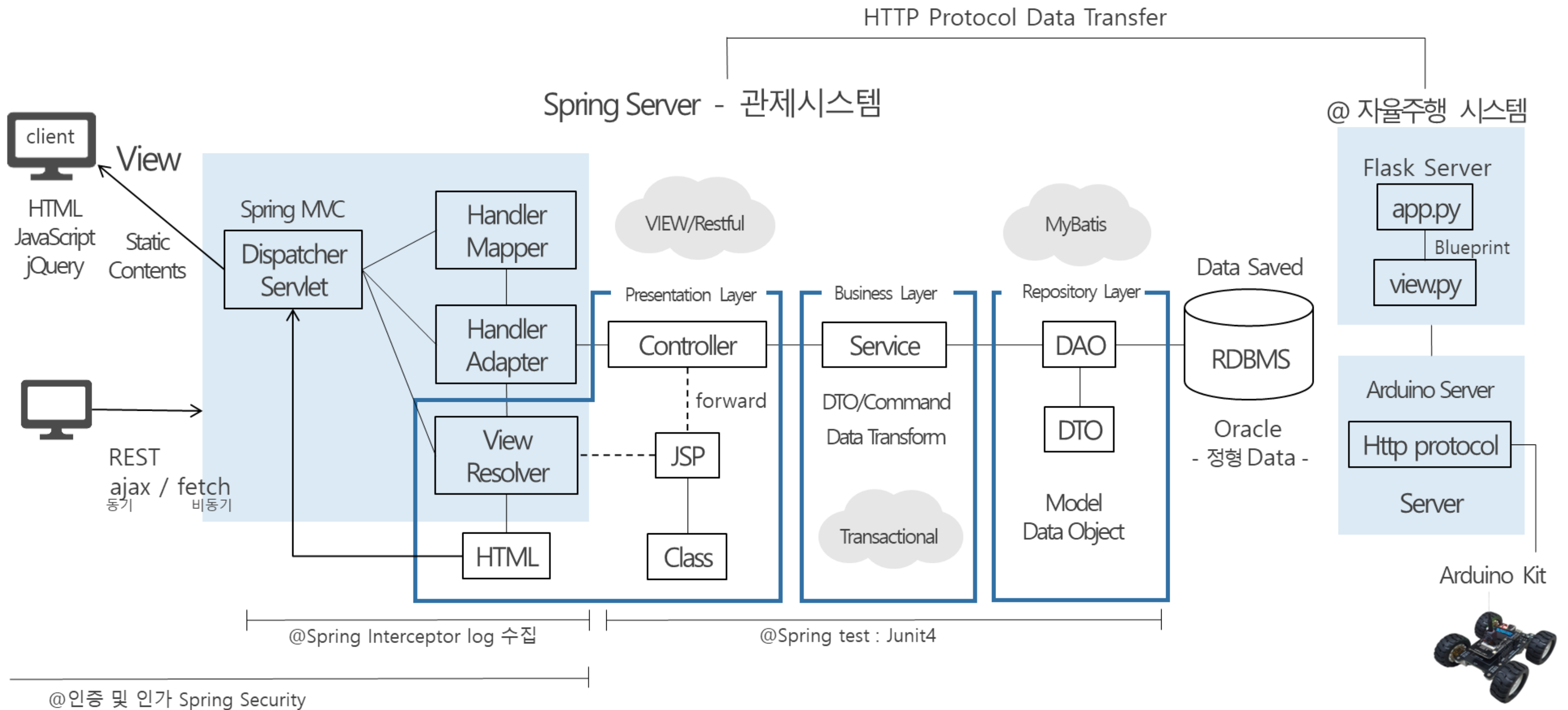
웹



Flask

web development,  
one drop at a time

# 시스템 구상도





**웹애플리케이션**

|

02

# 회원가입

## 회원가입

일반 사용자

개발자

선택

사용자명

사용자명

이메일

이메일

비밀번호

8자 이상 입력해주세요.

비밀번호 확인

비밀번호 재입력

주력 언어

없음

등록

취소

개발자 선택 시,  
나타나며 자신의  
주력언어 선택

등록(개발자)

등록(일반사용자)

localhost:8080의 메시지

개발자 확인용 퀴즈를 맞히시면 가입이 승인됩니다.

확인

Java 프로그램을 실행하기 위해 필요한 가장 기본적인 요소는 무엇입니까?  
a) JDK (Java Development Kit)  
b) JVM (Java Virtual Machine)  
c) JRE (Java Runtime Environment)  
d) JAR (Java Archive)

정답을 입력하세요

확인

정답 입력

localhost:8080의 메시지

등록을 성공했습니다.

확인

# 이용자 회원정보 조회

테스터    로그아웃

마이페이지

회원 정보

현재 비밀번호 입력

현재 비밀번호 입력

확인    취소

회 원 정 보

회원명	테스터
가입일자	2024.05.12
이메일	snoowert@naver.com
권한	developer

수정    탈퇴

# 운영자 회원정보 조회

손규진    로그아웃

마이페이지

회원 정보

회원 목록

회원목록

이용중인 회원 ▾

번호	회원명	탈퇴여부	가입일자	권한
19	테스터	N	2024.05.12	developer
18	ccc	N	2024.04.25	developer
17	nnn	N	2024.04.25	normal
16	www	N	2024.04.25	developer
15	xxx	N	2024.04.24	normal
14	ss	N	2024.04.24	developer
13	손규w	N	2024.04.22	developer
12	son	N	2024.04.15	normal
9	마	N	2024.04.11	developer
7	다	N	2024.04.11	normal

<<

<

1

2

>

>>

회 원 정 보

회원명

테스터

가입일자

2024.05.12

이메일

snoowert@naver.com

권한

developer

탈퇴 처리

클릭하여 탈퇴처리  
및 계정 복구를 할  
수 있다.

# 로그인

## 로그인

이메일을 입력하세요.

패스워드를 입력하세요.

로그인

```
<div class="col-sm-10 mx-auto">
  <h3>로그인</h3>
  <form action="login.do" method="post">
    <input name="retUrl" value="${retUrl}" type="hidden" />
    <div class="form-group has-feedback">
      <input type="text" class="form-control" name="email" placeholder="이메일을 입력하세요." value="">
      <span class="glyphicon glyphicon-envelope form-control-feedback"></span>
    </div>
    <div class="form-group has-feedback">
      <input type="password" class="form-control" name="password" placeholder="패스워드를 입력하세요." value="">
      <span class="glyphicon glyphicon-lock form-control-feedback"></span>
    </div>
    <div class="row">
      <!-- /.col -->
      <div class="col-sm-4">
        <button type="submit" class="btn btn-primary btn-block btn-flat">로그인</button>
      </div>
      <!-- /.col -->
    </div>
  </form>
  <!-- 로그인 실패 시 에러 메시지 표시 -->
  <c:if test="${param.error != null}">
    <p style="color: red;">아이디 또는 비밀번호가 맞지 않습니다.</p>
  </c:if>
</div>
```

```
public Authentication authenticate(Authentication auth) throws AuthenticationException {
    String login_id = (String) auth.getPrincipal(); // 로그인 시도한 ID를 가져온다
    String login_pwd = (String) auth.getCredentials(); // 로그인 시도한 Password 를 가져온다.

    try {
        MemberVO member = memberService.login(login_id, login_pwd);

        UserDetails authUser = new User(member);

        boolean invalidCheck = authUser.isEnabled();
        if (invalidCheck) {
            //로그인 성공
            // 스프링 시큐리티 내부 클래스로 인증 토큰 생성한다.
            UsernamePasswordAuthenticationToken result
                = new UsernamePasswordAuthenticationToken(
                    authUser.getUsername(), authUser.getPassword(),
                    authUser.getAuthorities());

            // 전달할 내용을 설정한 후
            result.setDetails(authUser);
            // 리턴한다. successHandler로 전송된다.
            return result;
        }
    }
}
```

```
<authentication-manager alias="authenticationManager">
  <authentication-provider ref="customAuthenticationProvider" />
</authentication-manager>
<form-login login-page="/member/LoginForm.do"
  login-processing-url="/member/Login.do"
  username-parameter="email"
  password-parameter="password"
  authentication-success-handler-ref="LoginSuccessHandler"
  authentication-failure-handler-ref="LoginFailureHandler">
  <beans:bean id="LoginSuccessHandler"
    class="com.spring.security.LoginSuccessHandler"
    p:defaultTargetUrl="/main/"
    p:loginUserLogSavePath-ref="LoginUserLogPath"
    p:loginUserLogSaveFileName-ref="LoginUserLogFileName" />
  <beans:bean id="LoginFailureHandler"
    class="com.spring.security.LoginFailureHandler" />
</form-login>
```

Security-Context

```
@Override
public void onAuthenticationSuccess(HttpServletRequest request,
    HttpServletResponse response,
    Authentication authentication) throws ServletException, IOException {

    User user = (User)authentication.getDetails();
    MemberVO loginUser = user.getMemberVO();

    // session 저장
    HttpSession session = request.getSession();
    session.setAttribute("loginUser", loginUser);
    session.setMaxInactiveInterval(6*3600);

    //application context
    ServletContext ctx = request.getServletContext();
    ctx.setAttribute("loginUser", loginUser.getMemberid());
}
```

로그인 성공시  
session에 저장,  
세션 유지시간 설정

제목

검색어를 입력하세요.

Q

번호	제 목	작성자	작성일	조회수
44	ss	손규w	2024-04-30	2
40	sdfghjkl	손규진	2024-04-22	22
39	123	손규진	2024-04-22	8
38	123	손규진	2024-04-19	1
29	ㅈㄷㅂㄷ12	손규진	2024-04-08	28
28	ㅂㅈㄷㅂㅈㄷㅈ1수정	손규진	2024-04-08	9
27	테스트용ㄴㅇㄴ	손규진	2024-04-08	3
26	테스트용	손규진	2024-04-08	3
15	새로쓴제목	손규진	2024-04-04	1
				1

제목 혹은  
내용으로 검색

테스트

손규진

작성일 2024.05.12

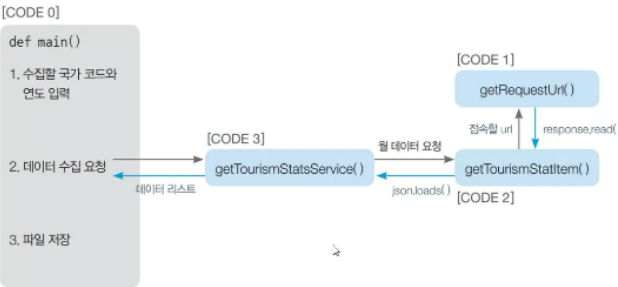
조회수 1

파일

로지스틱회귀리포트.docx

테스트 내용

- 공공데이터 크롤링
- 2. 프로그램 구성 설계하기



자료 등록

테스트

H B I S | — 66 | ☰ ☷ ☑ ☒ ☓ | ☐ ☑ ☒ | </> CB

테스트 내용



Markdown WYSIWYG

첨부파일 : Add File

파일 선택 화면정의서.pptx

등록

취소

# 자료실 파일 업로드

```
public class PdsRegistCommand {  
    private String pdstitle;  
    private String pdscontent;  
    private int memberid;  
    private List<MultipartFile> uploadFile;
```

```
@PostMapping("/regist")  
public ModelAndView regist(ModelAndView mnv, PdsRegistCommand command) throws Exception{  
    String url = "/pds/regist_success";  
  
    List<MultipartFile> pdsFile = command.getUploadFile();  
    String savePath = fileUploadDir;  
    List<PdsFileVO> pdsFileList = saveFileToAttaches(pdsFile, savePath);  
  
    PdsVO pds = command.toPdsVO();  
    pds.setPdsfilelist(pdsFileList);  
  
    pdsService.regist(pds);
```

```
@Override  
public void regist(PdsVO pds) throws SQLException {  
    // TODO Auto-generated method stub  
    List<PdsFileVO> pdsFileList = pds.getPdsfilelist();  
    pds.setPdsid(pdsDAO.selectPdsidSeqNext());  
    pdsDAO.insertPds(pds);  
  
    if(pdsFileList != null) for(PdsFileVO pdsFile : pdsFileList) {  
        pdsFile.setPdsid(pds.getPdsid());  
        pdsfileDAO.insertPdsFile(pdsFile);  
    }  
}
```

파일정보를  
DB에 저장

```
//파일업로드  
private List<PdsFileVO> saveFileToAttaches(List<MultipartFile> multiFiles,  
    String savePath )throws Exception{  
  
    if (multiFiles == null) return null;  
  
    //저장 -> attachVO -> list.add  
    List<PdsFileVO> pdsFileList = new ArrayList<PdsFileVO>();  
    for (MultipartFile multi : multiFiles) {  
        String uuid = UUID.randomUUID().toString().replace("-", "");  
        String fileName = uuid+"$$"+multi.getOriginalFilename();  
  
        //파일저장  
        File target = new File(savePath, fileName);  
        target.mkdirs();  
        multi.transferTo(target);  
  
        //attachVO  
        PdsFileVO pdsFile = new PdsFileVO();  
        pdsFile.setPdsfilename(fileName);  
        pdsFile.setPdsfiletype(fileName.substring(fileName.lastIndexOf('.') + 1)  
            .toUpperCase());  
  
        pdsFileList.add(pdsFile);  
    }  
    return pdsFileList;  
}
```

파일명 고유화를  
위한 UUID

파일을  
경로에 저장



# 토스트 UI

H B I S — 66 | ≡ ≡ ☑ ≡ ≡ | ☐ ☒ ☎ | </> CB

내용을 입력해 주세요.

```
const editor = new toastui.Editor({
  el: document.querySelector('#content'), // 에디터를 적용할 요소 (컨테이너)
  height: '500px', // 에디터 영역의 높이 값 (000px || auto)
  initialEditType: 'wysiwyg', // 최초로 보여줄 에디터 타입 (markdown || wysiwyg)
  placeholder: '내용을 입력해 주세요.', // 내용의 초기 값으로, 반드시 마크다운 문자열 형태여야 함
  previewStyle: 'vertical', // 마크다운 프리뷰 스타일 (tab || vertical)
  hooks: {
    async addImageBlobHook(blob, callback) {
      try {
        // FormData에 이미지 추가
        var formData = new FormData();
        formData.append('image', blob);

        // 서버로 이미지 업로드
        const response = await fetch('/pds/tui-editor/image-upload', {
          method: 'POST',
          body: formData
        });

        // 서버에서 반환된 파일명
        const filename = await response.text();
        // 이미지 URL 생성
        const imageUrl = '/pds/tui-editor/image-print?filename=' + filename;

        // 에디터에 이미지 삽입
        callback(imageUrl, 'image');
      } catch (error) {
        console.error('이미지 업로드 실패:', error);
      }
    }
  }
});
```

Base64 인코딩  
방지를 위한 함수

저장된 이미지 불러오기

```
// 파일명에서 확장자 추출
private String getFileExtension(String filename) {
  int dotIndex = filename.lastIndexOf(".");
  if (dotIndex == -1) {
    return ""; // 확장자 없음
  }
  return filename.substring(dotIndex + 1);
}

@PostMapping("/tui-editor/image-upload")
public ResponseEntity<String> uploadEditorImage(HttpServletRequest request, @RequestParam("image") MultipartFile image) {
  try {
    // 디렉토리가 존재하지 않으면 생성
    File directory = new File(uploadDir);
    if (!directory.exists()) {
      directory.mkdirs();
    }

    // 파일명 생성 (랜덤 UUID + 확장자)
    String filename = UUID.randomUUID().toString() + "." + getFileExtension(image.getOriginalFilename());

    // 업로드할 파일 생성
    File file = new File(uploadDir + File.separator + filename);

    // 이미지 파일 저장
    FileOutputStream outputStream = new FileOutputStream(file);
    outputStream.write(image.getBytes());
    outputStream.close();
    // 파일명 반환
    return ResponseEntity.ok(filename);
  } catch (IOException e) {
    e.printStackTrace();
    return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Image upload failed.");
  }
}

@GetMapping("/tui-editor/image-print")
public ResponseEntity<byte[]> serveFile(@RequestParam("filename") String filename) {
  Path filePath = Paths.get(uploadDir).resolve(filename);
  try {
    if (Files.exists(filePath) && Files.isReadable(filePath)) {
      byte[] data = Files.readAllBytes(filePath);
      HttpHeaders headers = new HttpHeaders();
      headers.setContentType(MediaType.IMAGE_JPEG);
      return new ResponseEntity<>(data, headers, HttpStatus.OK);
    } else {
      return new ResponseEntity<>(HttpStatus.NOT_FOUND);
    }
  } catch (IOException e) {
    e.printStackTrace();
    return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
  }
}
```

토스트 에디터의 이미지 업로드  
로컬 저장소로 변경



# 개발자 노트 피드

글쓰기

손규진

asd

수정 삭제

asd

등록

손규진

qweasd123

수정 삭제

손규진

123

수정 삭제

피드 게시물 작성

피드 제목

H B I S | — 66 | ☰ ☷ ☑ ☒ ☓ | 📁 🖼️ 🔗 | </> CB

Markdown

WYSIWYG

취소

등록

손규진

수정 삭제

테스트

손규진

수정 삭제

테스트

등록 취소

# 개발자 노트 일정 조회

```
@Override
public List<CalendarVO> callList(int projectId) throws SQLException {
    List<CalendarVO> cal = calendarDAO.selectCalendarList(projectId);

    DateFormat dateFomatter = new SimpleDateFormat("yyyy-MM-dd");

    for(CalendarVO calendar : cal) {
        calendar.setStartDate(dateFomatter.format(calendar.getStartd()));
        calendar.setEndDate(dateFomatter.format(calendar.getEndd()));
    }
    return cal;
}
```

Event 형태에 맞게  
Date객체 변환

```
@GetMapping("/calendar")
public ModelAndView calendar(int projectId, ModelAndView mnv) throws SQLException {
    String url = "/devnote/calendar";

    ProjectVO project = projectService.detail(projectId);
    List<CalendarVO> cal = projectService.callList(projectId);

    mnv.addObject("calendar", cal);
    mnv.addObject("projectId", projectId);
    mnv.addObject("project", project);
    mnv.setViewName(url);
    return mnv;
}
```

일정 데이터 전달

```
events : [
    <c:forEach var="event" items="${calendar}">
        {
            title: '${event.title}',
            start: '${event.startDate}',
            end: '${event.endDate}',
            id: '${event.id}',
        },
    </c:forEach>
],
```

일정 삭제를 위한  
데이터

2024년 5월						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6 테스트2	7	8	9 23	10	11
12	13	14	15 테스트1	16 테스트3	17	18
19 테스트1	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

# 개발자 노트 일정 등록

```
select: function(arg) {  
    // 캘린더에서 선택한 날짜 정보를 변수에 저장  
    var startDate = arg.startStr;  
    var endDate = arg.endStr;  
  
    // 프로젝트 창을 통해 일정 제목을 입력 받음  
    var title = prompt('일정 제목을 입력하세요.');
```

if(title == null){  
 return;  
}

// 회원번호와 프로젝트번호  
var memberId = \${loginUser.memberId}+0; // 예시  
var projectId = \${projectId}; // 예시

// 입력받은 정보를 서버로 전송하여 DB에 저장 요청  
saveEventToDB(title, startDate, endDate, memberId, projectId);  
},

```
function saveEventToDB(title, startDate, endDate, memberId, projectId) {  
    $.ajax({  
        url: '/devnote/save-event',  
        method: 'POST',  
        data: {  
            'title': title,  
            'startd': startDate,  
            'endd': endDate,  
            'memberId': memberId,  
            'projectId': projectId  
        },  
        success: function(response) {  
            alert('일정이 성공적으로 저장되었습니다.');
```

// 저장 성공 시 추가적인 작업 수행 가능  
calendar.addEvent({  
 title: title,  
 start: startDate,  
 end: endDate  
});  
window.location.reload();  
},  
error: function(xhr, status, error) {  
 alert('일정 저장 중 오류가 발생했습니다: ' + error);  
 // 저장 실패 시 오류 메시지 표시  
}

});  
}

```
@PostMapping("/save-event")  
@ResponseBody  
public String saveEventToDB(RegistCalendarCommand regCalCommand) throws Exception{  
    // eventData 객체로부터 데이터를 추출하여 CalendarVO 객체에 설정  
    CalendarVO calendarVO = regCalCommand.toVO();  
  
    // CalendarService를 사용하여 DB에 일정 저장  
    projectService.calInsert(calendarVO);  
  
    return "일정이 성공적으로 저장되었습니다.";  
}
```

PORTFOLIO

2023.05.17

THANK YOU!

손규진

One Step

THANK YOU