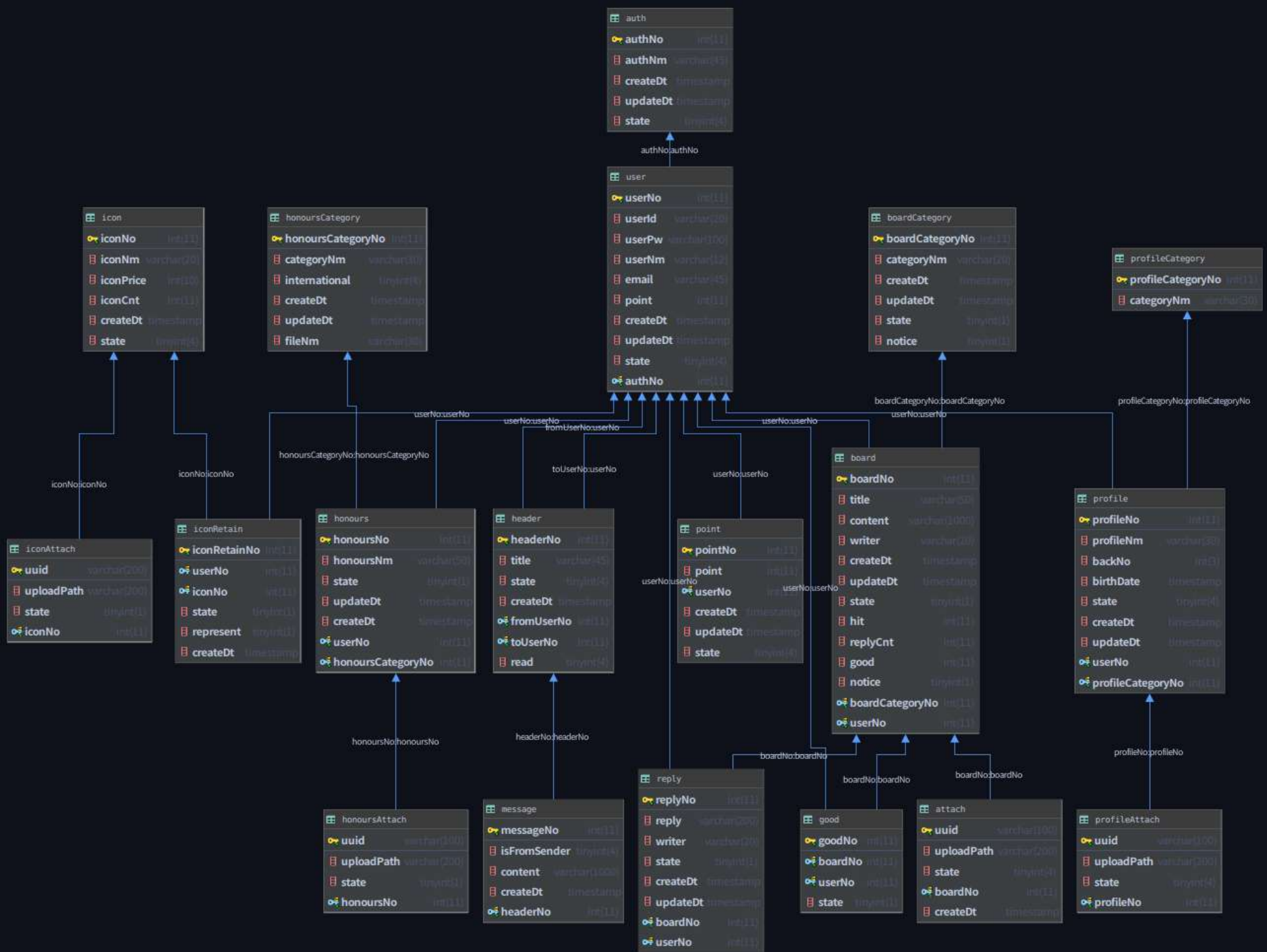
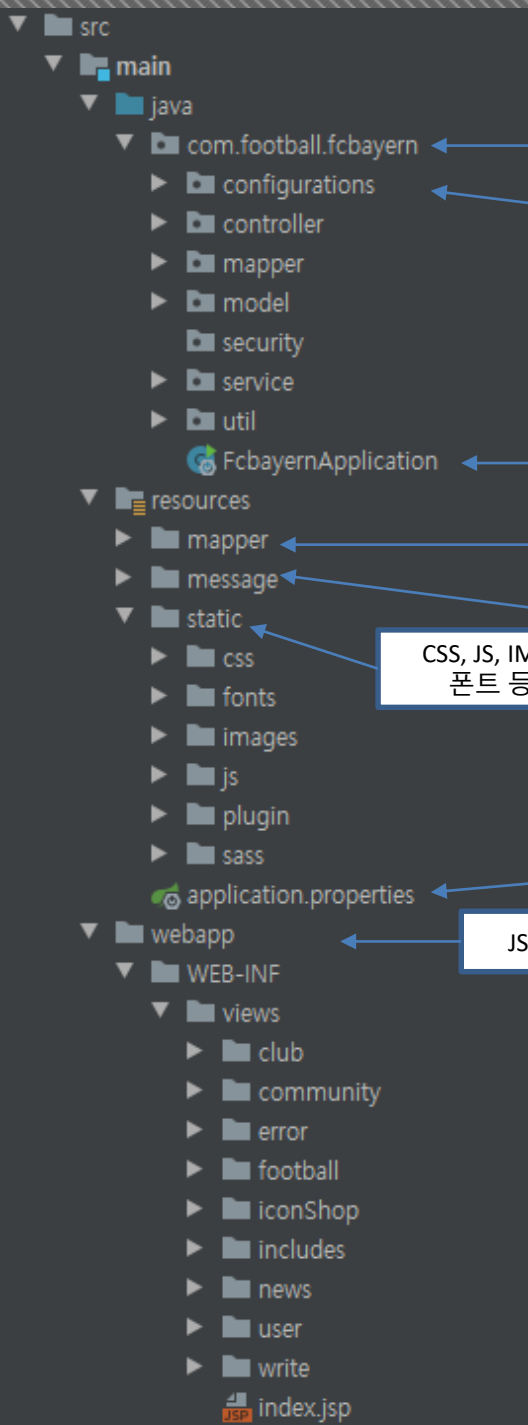


Fc Bayern



Bayern Munich





자바 코드의 경로

자바 메타 정보

메인 메서드

Mapper 파일

언어 설정에 따른
정보

CSS, JS, IMG,
폰트 등

스프링 외부 설정 파일

JSP파일 경로



unich

A 태그에
마우스 올린
경우 커서와
색상 변경

모든
페이지
이동은 ajax
사용



뉴스

FC 바이에른 뮌헨의 뉴스를 모아 놓은
게시판

[더 보기](#)



축구

축구와 관련된 이야기를 나누는 게시
판

[더 보기](#)



자유게시판

자유로운 주제를 가지고 이야기하는
게시판

[더 보기](#)



멀티 미디어

>다양한 축구 영상과 사진을 감상하는
게시판

[더 보기](#)



공지사항

가입 후 읽어 봐야할 공지사항에 대해
모아 놓은 게시판

[더 보기](#)



아이콘샵

아이콘을 구매 할 수 있도록 만들어
놓은 곳

[더 보기](#)

About FCB

Fc Bayern 뽀 사이트 업
니다.

클럽

팀
선수
타이틀

뉴스

뉴스

축구

축구

커뮤니티

자유게시판
멀티 미디어
공지사항

아이콘샵

아이콘샵
보관함



```

$.setLanguage').click(function () {
    const getLanguage = $(this).closest("a").attr('id');
    window.location.replace( url: "index?lang=" + getLanguage);
});

$('.teamLink').click() => {
    $(".content").load("/team?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
    history.pushState( data: {data: "/team"}, title: null, url: "/team?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
});

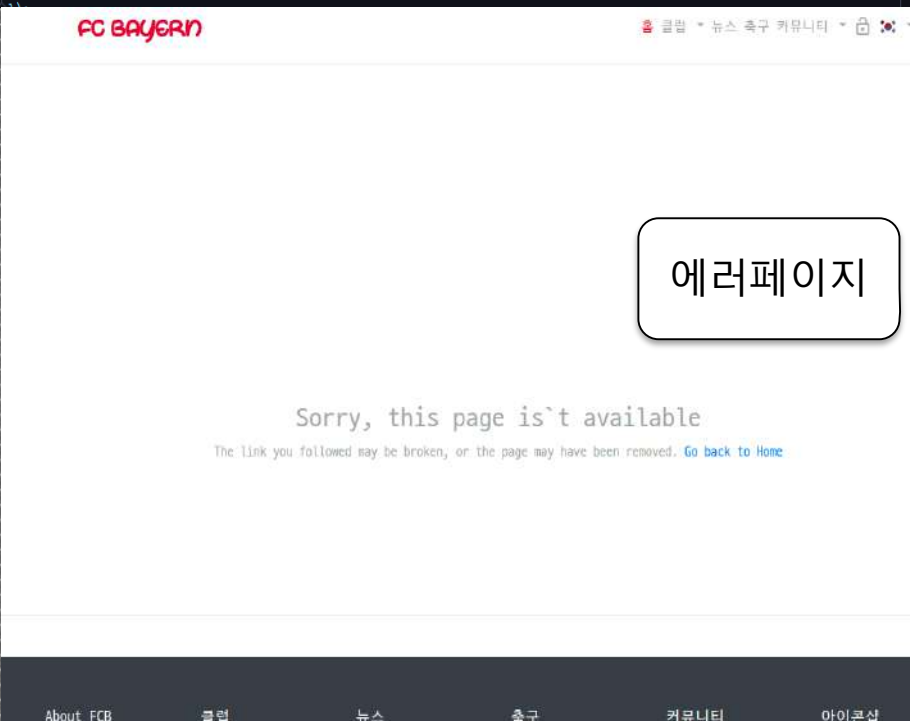
$('.playerLink').click() => {
    $(".content").load("/player?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
    history.pushState( data: {data: "/player"}, title: null, url: "/player?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
});

$('.honoursLink').click() => {
    $(".content").load("/honours?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
    history.pushState( data: {data: "/honours"}, title: null, url: "/honours?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
});

$('.newsLink').click() => {
    $(".content").load("/news?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
    history.pushState( data: {data: "/news"}, title: null, url: "/news?lang=" + getCookie( name: 'APPLICATION_LOCALE'));
});

```

Ajax로만 작성되어 있어 history.pushState를
이용하여 뒤로 가기 기능 구현



```

$(window).on('popstate', function (event) {
    const data = event.originalEvent.state;

    //push 된 history 없는 경우 처리하기 위하여
    if (data === null) {
        location.href = "/index";
    } else {
        $.get(data.data + "?lang=" + getCookie( name: 'APPLICATION_LOCALE'), {
            boardCategoryNo: data.boardCategoryNo,
            boardNo: data.boardNo,
            type: data.type,
            keyword: data.keyword,
            pageNum: data.pageNum
        }, (result) => {
            $(".content").html(result);
        });
    }
});

```

이벤트 감지하여 페이지 로드

```

@Controller
public class CustomErrorController implements ErrorController {

    @GetMapping("/error")
    public String errorPage(HttpServletRequest httpServletRequest) {
        Object status = httpServletRequest.getAttribute(RequestDispatcher.ERROR_STATUS_CODE);

        if (status != null) {
            int statusCode = Integer.parseInt(status.toString());

            if (statusCode == HttpStatus.NOT_FOUND.value()) {
                return getErrorPath() + "/error404";
            }

            if (statusCode == HttpStatus.FORBIDDEN.value()) {
                return getErrorPath() + "/error403";
            }

            if(statusCode == HttpStatus.BAD_REQUEST.value()){
                return getErrorPath() + "/error";
            }
        }

        return getErrorPath()+"/error";
    }
}

```

에러 담기 위한 객체

Spring security를 이용해 iconShop, Write, read 등의 페이지에 인증된 회원만 접근 할 수 있도록 설정하였으며, 커스텀 로그인 페이지를 이용하도록 만들

Sign In

Sign in

ID

Password

☐ 기억하기

☐ 가입하기

로그인

```

@EnableWebSecurity
//WebSecurityConfigurer 인스턴스를 편리하게 생성하기 위한 클래스
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

//Spring Security 에서 제공하는 비밀번호 암호화 객체
//Service에서 비밀번호 암호화 할 수 있도록 Bean으로 등록
@Bean
public PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder(); }

@Override
//WebSecurity를 이용하여 FilterChainProxy 을 생성
public void configure(WebSecurity webSecurity) throws Exception {
//Spring security 가 무시하여 static directory 하위 파일 목록은 항상 공개
webSecurity.ignoring().antMatchers( ...antPatterns: "/css/**", "/js/**", "/images/**", "/fonts/**", "*/plugin/**", "*/sass/**");
}

@Override
//HttpSecurity를 통해 HTTP 요청에 대한 보안구현
protected void configure(HttpSecurity httpSecurity) throws Exception {
//authorizeRequests
//antMatchers 를 통하여 특정 경로를 지정하여 역할에 맞는 설정을 걸어준다.
//csrf().disable() 토큰 사용하지 않으려면
httpSecurity.authorizeRequests().expression(new AuthorizationConfigurer().ExpressionInterceptorRegistry
//백미지 권한 설정
.antMatchers( ...antPatterns: "/write/**").authenticated()
.antMatchers( ...antPatterns: "/iconStorage/**").authenticated()
.antMatchers( ...antPatterns: "/read/**").authenticated()
.antMatchers( ...antPatterns: "/iconShop").authenticated()
.antMatchers( ...antPatterns: "/*").permitAll()
.antMatchers( "/*").permitAll()
.and().httpSecurity
//로그인 설정
//form 기반으로 인증을 하도록하여 로그인 정보는 기본적으로 HttpSession을 이용한다.
.formLogin().formLoginConfigurer(httpSecurity)
//커스텀 로그인 폼을 사용하기 위해서는 LoginPage()에서 사용
.loginPage("/signInModal")
//loginProcessingUrl()과 파라미터값으로 커스텀 로그인 form의 action 경로가 일치해야 한다.
.loginProcessingUrl("/signIn")
//실패시
.failureUrl("/login-error")
}

```

```

@Override
public UserDetails loadUserByUsername(String userId) throws UsernameNotFoundException {
HttpServletRequest httpServletRequest = ((ServletRequestAttributes) RequestContextHolder.getRequestAttributes()).getRequest();
String userPw = httpServletRequest.getParameter( name: "userPw");
UserModel userModel = findById(userId);
//matches 를 이용하여 비밀번호 값은 userPw와 encode 된 userPw과 일치하는지 확인
if (userModel != null && passwordEncoder.matches(userPw, userModel.getUserPw())) {
//CustomUserModel에서 User 상속받아서 생성자 통해서 변경
return new CustomUserModel(userModel.getUserId(), userModel.getUserPw(), auth(userModel.getAuthNm()), userModel.getUserNo(), userModel.getEmail(), userModel.getAuthNo(), userModel.getAuthNm());
}

//유저 정보와 맞으면 UsernameNotFoundException 예외 처리한다.
throw new UsernameNotFoundException(userId);
}

//Data 사본시 예외
@Getter
@Setter
@Override
public class CustomUserModel extends User {
private int userNo;
private String userId;
private String userName;
private String email;

private int authNo;
private String authNm;

public CustomUserModel(String username, String password, Collection<? extends GrantedAuthority> authorities, int userNo, String userId, String email, int authNo, String authNm) {
super(username, password, authorities);
this.userNo = userNo;
this.userId = userId;
this.userName = userName;
this.email = email;
this.authNo = authNo;
this.authNm = authNm;
}
}

```

세션에 원하는 값 추가하기
위해 User 상속받아서 확장

Sign Up

Sign Up

test

비밀번호를 확인해 주세요.

테스트

test@test.com

가입하기

```

"#signUpBtn").click(() => {
  const token = $("meta[name='_csrf']").attr("content");
  const header = $("meta[name='_csrf_header']").attr("content");
  const signUpData = {
    userId: $("#userId").val(),
    userPw: $("#userPw").val(),
    userNm: $("#userNm").val(),
    email: $("#email").val(),
  };

  if (signUpData.userPw === $("#userPwCheck").val()) {
    $.ajax({
      type: "post",
      url: "/user/signUp",
      data: signUpData,
      beforeSend: (xhr) => {
        xhr.setRequestHeader(header, token);
      },
      success: () => {
        iziToast.success({icon: 'icon-person'...});
      },
      error: () => {
        iziToast.error({title: 'Fail'...});
      }
    });
  } else if (signUpData.userPw !== $("#userPwCheck").val()) {
    $("#wrongPw").html("비밀번호를 확인해 주세요.");
  } else if (signUpData.userNm.length < 2) {
    $("#wrongNm").html("2글자 이상으로 입력해주세요.");
  }
});

```

유효성
체크

스를 모아 놓은



축구

축구와 관련된 이야기를 나누는 게시
판

더 보기



공지사항

이전을 감상하는

가입 후 읽어 봐야 할 공지사항에 대해
모아 놓은 게시판

더 보기

```

@AllArgsConstructor
public class UserServiceImpl implements UserService, UserDetailsService {

  private UserMapper userMapper;
  private PasswordEncoder passwordEncoder;

  @Transactional
  @Override
  public int signUp(UserModel userModel) {
    //중복아이디 체크
    if (checkNull(userModel.getUserId()) > 0) {
      return 0;
    }
    //암호화
    userModel.setUserPw(passwordEncoder.encode(userModel.getUserPw()));
    return userMapper.signUp(userModel);
  }

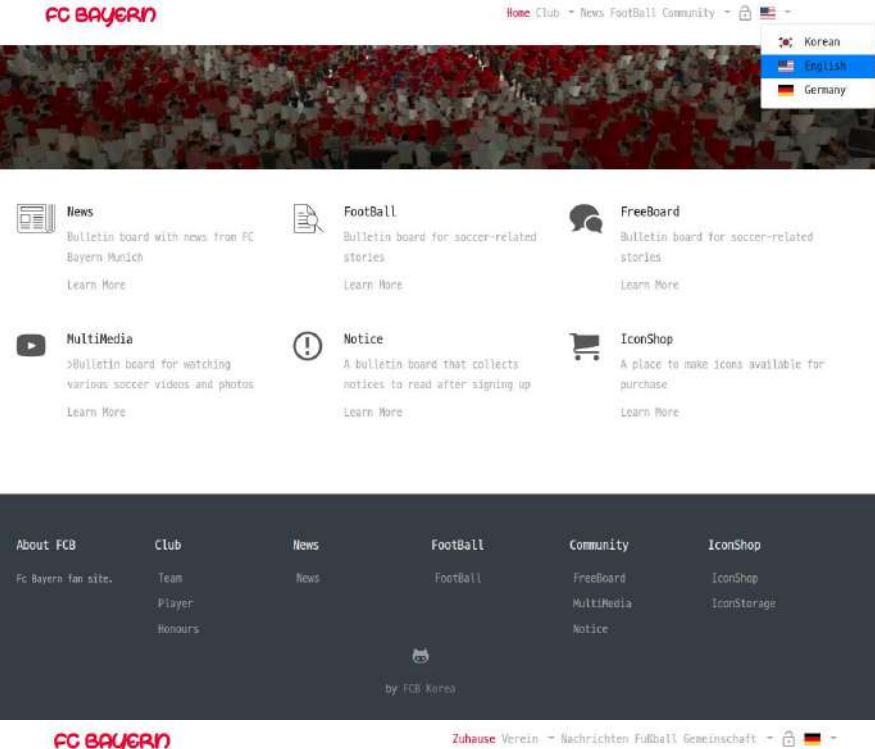
  @Override
  public UserModel findById(String userId) { return userMapper.findById(userId); }

  @Override
  public int getPoint(int userNo) { return userMapper.getPoint(userNo); }

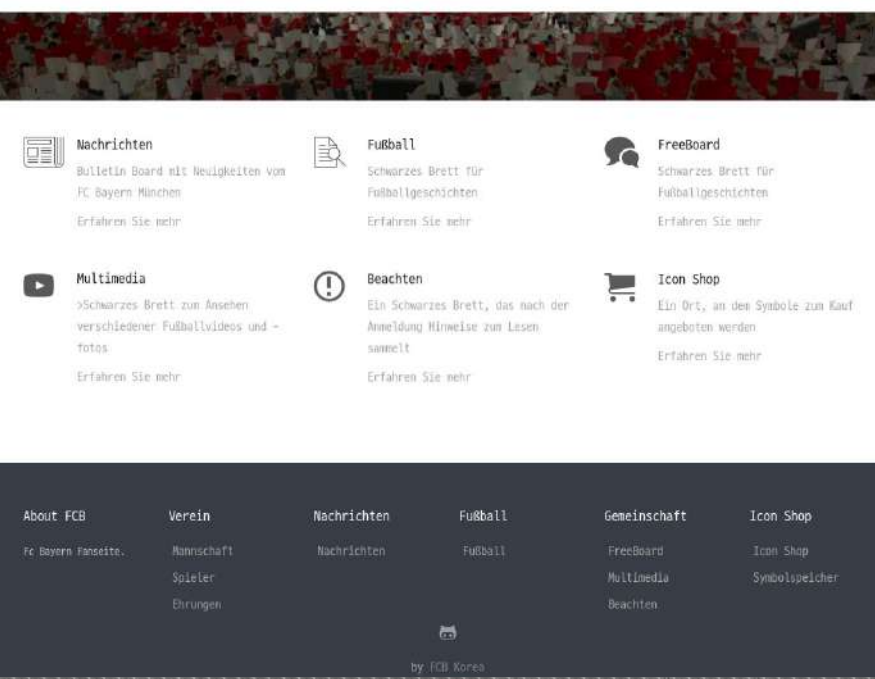
  @Override
  public int checkNull(String userId) { return userMapper.checkNull(userId); }
}

```

아이디
중복 체크BcryptEncoder
암호화



message.properties로
언어설정 후, 쿠키를 통해
현재 언어를 읽고
spring:message 를 통해
사용



@Configuration

```
public class LocaleConfiguration implements WebMvcConfigurer {
```

```
@Bean(name = "localeResolver")
```

```
public LocaleResolver getLocaleResolver() {
```

```
CookieLocaleResolver cookieLocaleResolver = new CookieLocaleResolver();
cookieLocaleResolver.setDefaultLocale(Locale.KOREAN);
cookieLocaleResolver.setCookieName("APPLICATION_LOCALE");
cookieLocaleResolver.setCookieMaxAge(60 * 60);
```

```
return cookieLocaleResolver;
```

```
}
```

쿠키를 통해 기본 언어와 시간 설정

```
@Bean(name = "messageSource")
```

```
public MessageSource getMessageResource() {
```

```
ReloadableResourceBundleMessageSource messageSource = new ReloadableResourceBundleMessageSource();
messageSource.setBasename("classpath:message/message");
messageSource.setDefaultEncoding("UTF-8");
messageSource.setCacheSeconds(10);
```

Message.properties
경로설정

```
return messageSource;
```

```
}
```

```
@Override
```

```
public void addInterceptors(InterceptorRegistry registry) {
```

```
LocaleChangeInterceptor localeChangeInterceptor = new LocaleChangeInterceptor();
localeChangeInterceptor.setParamName("lang");

registry.addInterceptor(localeChangeInterceptor).addPathPatterns("/*");
```

```
}
```

인터셉터 이용해 파라미터에 ?lang= 추가



클럽 소개 페이지



FC Bayern is a company but in the first instance it is a club with the membership at its core. With more than 270,000 members, FC Bayern is the biggest club in the world. From the earliest days, the club has met its responsibilities to the community by developing young talent and emphasising grassroots sport. This is by no means restricted to football. FC Bayern teams actively compete in basketball, handball, chess, veterans' football, sports schools, table tennis and gymnastics, some at elite national levels. And with more than 100 referees, the club boasts Europe's biggest refereeing section. FC Bayern as a club is firmly anchored in grassroots sport.

Following on from the Grünwald Stadium and later the Olympic Stadium, the stunning ground in the north of Munich is the third home venue in Bayern's long Bundesliga

every season with all 71,000 available seats and standing places being sold out for a 17 Bundesliga home games. Bayern's incredible attendance record invariably includes 68,000 full houses for European fixtures (the capacity is reduced by 3,000 in the all seat configuration). As a rule, shortly after the Bundesliga fixtures are announced at the beginning of July, more than half of Bayern's league games in the Allianz Arena sell out at once. What brings the fans streaming to Fröttmanning in their tens of thousands? The stars of the mighty Bayern Munich of course. But the fans can also expect to be well looked after, both before and after the game. 6,000 m² of catering facilities take care of culinary requirements. Europe's biggest multi-storey car park provides nearly 10,000 parking spaces. Since May 2012 there has been another compelling reason for fans and families to flock to the stadium in droves: the FC Bayern Erlebniswelt, housed at the Allianz Arena and already established as a must-see attraction for thousands of the five million visitors to Munich every year. Germany's biggest club museum tells the absorbing story of the club in an interactive installation covering more than 3,000 m².



The exhibition on Level 3 of the Allianz Arena features much more than countless fascinating exhibits, including trophies, and boots and shirts worn by Bayern greats past and present. The 113-year history of one of Europe's most famous and successful clubs is retold using an intoxicating blend of facts and infotainment, modern and emotionally moving, but at the same time entirely authentic. Trophies, cups and more than 500 exhibits are displayed using award-winning media technology and interactive

선수 소개 페이지

12



中文 - 韓 → * Tied Takwatt, Nikita Sato, Nguyen Pham, Ian Gervais, Corinne Jordan, Thang, Sam Kirtano, Jerome Gervais, Isaac Hui, Kinsley Goss, Michael Gossard, Gary King, Hans Piers, Edgar Austin
 中文 - 韓 → * Christian Filippi, Peter Jay, Pulgite Gossard, Alpinus Davis, Robert Leonardini, Thomas Kitter, Russell Turner, David Albin, Terry Gossard, Isaac Hernandez, Tim Perish, Joshua Thayer, Sam Kirtano, & Charles Hoffman

전체	골키퍼	수비수	미드필더	포워드
----	-----	-----	------	-----



DEFENDERS



MASCOTS ▾ AVATAR 등번호 2020. 12. 31. 오후 12:59 업로드

사진 파일을 드래그해 주세요.



선수 프로필 등록

```
public static String uploadFile(String uploadPath, String originalName, byte[] bytes) throws Exception{
```

```
    AwsS3Util awsS3Util = new AwsS3Util();
    final String bucketName = "woolution";
```

```
    UUID uuid = UUID.randomUUID();
```

```
    String saveName = "/" + uuid.toString() + "_" + originalName;
```

```
    String savedPath = calendarPath(uploadPath);
```

```
    String uploadedFileName = (savedPath + saveName).replace(File.separatorChar, newChar: '/');
    //AwsS3Util fileUpload 메서드를 사용해 업로드
    awsS3Util.fileUpload(bucketName, fileName: uploadPath + uploadedFileName, bytes);
```

```
    return uploadedFileName;
```

```
private static String calendarPath(String uploadPath){
```

```
    Calendar calendar = Calendar.getInstance();
```

```
    String yearPath = File.separator + calendar.get(Calendar.YEAR);
```

```
    String monthPath = yearPath + File.separator + new DecimalFormat( pattern: "00").format( number: calendar.get(Calendar.MONTH) + 1);
```

```
    String datePath = monthPath + File.separator + new DecimalFormat( pattern: "00").format(calendar.get(Calendar.DATE));
```

```
    mkdir(uploadPath, yearPath, monthPath, datePath);
```

```
    return datePath;
```

현재 날짜 폴더 위해서

```
private AmazonS3 amazonS3;
```

```
public AwsS3Util() {
    AWSCredentials awsCredentials = new BasicAWSCredentials(accessKey, secretKey);
    ClientConfiguration clientConfiguration = new ClientConfiguration();
    clientConfiguration.setProtocol(Protocol.HTTP);
    this.amazonS3 = new AmazonS3Client(awsCredentials, clientConfiguration) {
    };
    amazonS3.setEndpoint("s3.ap-northeast-2.amazonaws.com");
}
```

s3 버킷에 이미지 저장

```
public void fileUpload(String bucketName, String fileName, byte[] fileData) throws FileNotFoundException {
    String filePath = (fileName).replace(File.separatorChar, newChar: '/');// 파일 구분자를 '/'으로 설정
    ObjectMetadata objectMetadata = new ObjectMetadata();

    objectMetadata.setContentLength(fileData.length); //메타데이터 설정 128kb -> 파일 크기만큼으로 변경
    ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(fileData);

    amazonS3.putObject(bucketName, filePath, byteArrayInputStream, objectMetadata);
}
```

```
public String getFileURL(String bucketName, String fileName) {
    String imgName = (fileName).replace(File.separatorChar, newChar: '/');
    return amazonS3.generatePresignedUrl(new GeneratePresignedUrlRequest(bucketName, imgName)).toString();
}
```

s3 버킷에서 이미지 가져오기

배열화 처리를 통해서 여러 개 폴더 생성

```
private static void mkdir(String uploadPath, String... paths ){ //... 동일한 파라미터 여러개 받을 경우 자동으로 배열 처리

    if (new File(paths[paths.length - 1]).exists()) {
        return;
    }

    for (String s: paths){
        File dirPath = new File( pathname: uploadPath + s);

        if(!dirPath.exists()){
            dirPath.mkdir();
        }
    }
}
```


타이틀

National Honours

 X 29 +
German Championship

 X 19 +
DFB German Cup

 X 7 +
German Supercup

 X 6 +
League Cup

우승컵
소개

International Honours

 X 5 +
Champions League

 Champions League Winner 2013

 Champions League Winner 2001

 European Cup Winner 1976

콜랩스와 캐리셀 사용

 X 1 +
UEFA Cup

 X 1 +
UEFA Cup

뉴스



test

test
2020-03-29 | 오전 7:29



이적시장 종료

무명임으로 끝났다.....
2020-03-19 | 오전 2:55



코망 부상

왼쪽 허벅지 부상4주치
2020-03-17 | 오후 11:38



레반도프스키 득점

오예!~~~~~
2020-03-17 | 오후 10:48

사진이 없는 경우 디폴트
이미지

뉴스

1

제목 검색

등록

```
success: (thumb) => {  
  let thumbnail = "images/news/news.webp";  
  if (thumb !== "") {  
    // /aws/getImg?fileName= + attachFilter[0].uuid + "&directory=" + attachFilter[0].uploadPath + "  
    thumbnail = "/aws/getImg?fileName=" + thumb.uuid + "&directory=" + thumb.uploadPath;  
  }  
}
```


자유게시판

제목	작성자	날짜	조회	추천
공지 도배성글 작성하지 마세요	어드민	2020-03-28	2	0
공지 가입 후 필독	어드민	2020-03-27	5	0
아이콘 [10]	유저2	2020-03-28	20	2
테스트 [0]	관리자	2020-03-28	9	1
우승~ [0]	부자	2020-03-28	5	0
오늘 잊었던일 [0]	유저2	2020-03-28	3	-1
안녕하세요 [0]	유저	2020-03-28	2	0
테스트 게시물입니다. [3]	테스터	2020-03-26	17	0
테스트 [0]	테스터	2020-03-26	3	0
도배성 글 작성하지 마세요 [0]	admin	2020-03-19	22	0

1 2 3 4

제목

검색

등록

/유저 아이콘 불러오기

```

function getUserIcon(userNo) {
    let getIconImg = '';

    const getIconNo = $.ajax({
        url: "/iconShop/checkRepresent/" + userNo,
        type: "get",
        async: false,
    });

    if (Number(getIconNo.responseText) !== 0) {
        $.ajax({
            url: "/iconShopAttach/getAttachImg/" + getIconNo.responseText,
            type: "get",
            async: false,
            success: (result) => {
                getIconImg = userIcon(result)
            }
        });
    } else {
        getIconImg = ""
    }

    return getIconImg;
}

```

이미지 가져오는 ajax가 먼저 작동하는 현상 막기 위해 동기화처리

설정된 아이콘 없는 경우 처리

```

select boardCategoryNo, categoryNm, createDt, updateDt, state, notice
from boardCategory;

</select>

<!-- 게시물 등록 -->
<insert id="insertInfo">
    insert into board (title, content, writer, createDt, updateDt, state, notice, userNo, boardCategoryNo)
    VALUES (#{title}, #{content}, #{writer}, now(), now(), 1, #{notice}, #{userNo}, #{boardCategoryNo})
</insert>

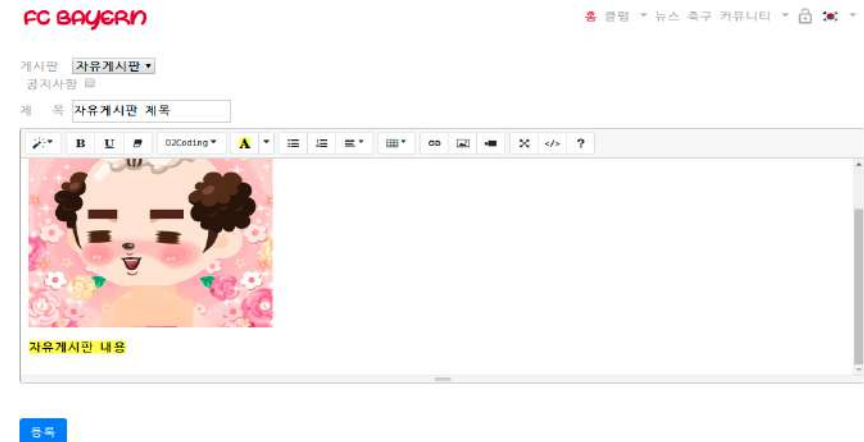
<!-- 글 리스트 가져오는것 -->
<select id="infoList" resultType="com.football.fcbayern.model.BoardModel">
    select boardNo, title, content, writer, createDt, updateDt, hit, good, replyCnt, userNo, notice, boardCategoryNo
    from board
    where state = 1
    and boardCategoryNo = #{boardCategoryNo}
    <include refid="criteria"/>
    order by notice desc, createDt desc
    limit #{pageStart}, #amount;
    # amount 10 으로 고정 지정
</select>

<!-- 검색시 필터링 -->
<sql id="criteria">
    <if test="type != null">
        <if test="type == 't'.toString()">
            and title like CONCAT('%', #{keyword}, '%')
        </if>
        <if test="type == 'c'.toString()">
            and content like CONCAT('%', #{keyword}, '%')
        </if>
        <if test="type == 'w'.toString()">
            and userNo like CONCAT('%', #{keyword}, '%')
        </if>
        <if test="type == 'tc'.toString()">
            and ( title like CONCAT('%', #{keyword}, '%') OR
            content
            like
            CONCAT('%', #{keyword}, '%'))
        </if>
        <if test="type == 'tw'.toString()">
            and ( userNo like CONCAT('%', #{keyword}, '%') OR
            content
            like
            CONCAT('%', #{keyword}, '%'))
        </if>
        <if test="type == 'twc'.toString()">
            and ( title like CONCAT('%', #{keyword}, '%') OR
            userNo
            like
            CONCAT('%', #{keyword}, '%') OR content
            like
            CONCAT('%', #{keyword}, '%'))
        </if>
    </if>
</sql>

</mapper>

```

공지사항 최상단으로 가져오기 위하여



```
import ...

Service
AllArgsConstructor

public class BoardServiceImpl implements BoardService {

    private BoardMapper boardMapper;

    private PointService pointService;

    @Override
    public int insertInfo(BoardModel boardModel) {
        pointService.insertPointInfo( point 3, boardModel.getUserNo());
        return boardMapper.insertInfo(boardModel);
    }

    @Override
    public List<BoardCategoryModel> category() { return boardMapper.category(); }

    @Override
    public List<BoardModel> infoList(CriteriaModel criteriaModel) { return boardMapper.infoList(criteriaModel); }

    @Override
    public BoardModel info(int boardNo) {
        boardMapper.hit(boardNo);
    }
}
```

글 작성시 3포인트 적립

```
//이미지가 콜백 형식으로 업로드 되기 때문에 미리 글을 생성해놓은 뒤 state
$('#summernote').summernote({
    placeholder: '내용을 입력하세요.',
    tabsize: 2,
    height: 300,
    minHeight: null, // set minimum height of editor
    maxHeight: null, // set maximum height of editor
    focus: true,
    lang: 'ko-KR',
    //콜백 onImageUpload 통해서 aws에 저장후 바로 불러와서 본문에 삽입하도록처리
    callbacks: {
        onImageUpload: (uploadFiles) => {
            for (let i = 0; i < uploadFiles.length; i++) {
                sendFile(uploadFiles[i]);
            }
        }
    }
});
```

콜백을 통해서 aws에 저장 후 본문에 삽입

```
//summernote 콜백 작동시킬 때 사용하기 위한 ajax
function sendFile(uploadFiles) {
    const formData = new FormData();

    formData.append( name: 'uploadFile', uploadFiles);

    $.ajax({
        url: '/boardAttach/insertImg',
        processData: false,
        contentType: false,
        data: formData,
        type: 'POST',
        beforeSend: (xhr) => {
            xhr.setRequestHeader(header, token);
        },
        success: (result) => {
            //uuid 와 folderPath 분리하기 위해
            const path = result.split("/");
            let temp = "";
            for (let i = 0; i < path.length - 1; i++) {
                temp += path[i] + "/";
            }

            const boardAttach = {
                uploadPath: 'boardAttach' + temp,
                uuid: path[path.length - 1]
            };

            const url = '/aws/getImg?fileName=' + boardAttach.uuid + '&directory=' + boardAttach.uploadPath;
            //summernote에 aws에서 저장된 이미지 불러와서 삽입
            $('#summernote').summernote("insertImage", url);
        }
    });
}
```

Aws에서 이미지 불러와 삽입



테스트

추천

비추천

댓글 3개

관리자 2020-03-26 수정 | 삭제

테스트

유저 2020-03-26 수정 | 삭제

테스트

유저 2020-04-01 수정 | 삭제

테스트

1

목록 수정 삭제

Fail Already it. x

세션에서 회원 권한과 유저 아이디를 가져와 비교하여 글 작성자이거나 운영자인 경우 수정 삭제 가능

중복으로 추천을 하려는 경우 게시글 번호, 유저 번호로 추천 테이블을 조회하여 에러 처리



Bayern Munich

message: 'Please, You do not have permission.',

});

}

})

} else {

iziToast.error({

title: 'Fail',

message: 'Already it.',

});

}

})

});

\$("#badBtn").click(() => {

const data = {

userNo: 1,

boardNo: boardNo,

중복 체크

\$.get("/good/check", data, (result) => {

if (result === 0) {

\$.ajax({

url: "/good/bad",

data: data,

type: "post",

beforeSend: (xhr) => {

xhr.setRequestHeader(header, token);

},

success: () => {...},

error: () => {...}

})

} else {

iziToast.error({

title: 'Fail',

message: 'Already it.',

});

}

})

});

아이콘샵



이름 : 요복이

가격 : 1223 수량 : 5

구입



이름 : 레반도프스키

가격 : 900 수량 : 9

구입



이름 : 19기념

가격 : 5000 수량 : 100

구입



이름 : 새 로고

가격 : 5000 수량 : 100

구입



이름 : 19-20 우승

가격 : 5000 수량 : 100

구입



이름 : 피카츄

가격 : 1111 수량 : 3

구입



이름 : 포켓몬

가격 : 100 수량 : 10

구입



이름 : 18 유니폼

가격 : 5000 수량 : 100

구입



이름 : 18-19 우승

가격 : 5000 수량 : 100

구입



이름 : 분데스리가 로고

가격 : 5000 수량 : 100

구입

1

2

3

```
@Override
public int insertIcon(IconModel iconModel) { return iconShopMapper.insertIcon(iconModel); }

@Override
public List<IconModel> getInfolist(CriteriaModel criteriaModel) {
    return iconShopMapper.getInfolist(criteriaModel);
}

@Override
public int getTotalCnt() { return iconShopMapper.getTotalCnt(); }

@Transactional
@Override
public int buyIcon(int userNo, int iconNo) {
    int price = getPrice(iconNo);
    int userPoint = userService.getPoint(userNo);

    //포인트 부족하면 0 리턴
    if (userPoint < price || checkRetain(userNo, iconNo) != 0) {
        return 0;
    }

    pointService.insertPointInfo( point: price * -1, userNo);
    beforeRepresent(userNo);
    return iconShopMapper.buyIcon(userNo, iconNo);
}
```

유저의 포인트와 아이콘 가격을 이용해 계산 후, iconRetain 테이블에 보유 등록

보관함



이름 : 골드 블랙 로고

대표로 설정



이름 : 홈

대표로 설정



이름 : test

대표로 설정



이름 : ?

대표로 설정



이름 : 원정

대표로 설정

1

```
@AllArgsConstructor
@Service
public class IconStorageServiceImpl implements IconStorageService {

    private IconStorageMapper iconStorageMapper;
    private IconShopService iconShopService;

    @Override
    public List<IconAttachModel> getVerIconList(int userNo, CriteriaModel criteriaModel) {
        return iconStorageMapper.getVerIconList(userNo, criteriaModel.getPageStart(), criteriaModel.getAmount());
    }

    @Override
    public int getTotalCnt(int userNo) { return iconStorageMapper.getTotalCnt(userNo); }

    @Transactional
    @Override
    public int represent(int userNo, int iconNo) {
        iconShopService.beforeRepresent(userNo);
        return iconStorageMapper.represent(userNo, iconNo);
    }
}
```

대표로 하나만 설정되어야 하기때문에 다른 아이콘에 대해 비활성 처리