

차량 렌트 예약 서비스

웹 프레임워크 프로그래밍 과제 2

소프트웨어전공 김두환

목차

차량 렌트 예약 서비스

01 프로젝트 개요

02 컴포넌트 설명

03 DI 설정

04 AOP 설정

05 트랜잭션 설정


06 인터셉터 설정

07 실행 흐름


01 프로젝트 개요



예약 가능 차량 목록

차량이미지	모델명	분류	운행가능지역	차량예약
	그랜저	대형	서울	예약하기
	모닝	소형	서울	예약하기
				

차량 예약 확인

차량이미지	
차량번호	1234
모델명	그랜저
분류	대형
운행가능지역	서울
예약자	dhkim

[차량 목록 보기](#)

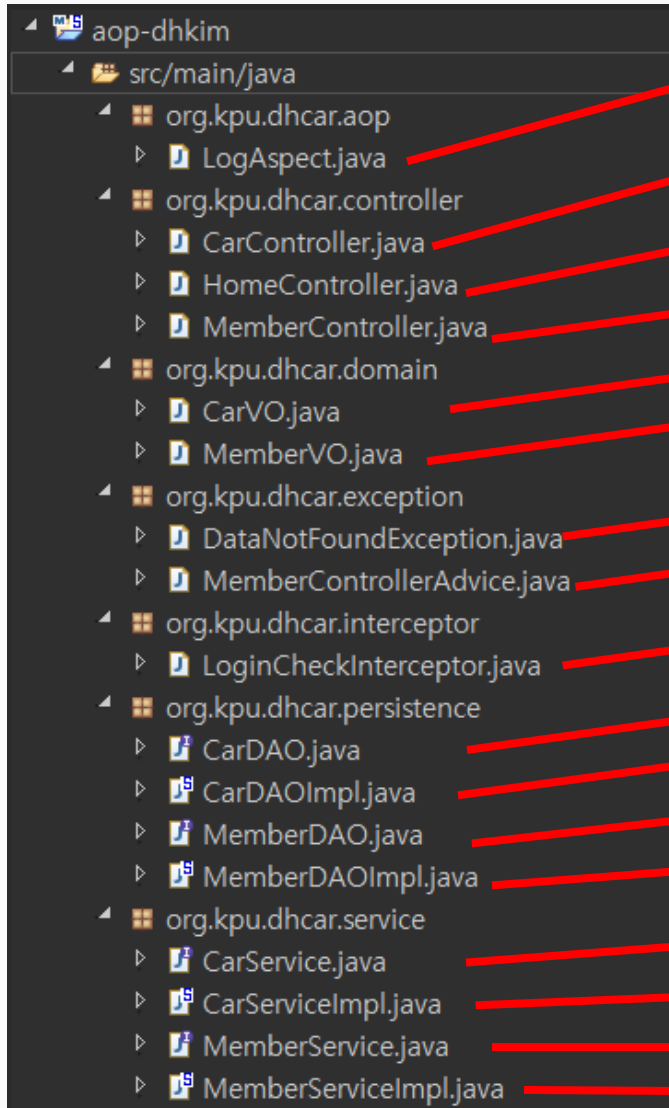
차량 목록 확인

예약 되어있지 않은 차량만
목록에서 확인 가능

예약하기

현재 로그인 한 사용자의
ID로 예약 등록

02 컴포넌트 설명



LogAspect.java : 로그를 출력하는 around어드바이스와 pointcut 관리

CarController.java : 차량 관리 컨트롤러

HomeController.java : 메인화면(로그인화면) 컨트롤러

MemberController.java : 회원 관리 컨트롤러

CarVO.java : 차량 정보 도메인 클래스

MemberVO.java : 회원 정보 도메인 클래스

DataNotFoundException.java : 데이터 없을 때 예외 처리 클래스

MemberControllerAdvice.java : 예외 처리 공통 클래스

LoginCheckInterceptor.java : 로그인 세션을 관리하는 인터셉터

CarDAO.java : 차량 정보 데이터 액세스 인터페이스

CarDAOImpl.java : 차량 정보 데이터 액세스 구현 클래스

MemberDAO.java : 회원 정보 데이터 액세스 인터페이스

MemberDAOImpl.java : 회원 정보 데이터 액세스 구현 클래스

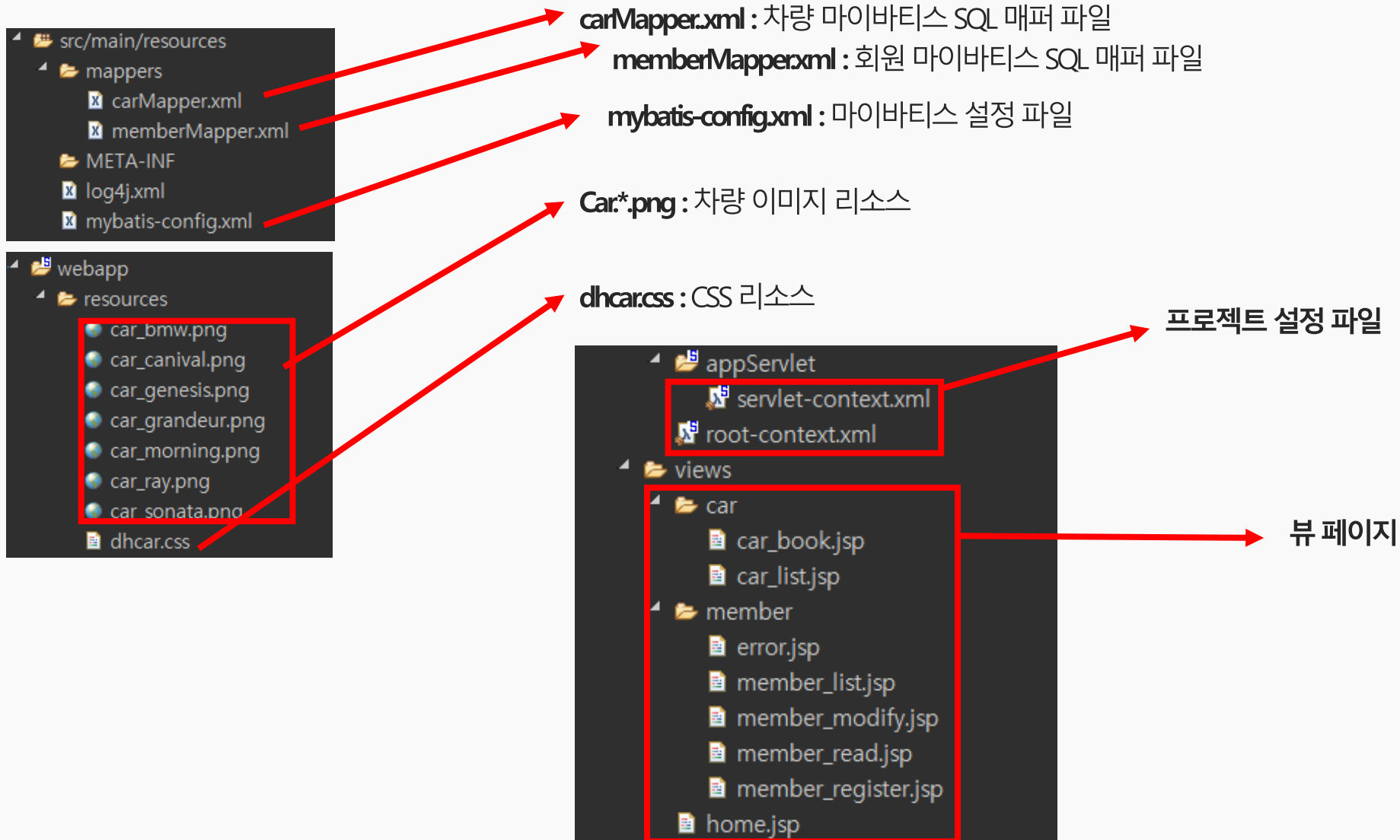
CarService.java : 차량 정보 서비스 인터페이스

CarServiceImpl.java : 차량 정보 서비스 구현 클래스

MemberService.java : 회원 정보 서비스 인터페이스

MemberServiceImpl.java : 회원 정보 서비스 구현 클래스

02 컴포넌트 설명

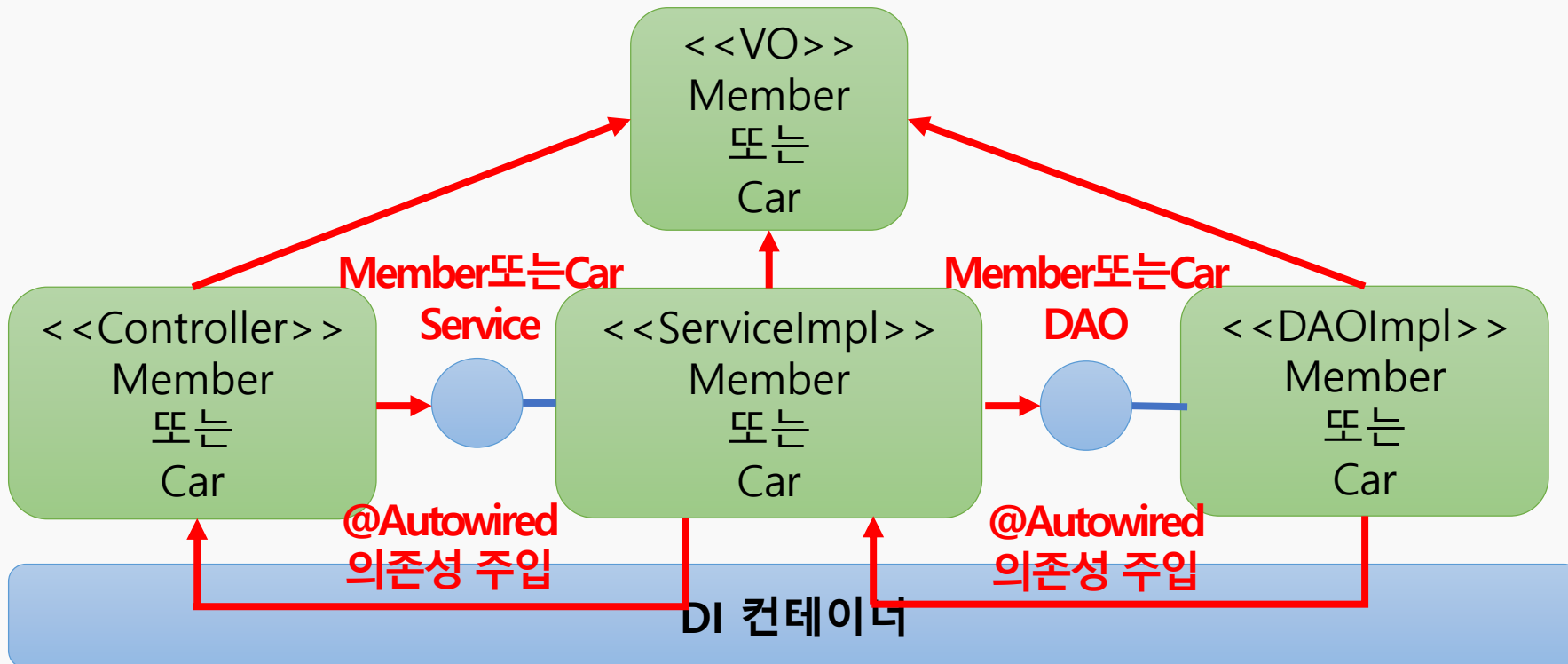


03 DI 설정

```
<!-- DI설정을 위한 패키지의 애노테이션 적용된 빈 자동 등록 및 @Autowired 활성화 by dhkim -->
<context:component-scan base-package="org.kpu.dhcar.persistence"/>
<context:component-scan base-package="org.kpu.dhcar.service"/>
<context:component-scan base-package="org.kpu.dhcar.aop"/>
```

rootContext.xml

DI 설정 및 의존성 주입
애노테이션 활성화를 위해
<context:component-scan>
사용



각 인터페이스를 통한 참조
결합도를 낮추기 위함

@Autowired 애노테이션 설정

애노테이션 설정 방식으로
의존성 주입하고,
별도 DI 컨테이너에서 관리

04 AOP 설정



```
@Component
@Aspect
public class LogAspect {
    private static final Logger logger = LoggerFactory.getLogger(LogAspect.class);

    @Around(
        "execution(* org.kpu.dhcar.*Controller.*(..))"
        + " or execution(* org.kpu.dhcar.service.*Impl.*(..))"
        + " or execution(* org.kpu.dhcar.persistence.*Impl.*(..)) "
    )
    public Object printLog(ProceedingJoinPoint pjp) throws Throwable {
        long startTime = System.currentTimeMillis();
        Object result = pjp.proceed();

        String type = pjp.getSignature().getDeclaringTypeName();
        String packageName = "";

        if(type.contains("Controller")) {
            packageName = "Controller : ";
        }
        else if(type.contains("Service")) {
            packageName = "Service : ";
        }
        else if(type.contains("DAO")) {
            packageName = "Persistence : ";
        }

        long endTime = System.currentTimeMillis();

        logger.info(packageName + type + "." + pjp.getSignature().getName() + " 메소드 호출");
        logger.info("실행된 시간 : " + (endTime - startTime) + "ms");
        logger.info("-----");

        return result;
    }
}
```

Aspect 정의 클래스

LogAspect.java

- Aspect 선언

@Aspect

- DI 컴포넌트 선언

@Component

- Advice 선언

@Around

- Pointcut 선언

Controller.(..): 컨트롤러 패키지의 모든 클래스 메소드

Impl.(..): 서비스 패키지의 모든 구현 클래스 메소드

Impl.(..): 퍼시스턴스 패키지의 모든 구현 클래스 메소드

- JoinPoint 실행된 해당 클래스 이름 추출

- 문자열로 패키지 구분

05 트랜잭션 설정



```
@Transactional(  
    propagation = Propagation.REQUIRED,  
    isolation = Isolation.SERIALIZABLE,  
    timeout = 10  
)  
@Override  
public void updateCar(CarVO car) throws Exception {  
    carDAO.update(car);  
}
```

차량과 회원 업데이트 메소드 적용

예약과 정보 수정 트랜잭션 처리 중 에러가 발생하면 트랜잭션을 롤백한다.

격리 상태 SERIALIZABLE

한 차량을 예약 트랜잭션 진행 중 다른 트랜잭션이 접근하지 못 하게 한다.

```
@Transactional(  
    propagation = Propagation.REQUIRED,  
    isolation = Isolation.READ_COMMITTED,  
    timeout = 10  
)  
@Override  
public void updateMember(MemberVO member) throws Exception {  
    memberDAO.update(member);  
}
```


06 인터셉터 설정

```
public class LoginCheckInterceptor extends HandlerInterceptorAdapter{

    // 컨트롤러보다 먼저 수행되는 메소드
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler
        throws Exception {

        HttpSession session = request.getSession();
        Object obj = session.getAttribute("login");
        // 로그인 되어있지 않은 상태이므로 로그인 페이지로 리다이렉트하고, false로 설정하여 컨트롤러로 넘어가지 않음
        if(obj == null) {
            response.sendRedirect("/dhcar");
            return false;
        }
        // true로 설정하여 컨트롤러로 넘어감
        return true;
    }
}
```

컨트롤러 이전 수행 메소드

DispatcherServlet이 컨트롤러 호출 하기 전 처리

로그인 세션 관리

"login" 세션 확인하여 null이면 메인 페이지로 리다이렉트 하고 false 리턴하여 컨트롤러 실행되지 않게 한다.

```
<beans:bean id="LoginCheckInterceptor" class="org.kpu.dhcar.interceptor.LoginCheckInterceptor">
</beans:bean>
```

인터셉터 빈 등록

sevletContext.xml 에 설정

```
<interceptors>
    <interceptor>
        <mapping path="/dhcar/**"/>
        <beans:ref bean="LoginCheckInterceptor"/>
    </interceptor>
</interceptors>
```

인터셉터 적용 범위 설정

/dhcar/** 설정하여 모든 페이지에 적용

07 실행 흐름



DH-Car 로그인

LOGIN ID	<input type="text" value="ID를 입력하세요."/>
PASSWORD	<input type="password" value="PASSWORD를 입력하세요"/>

로그인

회원가입

메인화면_로그인화면

- 로그인을 해야 차량 예약 가능하므로 메인 화면이다.
- 로그인 실패 시 메인화면으로 리다이렉트 한다.

07 실행 흐름



예약 가능 차량 목록

차량이미지	모델명	분류	운행가능지역	차량예약
	그랜저	대형	서울	예약하기
	모닝	소형	서울	예약하기
				



차량 예약 확인

차량이미지	
차량번호	1234
모델명	그랜저
분류	대형
운행가능지역	서울
예약자	dhkim

[차량 목록 보기](#)

차량예약

- 로그인하면 차량 목록 나옴
- 예약하기 누르면 로그인 된 아이디가 자동으로 예약자 아이디에 DB저장

num	carname	category	area	image	booker
1234	그랜저	대형	서울	car_grandeur	dhkim
2234	모닝	소형	서울	car_morning	NULL
3234	쏘나타	중형	인천	car_sonata	NULL
4234	카니발	승합	경기	car_canival	NULL
5234	BMW	수입	부산	car_bmw	NULL

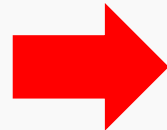
07 실행 흐름

DH-Car-회원가입

ID	<input type="text" value="hs"/>
PASSWORD	<input type="password" value=".."/>
NAME	<input type="text" value="허훈식"/>
AGE	<input type="text" value="30"/>
LICENSE	<input type="text" value="1종보통"/>
MOBILE	<input type="text" value="010-1234-1234"/>
EMAIL	<input type="text" value="hs@kpu.co.kr"/>

보내기

다시 작성



DH-Car-회원목록

[학생 등록](#)

ID	PWD	MEMBERNAME	AGE	LICENSE	MOBILE	EMAIL	
abc	abc	이영진	28	2종보통	010-1234-1234	dove209@naver.com	정보 보기 정보 수정
dhkim	1234	김두환	27	1종보통	010-4181-9965	dhkim@naver.com	정보 보기 정보 수정
dhkim3	1234	김세환	37	2종보통	010-4181-9966	dhkim3@naver.com	정보 보기 정보 수정
dhkim4	1234	김네환	47	2종오토	010-4181-9967	dhkim4@naver.com	정보 보기 정보 수정
dhkim5	1234	김다환	57	1종대형	010-4181-9968	dhkim5@naver.com	정보 보기 정보 수정
dhkim6	1234	김여환	67	국제면허증	010-4181-9969	dhkim6@naver.com	정보 보기 정보 수정
hs	hs	허훈식	30	1종보통	010-1234-1234	hs@kpu.co.kr	정보 보기 정보 수정

회원가입

- 회원가입 후 목록, 정보 확인
- DB에 저장

DH-Car-회원정보

ID	hs
PASSWORD	hs
NAME	허훈식
AGE	30
LICENSE	1종보통
MOBILE	010-1234-1234
EMAIL	hs@kpu.co.kr

[학생 목록 보기](#)

id	passwd	membername	age	license	mobile	email
dhkim4	1234	김네환	47	2종오토	010-4181-9967	dhkim4@naver.com
dhkim5	1234	김다환	57	1종대형	010-4181-9968	dhkim5@naver.com
dhkim6	1234	김여환	67	국제면허증	010-4181-9969	dhkim6@naver.com
hs	hs	허훈식	30	1종보통	010-1234-1234	hs@kpu.co.kr

07 실행 흐름

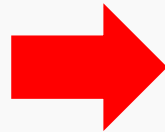


DH-Car 회원정보수정

ID	<input type="text" value="hs"/>
PASSWORD	<input type="text" value="hs"/>
NAME	<input type="text" value="허훈식"/>
AGE	<input type="text" value="20"/>
LICENSE	<input type="text" value="1종보통"/>
MOBILE	<input type="text" value="010-1234-1234"/>
EMAIL	<input type="text" value="hs@kpu.co.kr"/>

최종 수정

다시작성



DH-Car 회원정보

ID	hs
PASSWORD	hs
NAME	허훈식
AGE	20
LICENSE	1종보통
MOBILE	010-1234-1234
EMAIL	hs@kpu.co.kr

[학생 목록 보기](#)

회원정보 수정

- 수정하고 회원정보에서 확인
- DB에 수정된 데이터 확인

id	passwd	membername	age	license	mobile	email
dhkim4	1234	김네환	47	2종오토	010-4181-9967	dhkim4@naver.com
dhkim5	1234	김다환	57	1종대형	010-4181-9968	dhkim5@naver.com
dhkim6	1234	김여환	67	국제면허증	010-4181-9969	dhkim6@naver.com
hs	hs	허훈식	20	1종보통	010-1234-1234	hs@kpu.co.kr

프로젝트 후기



일정 관리의 어려움

다른 프로젝트도 마찬가지로 프로젝트 진행하면서
실력도 중요하지만 그만큼 일정 관리가 중요하다는 것을
깊이 깨닫는 계기가 되었다.

프레임워크 기술 적용 분야의 이해 부족

MyBatis 같은 경우는 DB 연동에 적용한다는 것을 알고 있었지만,
AOP나 인터셉터와 같은 개념은 어느 부분에 어떻게 응용할 지에
대해서 앞으로 연구가 더 필요하다고 느꼈다.

더 완성된 웹서버 구축에 대한 욕심

지난 학기 웹프로그래밍을 들었을 때보다 더 실무적인 기술을 배운
것 같은 느낌이다.
따라서 종강 후에 시간을 투자하여 지금보다 훨씬 완성된 웹서버를
구축하고 스프링부트도 공부할 계획이다.



감사합니다.

소프트웨어전공 김두환