

### <좋아요 기능>



```
1
$("${btnLike").click(function () {
    $.ajax({
        url:"CommuBbsController",
        data: {command: 'like', seq: ${comdto.seq }, userid: ${current_user.seq }},
        type:"post",
        success : function (data) {
            var result = JSON.parse(data);

            if(result.status == 404){
                $("${img#like_img").attr('src', './img/empty_heart.png');
            } else {
                $("${img#like_img").attr('src', './img/heart.png');
            }

            $("${span#like_count").html(result.like_count);
        }
    });
});
```

좋아요 버튼 눌렀을 때 ajax 함수를 통해 좋아요 버튼 상태를 변경하고 카운트 수 증가시켜줍니다.

```
String Sseq = req.getParameter("seq");
int seq = Integer.parseInt(Sseq);
String Suser = req.getParameter("userid");
int user = Integer.parseInt(Suser);
System.out.println("seq " + seq + " userid " + user);

CommuBbsService comService = CommuBbsService.getInstance();

int like_count = 0;
HashMap<String, Integer> status = new HashMap<>();

3 boolean check = comService.Prevent_duplication(user, seq);

if (check) {
    // 테이블에서 해당 행을 삭제( 제거 ) 한다.
    comService.likeTB_delete(user, seq);

    // status, like count 를 json으로 전송한다.
    status.put("status", 404);
} else {
    // 테이블에서 해당 행을 삭제( 제거 ) 한다.
    comService.likeTB_insert(user, seq);

    // status, like count 를 json으로 전송한다.
    status.put("status", 200);
}

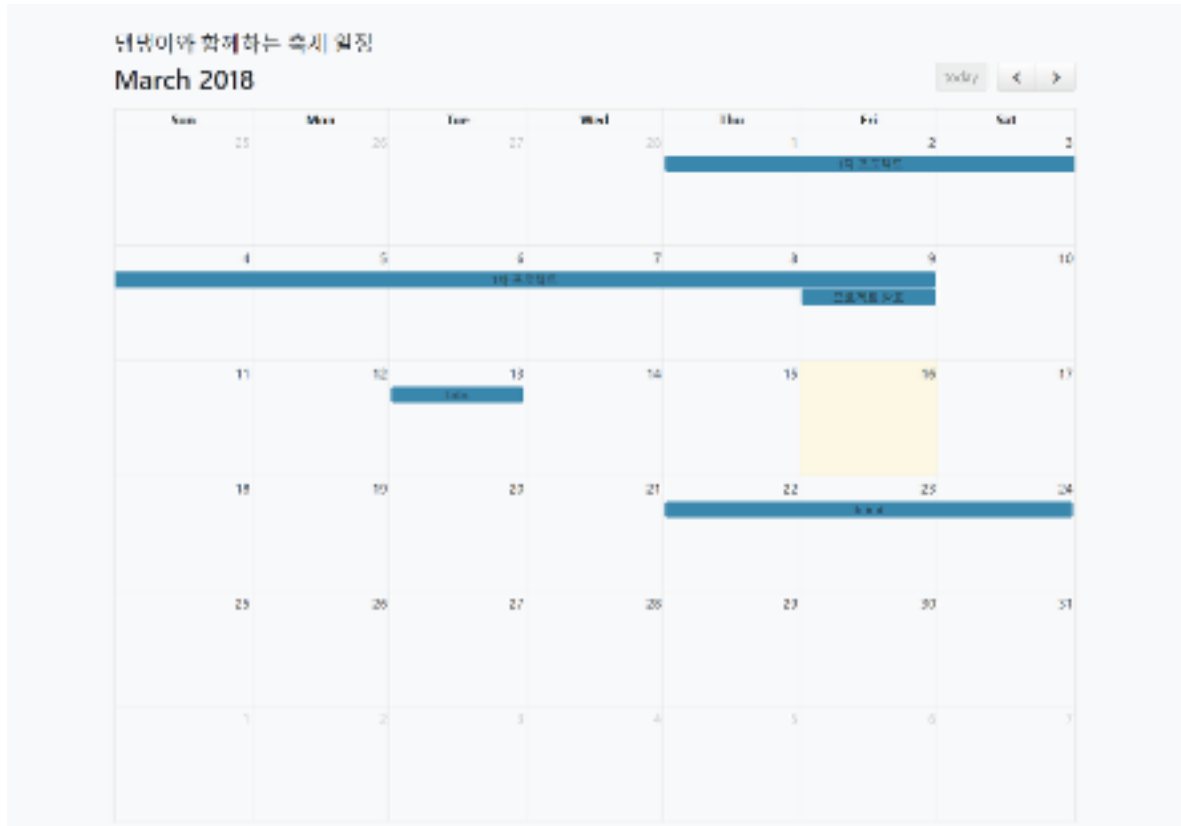
// 테이블을 게시글 seq 로 count(*)
like_count = comService.getLikeCount(seq);
status.put("like_count", like_count);
4 String json = new Gson().toJson(status);

System.out.println(json);
resp.getWriter().write(json);
```

2. HashMap으로 status 보냄 (202 : 좋아요, 404 : 취소)
3. 게시물 클릭한 유저가 좋아요 눌렀나 안눌렀나 확인하는 함수
  - 좋아요 테이블에 (user\_seq, bbs\_seq) 쌍이 존재하는지 확인
4. Json파일을 Gson으로 변환해 보낸다

# DengDengE 개발 기능(상세)

## 6. 행사일정



1. 달력은 Full Calendar API를 사용하여 띄워준다.
2. 행사가 있는 해당 날짜를 선택 시, 일정의 대한 내용이 보여진다.(Modal)
3. 다음 지도 API를 사용하여 행사 일정의 장소를 공지한다.
4. 행사일정 추가는 admin페이지에서 가능하다.

### Controller

```
} else if(command.equals("getEvents")) {
    CalendarService calendarService = CalendarService.getInstance();
    List<CalendarDto> events = calendarService.getEvents();

    String json = new Gson().toJson(events);

    resp.getWriter().write(json);
} else if(command.equals("getEvent")) {
    String sseq = req.getParameter("target");
    int seq = Integer.parseInt(sseq);

    System.out.println(seq);
    CalendarService calendarService = CalendarService.getInstance();
    CalendarDto event = calendarService.getEvent(seq);

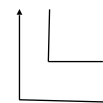
    String json = new Gson().toJson(event);

    resp.getWriter().write(json);
}
```

### Service

```
public List<CalendarDto> getEvents(){
    return calendarDao.getEvents();
}

public CalendarDto getEvent(int seq){
    return calendarDao.getEvent(seq);
}
```



DAO



DB