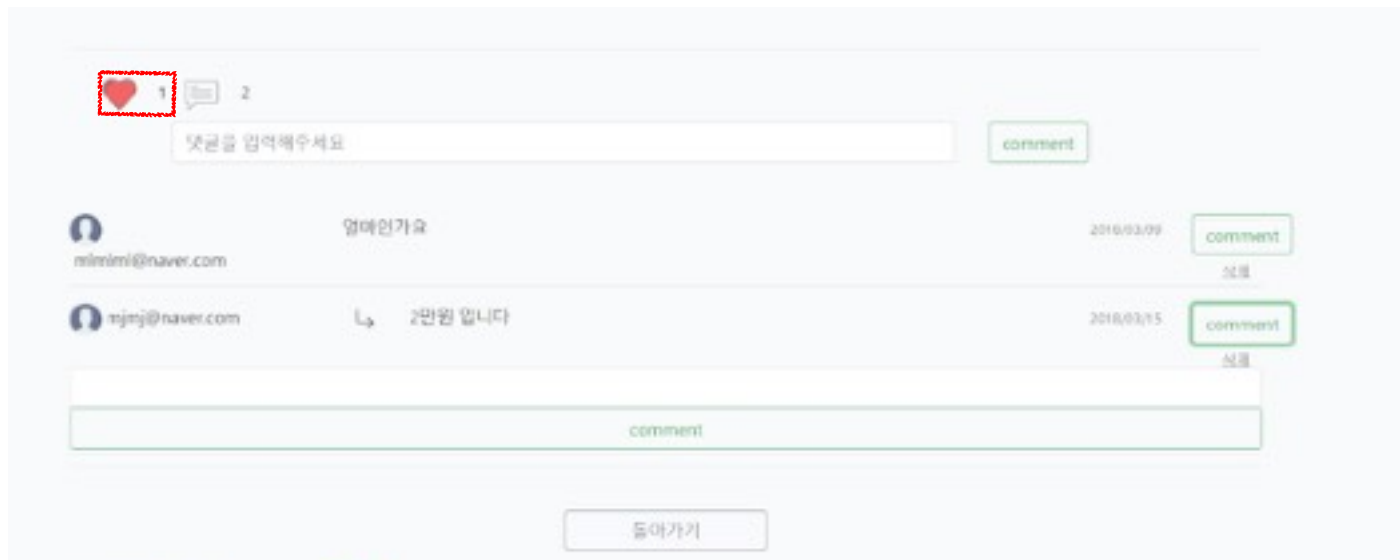


DengDengE 땡땡이

- 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.

controller

```
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();

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

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

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

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

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

// 테이블을 게시글 seq = 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);
```

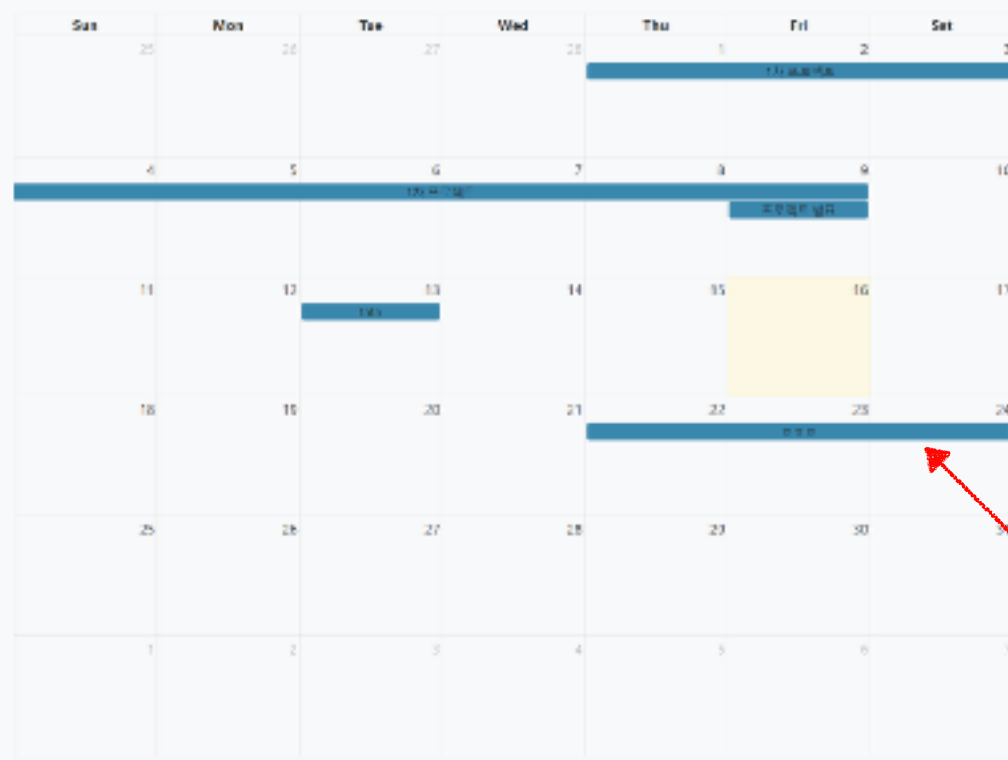
2. HashMap으로 status 보냄 (202 : 좋아요, 404 : 취소)
3. 게시물 클릭한 유저가 좋아요 눌렀나 안눌렀나 확인하는 함수
 - 좋아요 테이블에 (user_seq, bbs_seq) 쌍이 존재하는지 확인
4. Json파일을 Gson으로 변환해 보낸다

DengDengE 땡땡이

- 행사 일정 기능

땡땡이와 함께하는 축제 일정

March 2018



1. Full Calendar API를 통해 데이터 베이스에 등록된 일정을 보여줍니다.
2. 등록된 일정은 DengAdmin에서 추가한 일정입니다.
3. 제휴를 맺은 애견샵 등의 홍보성 일정을 등록해주기 위해 사용합니다.
4. 일정을 클릭하면 상세 정보와 위치 정보를 다음 지도 API를 사용하여 보여줍니다.

1

```
$(document).ready(function() {  
    $.ajax({  
        url : 'CalendarController',  
        data : { command : 'getEvents' },  
        method : 'POST',  
        success : function (data) {  
            var events = [];  
            var temp = JSON.parse(data);  
  
            for(var i=0; i<temp.length; i++){  
                var obj = {};  
                obj.title = temp[i].title;  
                obj.start = temp[i].start;  
                obj.end = temp[i].end;  
                obj.target = temp[i].seq;  
  
                events[i] = obj;  
            }  
  
            $('#calendar').fullCalendar({  
                events: events,  
                selectable: true,  
                eventClick: function(calEvent, jsEvent, view) {  
                    showEvent(calEvent.target);  
                }  
            })  
        }  
    })  
    sendMasterTableSet();  
});
```

페이지가 로딩될 때 모든 이벤트를
달력에 추가합니다.

Controller

```
2  
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);  
}
```