

1. 요구사항

- 상품 상세페이지에 로그인한 회원만 찜을 할 수 있어야 한다.
: 비 로그인시 로그인 화면으로 페이지 이동하도록 조치
- 권한 등급이 'MEMBER'(회원)만 찜을 할 수 있어야 한다.
: 그 외의 권한 등급의 경우 '찜불가'라는 화면이 뜨도록 조치
- 찜을 하면 마이페이지에 있는 찜한 상품 리스트 항목으로 페이지 이동할건지 알림 창이 뜨도록 한다.
- 찜이 되면 해당 상품이 찜이 되었다는 UI가 뜰 수 있도록 한다.
- 찜한 상태에서 찜하기 버튼을 다시 누르면 다시 초기화가 될 수 있도록 한다.

2. 구현사항

2-1. Controller

```
// 상품 상세페이지
@GetMapping("/prdct/{prdct_id}")
public ModelAndView productDetail(@PathVariable("prdct_id") String p_id,
    @AuthenticationPrincipal MemberDetails memberDetails,
    PrdReviewCriteria rcric, PrdQnACriteria qcric, PrdctLikeVO
    prdctLikeVO, ModelAndView mav) throws Exception {

    ... 중략

    // 해당 상품 찜 여부 확인용 데이터 가져오기
    log.info("prdLike...");
    mav.addObject("prdLikeval", commonService.getPrdLikeval(p_id));

    return mav;
}

// 상품 상세페이지 찜하기 기능
@Transactional(rollbackFor = Exception.class)
@PostMapping("/prdct/{prdct_id}")
public ResponseEntity<String> prdctLike(@RequestBody PrdctLikeVO prdctLikeVO) {
    ResponseEntity<String> entity = null;

    log.info("prdctLike...");
    try {
        commonService.setPrdctLike(prdctLikeVO);
        entity = new ResponseEntity<String>("SUCCESS", HttpStatus.OK);
    } catch (Exception e) {
        e.printStackTrace();
        entity = new ResponseEntity<String>(e.getMessage(),
        HttpStatus.BAD_REQUEST);
    }

    return entity;
}

// 상품 상세페이지 찜취소 기능
```

```

@Transactional(rollbackFor = Exception.class)
@DeleteMapping("/prdct/{prdct_id}")
public ResponseEntity<String> prdctLikeCancel(PrdctLikeVO prdctLikeVO) {
    ResponseEntity<String> entity = null;

    log.info("prdctLikeCancel...");
    try {
        commonService.prdctLikeCancel(prdctLikeVO);
        entity = new ResponseEntity<String>("SUCCESS", HttpStatus.OK);
    } catch (Exception e) {
        e.printStackTrace();
        entity = new ResponseEntity<String>(e.getMessage(),
        HttpStatus.BAD_REQUEST);
    }

    return entity;
}

```

2-2. Service

```

// 해당 상품 찜 여부 확인용 데이터 가져오기
public PrdctLikeVO getPrdLikeVal(String prdct_id);

// 상품 상세 페이지 찜하기
public void setPrdctLike(PrdctLikeVO prdctLikeVO);

// 상품 상세페이지 찜취소 기능
public int prdctLikeCancel(PrdctLikeVO prdctLikeVO);

```

```

// 해당 상품 찜 여부 확인용 데이터 가져오기
@Override
public PrdctLikeVO getPrdLikeVal(String prdct_id) {
    log.info("getPrdLikeVal...");
    return commonMapper.getPrdLikeVal(prdct_id);
}

// 상품 상세 페이지 찜하기
@Override
public void setPrdctLike(PrdctLikeVO prdctLikeVO) {
    log.info("setPrdctLike...");
    commonMapper.setPrdctLike(prdctLikeVO);
}

// 상품 상세페이지 찜취소 기능
@Override
public int prdctLikeCancel(PrdctLikeVO prdctLikeVO) {
    log.info("prdctLikeCancel...");
    return commonMapper.prdctLikeCancel(prdctLikeVO);
}

```

2-3. Mapper

```
// 해당 상품 찜 여부 확인용 데이터 가져오기
public PrdctLikeVO getPrdctLikeVal(String prdct_id);

// 상품 상세 페이지 찜하기
public void setPrdctLike(PrdctLikeVO prdctLikeVO);

// 상품 상세페이지 찜취소 기능
public int prdctLikeCancel(PrdctLikeVO prdctLikeVO);
```

```
<!-- 해당 상품 찜 여부 확인용 데이터 가져오기 -->
<select id="getPrdctLikeVal" resultType="edu.bit.ex.vo.PrdctLikeVO">
    <![CDATA[
        SELECT * FROM prdct_like WHERE prdct_id = #{prdct_id}
    ]]>
</select>

<!-- 상품 상세 페이지 찜하기 -->
<insert id="setPrdctLike">
    <![CDATA[
        INSERT INTO prdct_like (prdct_like_number, mbr_id, prdct_id, board_id)
        VALUES (prdct_like_seq.nextval, #{mbr_id}, #{prdct_id}, #{board_id})
    ]]>
</insert>

<!-- 상품 상세페이지 찜취소 기능 -->
<delete id="prdctLikeCancel">
    <![CDATA[
        DELETE FROM prdct_like WHERE prdct_like_number = #{prdct_like_number}
    ]]>
</delete>
```

2-4. View(JSP)

```
<!-- 찜하기 버튼을 누를경우 이벤트 발생 -->
<script type="text/javascript">
    $(document).ready(function() {
        $('#prdct_like_dis').click(function(event) {
            event.preventDefault();
            // 비로그인 상태시 찜하기 버튼을 누르면
            if ("${mbr.mbr_id}" == "") {
                if (confirm("로그인 한 회원만 이용가능합니다. 로그인 하시겠습니까?")) {
                    // 승낙하면 로그인 페이지로 이동
                    location.href = '${pageContext.request.contextPath}/login';
                } else {
                    // 거부하면 해당 페이지 새로고침
                    location.reload();
                }
            }
            // 로그인 상태시 찜하기 버튼을 누르면
        } else {
            // 해당 멤버ID와 상품ID의 정보를 가져온다
            var mbr_id = "${mbr.mbr_id}";
            var prdct_id = "${prdct.prdct_id}";
            var board_id = ${prdct.board_id};
```

```

console.log("mbr_id: " + mbr_id);
console.log("mbr_id type: " + (typeof mbr_id));
console.log("prdct_id: " + prdct_id);
console.log("prdct_id type: " + (typeof prdct_id));
console.log("board_id: " + board_id);
console.log("board_id type: " + (typeof board_id));

var form = {
    mbr_id : mbr_id,
    prdct_id : prdct_id,
    board_id : board_id
};

$.ajax({
    type : "POST",
    url : "${pageContext.request.contextPath}/prdct/{prdct_id}",
    cache : false,
    contentType : 'application/json; charset=utf-8',
    data : JSON.stringify(form),
    success : function(result) {
        console.log(result);
        if (result == "SUCCESS") {
            console.log("찜 성공!")
            if (confirm("해당 상품을 찜하셨습니다. 목록 페이지로 이동
하시겠습니까?")) {
                // 승낙하면 마이페이지의 찜하기 리스트로 이동
                location.href =
                '${pageContext.request.contextPath}/member/mypage/like';
            } else {
                // 거부하면 해당 페이지 새로고침하여 찜한거 반영되게하기
                (HTTP의 속성 때문...)
                location.reload();
            }
        }
    },
    error : function(e) {
        console.log(e);
        alert('찜할 수 없습니다.');
```

location.reload(); // 실패시 새로고침하기

```

    }
})
}
});
});
</script>
<!-- 찜취소 버튼을 누를경우 이벤트 발생 -->
<script type="text/javascript">
$(document).ready(function() {
    $('#prdct_like_ena').click(function(event) {

        event.preventDefault();

        // FormData 객체 생성
        var formData = new FormData();

        // 정보를 가져와 FormData에 append 한다
        var prdct_like_number = $('#prd_like_num').text();

```

```

        console.log("prdct_like_number: " + prdct_like_number);
        console.log("prdct_like_number: " + (typeof prdct_like_number));

        formData.append("prdct_like_number", prdct_like_number);

        $.ajax({
            type : 'DELETE',
            url : $(this).attr("href"),
            cache : false,
            processData: false,
            contentType: false,
            data: formData,
            success: function(result) {
                console.log(result);
                if (result == "SUCCESS") {
                    console.log("찜 취소 성공!")
                    alert('해당 상품을 찜 취소 하셨습니다.');
```

location.href =

```

                    '${pageContext.request.contextPath}/prdct/${prdct.prdct_id}';
                }
            },
            error : function(e) {
                console.log(e);
                alert('찜 취소 할 수 없습니다.');
```

location.reload(); // 실패시 새로고침하기

```

            }
        })
    });
});
</script>

... 중략

<!-- 상품 찜 기능 -->
<div class="icon" style="float: left; padding-left: 20px; padding-top: 10px;">
    <!-- 찜하기 했을 경우 해당 정보 가져오기 -->
    <span id="prd_like_num" style="display:
none;">${prdLikeVal.prdct_like_number}</span>
    <!-- 찜하기 기능은 고객(MEMBER 권한)만 이용할 수 있게 설정 --%>
    <sec:authorize access="isAnonymous()">
        <!-- 로그인 상태가 아니므로 자동으로 로그인 comfirm창이 뜨게 설정 --%>
        <i id="prdct_like_dis" class="fa fa-heart-o fa-2x"
onclick="location.href='${pageContext.request.contextPath}/prdct/{prdct_id}'">찜
하기</i>
    </sec:authorize>
    <sec:authorize access="hasAuthority('ADMIN')">
        <i class="fa fa-heart-o fa-2x">찜불가</i>
    </sec:authorize>
    <sec:authorize access="hasAuthority('SELLER')">
        <i class="fa fa-heart-o fa-2x">찜불가</i>
    </sec:authorize>
    <sec:authorize access="hasAuthority('MEMBER')">
        <c:choose>
            <!-- prdct_like 테이블을 가져와 비교후 예전에 찜하기를 했었다면 찜취소로 활성
화가 된다 --%>
            <c:when test="${(prdLikeVal.prdct_id eq prdct.prdct_id) and
(prdLikeVal.mbr_id eq mbr.mbr_id)}">

```

```

        <i id="prdct_like_ena" class="fa fa-heart fa-2x"
onclick="location.href='${pageContext.request.contextPath}/prdct/{prdct_id}'">찜
취소</i>

    </c:when>
    <%-- prdct_like 테이블을 가져와 비교후 예전에 찜하기를 안했다면 (혹은 찜취소를
했었다면) 찜하기로 활성화가 된다 --%>
    <c:otherwise>
        <i id="prdct_like_dis" class="fa fa-heart-o fa-2x"
onclick="location.href='${pageContext.request.contextPath}/prdct/{prdct_id}'">찜
하기</i>

    </c:otherwise>
</c:choose>
</sec:authorize>
</div>

```

3. 구현 도중 느낀점

- REST를 이용한 Spring 개발도중 Mapping을 어떤 상황에서 적재적소로 활용해야 하는지 그리고 URI가 중복되지 않도록 한다.
- Mybatis에서 '#{}'를 이용하여 값을 유동적으로 넘길 때 NULL 값을 넘기진 않았는지 확인한다.
- AJAX로 값을 넘길때 해당 값이 올바르게 되어있는지 console 로그로 값과 타입을 반드시 확인한다.
- DB내 테이블을 새로 추가해야 되는 사항이 있을 때 불필요하게 많은 컬럼들을 설정하지 않는다.