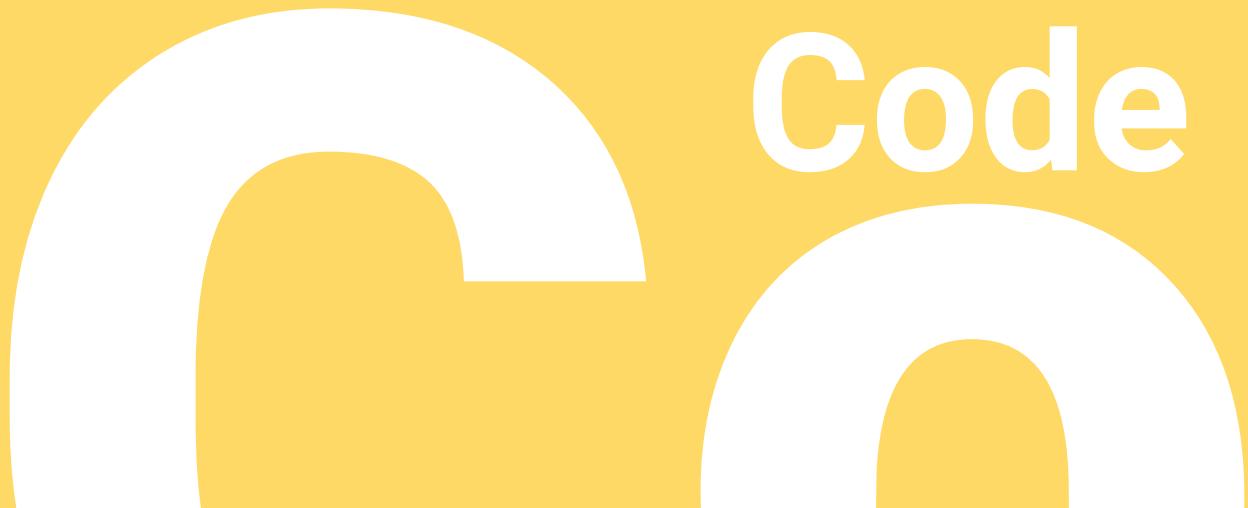


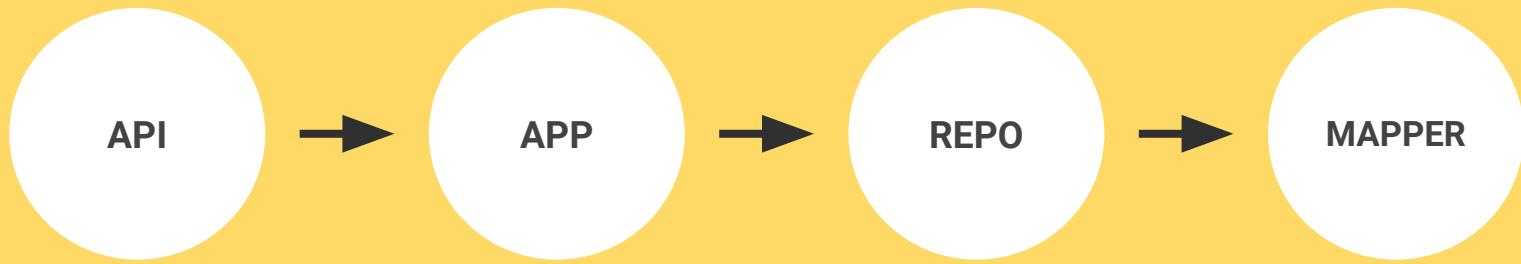
2023

New  
Developer

LEE YUL



Code



# API

# API

## 상품 등록 API

```
@PostMapping("save")
public ResponseEntity<Integer> save(@Valid PdtDto.Create dto, Principal principal) throws IOException {
    pdtApp.save(dto, principal.getName());
    pdtImgApp.save(PdtImgDto.ListFile.builder().pdtNo(dto.getPdtNo()).files(dto.getImages()).build());
    return ResponseEntity.ok().body(1);
}
```

## 상품 수정 API

```
@PutMapping("update")
public ResponseEntity<Integer> update(@Valid PdtDto.Update dto, Principal principal) throws IOException {
    pdtApp.update(dto, principal.getName());
    if (dto.getImages() != null) {
        pdtImgApp.update(PdtImgDto.ListFile.builder().pdtNo(dto.getPdtNo()).files(dto.getImages()).build());
    }
    return ResponseEntity.ok().body(1);
}
```

# API

## 상품 삭제 API

```
@DeleteMapping("delete")
public ResponseEntity<Integer> delete(@NotNull Long pdtNo, Principal principal) {
    pdtApp.delete(pdtNo, principal.getName());
    pdtImgApp.deleteAll(pdtNo);
    pdtFavoriteApp.delete(PdtFavoriteDto.Delete.builder().pdtNo(pdtNo).build(), principal.getName());
    pdtRptApp.deleteByPdtNo(pdtNo);
    return ResponseEntity.ok().body(1);
}
```

## 상품 상세조회 API

```
@GetMapping("find-by-pdt-no")
public ResponseEntity<PdtDto.Detail> findByPdtNo(@NotNull Long pdtNo, Principal principal) {
    return ResponseEntity.ok().body(pdtApp.findByPdtNo(pdtNo, principal.getName()));
}
```

# API

## 상품 목록 조회 API

```
@GetMapping("find-all-by-data")
public ResponseEntity<List<PdtDto.ReadList>> findAllByData(
    @Valid PdtDto.ListData dto, Principal principal, BindingResult bindingResult) {
    return ResponseEntity.ok().body(pdtApp.findAllByData(dto, principal.getName()));
}
```

## 상품 판매자번호 조회 API

```
@GetMapping("find-pdt-seller-no")
public ResponseEntity<Long> findPdtSellerNo(@NotNull Long pdtSellerNo) {
    return ResponseEntity.ok().body(pdtApp.findPdtSellerNo(pdtSellerNo));
}
```

# API

## 회원번호로 상품조회 API

```
@GetMapping("find-all-by-mem-no")
public ResponseEntity<List<PdtDto.ReadList>> findAllByMemNo(Principal principal) {
    return ResponseEntity.ok().body(pdtApp.findAllByMemNo(principal.getName()));
}
```

## 상품 카테고리조회 API

```
@GetMapping("find-all")
public ResponseEntity<List<PdtCateDto.Read>> findAll() {
    return ResponseEntity.ok().body(pdtCateApp.findAll());
}
```

# API

## 상품 관심등록 API

```
@PostMapping("save")
public ResponseEntity<Integer> save(@Valid PdtFavoriteDto.Create dto, Principal principal) {
    return ResponseEntity.ok().body(pdtFavoriteApp.save(dto, principal.getName()));
}
```

## 상품 관심해제 API

```
@DeleteMapping("delete")
public ResponseEntity<Integer> delete(@Valid PdtFavoriteDto.Delete dto, Principal principal) {
    return ResponseEntity.ok().body(pdtFavoriteApp.delete(dto, principal.getName()));
}
```

# API

## 상품 관심목록 조회 API

```
@GetMapping("find-all-by-mem-no")
public ResponseEntity<List<PdtDto.ReadList>> findAllByMemNo(PdtFavoriteDto.ReadList dto, Principal principal){
    return ResponseEntity.ok().body(pdtFavoriteApp.findAllByMemNo(dto, principal.getName()));
}
```

## 상품신고 등록 API

```
@PostMapping("save")
public ResponseEntity<Integer> save(@Valid PdtRptDto.Create dto, Principal principal) {
    return ResponseEntity.ok().body(pdtRptApp.save(dto, principal.getName()));
}
```

**APP**

# APP

## 상품 등록 APP

```
public Integer save(PdtDto.Create dto, String userName) {  
    dto.setPdtSellerNo(memRepo.findIdByMemUsername(userName));  
    return pdtRepo.save(dto);  
}
```

## 상품 수정 APP

```
public Integer update(PdtDto.Update dto, String userName) {  
    dto.setPdtSellerNo(memRepo.findIdByMemUsername(userName));  
    return pdtRepo.update(dto);  
}
```

# APP

## 상품 삭제 APP

```
public Integer delete(Long pdtNo, String userName) {  
    return pdtRepo.delete(PdtDto.Delete  
        .builder()  
        .pdtNo(pdtNo)  
        .pdtSellerNo(memRepo.findIdByUsername(userName))  
        .build());  
}
```

# APP

## 상품 상세조회 APP

```
public PdtDto.Detail findByPdtNo(Long pdtNo, String userName) {
    // 로그인 회원 번호 가져오기
    Long memNo = memRepo.findByIdByUsername(userName);

    // 상품 정보
    PdtDto.Detail dto = pdtRepo.findByPdtNo(pdtNo)
        .orElseThrow(ProductNotFoundException::new);
    for (PdtImgDto.Read read : dto.getImages()) {
        read.setImgPath(imgUrl + read.getImgPath());
    }

    // 회원정보
    MemDto.Member member = memRepo.findById(dto.getProduct().getPdtSellerNo())
        .orElseThrow(MemberNotFoundException::new);
    member.setImgPath(imgUrl + member.getImgPath());
    dto.setMember(member);

    // 판매내역
    List<PdtDto.ReadList> sellerList = pdtRepo.findAllByData(PdtDto.ListData
        .builder()
        .pdtNo(pdtNo)
        .addressSi(dto.getAddress().getAddressSi())
        .addressGu(dto.getAddress().getAddressGu())
        .addressDong(dto.getAddress()
            .getAddressDong())
        .start(1)
        .end(4)
        .pdtSellerNo(member.getMemNo())
        .build());
}
```

```
for (PdtDto.ReadList read : sellerList) {
    read.setImgPath(imgUrl + read.getImgPath());
}
dto.setSellerList(sellerList);

// 카테고리 내역검색
List<PdtDto.ReadList> cateList = pdtRepo.findAllByData(PdtDto.ListData
    .builder()
    .pdtNo(pdtNo)
    .addressSi(dto.getAddress().getAddressSi())
    .addressGu(dto.getAddress().getAddressGu())
    .addressDong(dto.getAddress().getAddressDong())
    .start(1)
    .end(4)
    .pdtCate(dto.getPdtCate())
    .build());
for (PdtDto.ReadList read : cateList) {
    read.setImgPath(imgUrl + read.getImgPath());
}
dto.setCateList(cateList);

pdtRepo.increaseView(PdtDto.ViewCount.builder().memNo(memNo).pdtNo(pdtNo).build());

// 관심수
dto.getProduct().setFavoriteCount(pdtFavoriteRepo.countByPdtNo(pdtNo));

// 관심체크
dto.getProduct().setFavoriteCheck(pdtFavoriteRepo.active(dto.getProduct().getPdtNo(), memNo));

return dto;
}
```

# APP

## 상품 목록 조회 APP

```
public List<PdtDto.ReadList> findAllByData(PdtDto.ListData dto, String userName) {  
    List<PdtDto.ReadList> list = pdtRepo.findAllByData(dto);  
    Long memNo = memRepo.findIdByMemUsername(userName);  
    for (PdtDto.ReadList pdt : list) {  
        pdt.setImgPath(imgUrl + pdt.getImgPath());  
        // 관심수  
        pdt.setFavoriteCount(pdtFavoriteRepo.countByPdtNo(pdt.getPdtNo()));  
        // 관심체크  
        pdt.setFavoriteCheck(pdtFavoriteRepo.active(pdt.getPdtNo(), memNo));  
        // 채팅수  
        pdt.setChatCount(chatRepo.countChatByPdtNo(pdt.getPdtNo()));  
        // 채팅체크  
        pdt.setChatCheck(chatRepo.countChatByPdtNoAndMemNo(ChatDto.ChatLike  
            .builder()  
            .chatPdtNo(pdt.getPdtNo()).chatMemNo(memNo)  
            .build()));  
    }  
    return list;  
}
```

# APP

상품 판매자번호 조회 APP

```
public Long findPdtSellerNo(Long pdtSellerNo) {  
    return pdtRepo.findSellerNoById(pdtSellerNo);  
}
```

# APP

## 상품 판매자번호로 조회 APP

```
public List<PdtDto.ReadList> findAllByMemNo(String username) {  
    Long memNo = memRepo.findByIdByUsername(username);  
    List<PdtDto.ReadList> list = pdtRepo.findAllByMemNo(memNo);  
    list.forEach(item-> {  
        item.setImgPath(imgUrl + item.getImgPath());  
        item.setFavoriteCount(pdtFavoriteRepo.countByPdtNo(item.getPdtNo()));  
        item.setFavoriteCheck(pdtFavoriteRepo.active(item.getPdtNo(), memNo));  
        item.setChatCount(chatRepo.countChatByPdtNo(item.getPdtNo()));  
        item.setChatCheck(chatRepo.countChatByPdtNoAndMemNo(ChatDto.ChatLike  
                .builder()  
                .chatPdtNo(item.getPdtNo())  
                .chatMemNo(memNo)  
                .build()));  
    });  
    return list;  
}
```

# APP

## 상품 카테고리 APP

```
public List<PdtCateDto.Read> findAll() {  
    return pdtCateRepo.findAll();  
}
```

## 상품 관심등록 APP

```
public Integer save(PdtFavoriteDto.Create dto, String userName) {  
    dto.setMemNo(memRepo.findIdByUsername(userName));  
    return pdtFavoriteRepo.save(dto);  
}
```

# APP

## 상품 관심해제 APP

```
public Integer delete(PdtFavoriteDto.Delete dto, String username) {  
    dto.setMemNo(memRepo.findIdByMemUsername(username));  
    return pdtFavoriteRepo.delete(dto);  
}
```

## 상품 관심목록 APP

```
public List<PdtDto.ReadList> findAllByMemNo(PdtFavoriteDto.ReadList dto ,String userNmae){  
    dto.setMemNo(memRepo.findIdByMemUsername(userNmae));  
    List<PdtDto.ReadList> list = pdtFavoriteRepo.findAllByMemNo(dto);  
    for(PdtDto.ReadList read : list){  
        read.setImgPath(imgUrl + read.getImgPath());  
    }  
    return list;  
}
```

# APP

## 상품 이미지 등록 APP

```
public Integer save(PdtImgDto.ListFile dto) throws IOException {
    Long i = 0L;
    for (MultipartFile files : dto.getFiles()) {
        try {
            String fileName = "product/" + dto.getPdtNo()
                + "-" + ++i + "." + files.getOriginalFilename()
                .substring( beginIndex: files.getOriginalFilename()
                    .lastIndexOf( str: ".") + 1).toLowerCase();
            String imgSaveDir = new File( pathname: "").getAbsolutePath() + "\\images/";
            File pdtImg = new File( pathname: imgSaveDir + fileName);
            pdtImg.getParentFile().mkdirs();
            files.transferTo(pdtImg);
            pdtImgRepo.save(PdtImgDto.Create.builder().pdtNo(dto.getPdtNo()).imgNo(i).imgPath(fileName).build());
        } catch (IOException e) {
            throw new ProductImageSaveFailException();
        }
    }
    return 1;
}
```

# APP

## 상품 이미지 수정 APP

```
public Integer update(PdtImgDto.ListFile dto) throws IOException {
    List<PdtImgDto.Read> oldList = pdtImgRepo.searchOfPdtNo(dto.getPdtNo());
    for (PdtImgDto.Read deleteImg : oldList) {
        String imgUpdateDir = new File( pathname: "").getAbsolutePath() + "\\\" + "/images/";
        File file = new File( pathname: imgUpdateDir + deleteImg.getImgPath());
        file.delete();
    }
    pdtImgRepo.deleteAll(dto.getPdtNo());
    save(dto);
    return 1;
}
```

# APP

## 상품 이미지 삭제 APP

```
public Integer deleteAll(Long pdtNo) {  
    List<PdtImgDto.Read> oldList = pdtImgRepo.searchOfPdtNo(pdtNo);  
    for (PdtImgDto.Read deleteImg : oldList) {  
        String imgUpdateDir = new File( pathname: "").getAbsolutePath() + "\\\" + "/images/";  
        File file = new File( pathname: imgUpdateDir + deleteImg.getImgPath());  
        file.delete();  
    }  
    return pdtImgRepo.deleteAll(pdtNo);  
}
```

# APP

## 상품 신고 등록 APP

```
public Integer save(PdtRptDto.Create dto, String userName) {  
    // 로그인한 사용자(신고자 번호)  
    Long memNo = memRepo.findByIdByUsername(userName);  
    // 사용자 번호값 넣기  
    dto.setRptMemNo(memNo);  
    // 이미 신고한 사용자  
    Long reporter = pdtRptRepo.findByIdBySearch(PdtRptDto.Search  
        .builder()  
        .rptPdtNo(dto.getRptPdtNo())  
        .rptMemNo(dto.getRptMemNo())  
        .build());  
}
```

```
// 신고자 일일 신고횟수 체크  
Integer rptCount = pdtRptRepo.countOfMemNo(memNo);  
// 신고자번호와 이미저장된 번호가 일치한다면 저장 실패 예외처리  
if (reporter != null && memNo.equals(reporter)) {  
    throw new ReportSaveFailException();  
    // 회원 당 신고횟수가 5이상이면 저장 실패 예외처리  
} else if (rptCount >= 5) {  
    throw new ReportCountExcessException();  
} else {  
    return pdtRptRepo.save(dto);  
}  
}
```

**REPO**

# REPO

## 상품 신고 등록 APP

```
public interface PdtRepo {  
    // 판매글 작성  
    ▲ Lee Yul  
    public Integer save(PdtDto.Create dto);  
  
    // 판매글 수정  
    ▲ Lee Yul  
    public Integer update(PdtDto.Update dto);  
  
    // 글 읽기  
    7개 사용 위치 ▲ Lee Yul  
    public Optional<PdtDto.Detail> findByPdtNo(Long pdtNo);  
  
    // 조회수 증가  
    1개 사용 위치 ▲ Lee Yul  
    public Integer increaseView(PdtDto.ViewCount dto);
```

```
// 회원별 글목록  
3개 사용 위치 ▲ Lee Yul  
public List<PdtDto.ReadList> findAllByData(PdtDto.ListData dto);  
  
// 판매글 삭제  
▲ Lee Yul  
public Integer delete(PdtDto.Delete dto);  
  
// 판매글 전체 삭제  
0개의 사용위치 ▲ jasonKim  
Integer deleteAllBySellerNo(Long sellerNo);  
  
// 판매자 찾기  
6개 사용 위치 ▲ Lee Yul  
public Long findSellerNoById(Long pdtNo);  
  
// 등록한 상품 찾기  
▲ jasonKim  
List<PdtDto.ReadList> findAllByMemNo(Long memNo);
```

# REPO

## 상품 카테고리 REPO

```
public interface PdtCateRepo {  
    // 카테고리 목록  
    ━ Lee Yul  
    public List<PdtCateDto.Read> findAll();
```

# REPO

## 상품 관심 REPO

```
public interface PdtFavoriteRepo {  
    // 관심 등록  
    // Lee Yul  
    public Integer save(PdtFavoriteDto.Create dto);  
  
    // 관심 해제  
    // Lee Yul  
    public Integer delete(PdtFavoriteDto.Delete dto);  
  
    // 관심 목록  
    // Lee Yul  
    public List<PdtDto.ReadList> findAllByMemNo(PdtFavoriteDto.ReadList dto);
```

```
// 상품별 관심수 확인  
// 사용 위치 ▾ Lee Yul  
public Integer countByPdtNo(Long pdtNo);  
  
// 관심글 확인  
// 사용 위치 ▾ Lee Yul  
public Boolean active(Long pdtNo, Long memNo);  
  
// 회원별 전체 삭제  
// 사용 위치 ▾ jasonKim  
Integer deleteAllByMemNo(Long memNo);
```

# REPO

## 상품 이미지 REPO

```
public interface PdtImgRepo {  
    // 이미지 저장  
    ♦ Lee Yul  
    public Integer save(PdtImgDto.Create dto);  
  
    // 이미지 수정  
    ♦ Lee Yul  
    public Integer update(PdtImgDto.Update dto);  
  
    // 이미지 개별삭제  
    0개의 사용위치 ♦ Lee Yul  
    public Integer deleteByImgNo(PdtImgDto.Delete dto);
```

```
// 이미지 전체삭제  
♦ Lee Yul  
public Integer deleteAll(Long pdtNo);  
  
// 상품 이미지 조회  
3개 사용 위치 ♦ Lee Yul  
public List<PdtImgDto.Read> searchOfPdtNo(Long pdtNo);  
  
// 메인 이미지 조회  
2개 사용 위치 ♦ Lee Yul  
public String findMainImg(Long pdtNo);
```

# REPO

## 상품 이미지 REPO

```
public interface PdtRptRepo {  
    // 신고등록  
    ▲ Lee Yul  
    public Integer save(PdtRptDto.Create dto);  
    // 신고자번호 조회  
    1개 사용 위치 ▲ jasonKim  
    public Integer findIdBySearch(PdtRptDto.Search dto);  
    // 신고자 일일 신고횟수 체크  
    1개 사용 위치 ▲ Lee Yul  
    public Integer countOfMemNo(Long memNo);  
    // 상품 신고횟수 체크  
    0개의 사용위치 ▲ Lee Yul  
    public Integer countOfRptPdtNo(Long rptPdtNo);  
    // 신고누적5회인 상품번호 조회  
    1개 사용 위치 ▲ Lee Yul  
    public List<Long> findPdtNoByCount();
```

```
public List<Long> findPdtNoByCount();  
// 상품 삭제  
1개 사용 위치 ▲ Lee Yul  
public Integer deleteOfPdt(Long PdtNo);  
// 상품 이미지 삭제  
1개 사용 위치 ▲ Lee Yul  
public Integer deleteOfImg(Long PdtNo);  
// 상품 관심 삭제  
1개 사용 위치 ▲ Lee Yul  
public Integer deleteOfFavorite(Long PdtNo);  
// 상품 신고 삭제  
2개 사용 위치 ▲ Lee Yul  
public Integer deleteOfRpt(Long PdtNo);  
// 신고전체삭제  
▲ Lee Yul  
public Integer deleteAll();
```

# **MAPPER**

# MAPPER

## 상품 등록 MAPPER

```
<insert id="save">
    <selectKey resultType="long" order="BEFORE" keyProperty="pdtNo">
        select nvl(max(pdt_no),0)+1 from Product
    </selectKey>
    insert into
        PRODUCT( pdt_no, pdt_cate_no, pdt_seller_no, pdt_name, pdt_desc, pdt_price)
    values
        ( #{pdtNo}, #{pdtCateNo}, #{pdtSellerNo}, #{pdtName}, #{pdtDesc}, #{pdtPrice})
</insert>
```

# MAPPER

## 상품 수정 MAPPER

```
<update id="update">
    update PRODUCT
    <trim prefix="set" suffixOverrides=",">
        <if test="pdtCateNo != null">pdt_cate_no=#{pdtCateNo},</if>
        <if test="pdtAddressNo != null">pdt_address_no=#{pdtAddressNo},</if>
        <if test="pdtName != null">pdt_name=#{pdtName},</if>
        <if test="pdtDesc != null">pdt_desc=#{pdtDesc},</if>
        <if test="pdtPrice != null">pdt_price=#{pdtPrice},</if>
        <if test="pdtStatus != null">pdt_status=#{pdtStatus},</if>
    </trim>
    where pdt_no=#{pdtNo} and pdt_seller_no = #{pdtSellerNo}
</update>
```

# MAPPER

## 상품 삭제 MAPPER

```
<delete id="delete">
    delete from PRODUCT where pdt_no=#{pdtNo} and pdt_seller_no = #{pdtSellerNo}
</delete>
```

## 상품 조회수 MAPPER

```
<update id="increaseView">
    update PRODUCT set pdt_views = pdt_views+1 where pdt_no = #{pdtNo} and pdt_seller_no != #{memNo}
</update>
```

# MAPPER

## 상품 상세조회 MAPPER

```
<select id="findByPdtNo" resultMap="pdtDtoRead">
    select
        p.pdt_no, p.pdt_seller_no, p.pdt_name, p.pdt_desc, p.pdt_status, p.pdt_price, p.pdt_views, p.pdt_created_at,
        d.address_no, d.mem_no, d.address_si, d.address_gu, d.address_dong,
        i.img_no, i.img_path,
        c.pdt_cate
    from
        PRODUCT p left outer join ADDRESS d on p.pdt_seller_no = d.mem_no and p.pdt_seller_no = d.mem_no
        left outer join PRODUCT_IMAGE i on p.pdt_no = i.pdt_no
        left outer join PRODUCT_CATEGORY c on p.pdt_cate_no = c.pdt_cate_no
    where
        p.pdt_no=#{pdtNo}
</select>
<resultMap id="pdtDtoRead" type="com.hts.market.domain.product.dto.PdtDto$Detail">
    <result column="pdt_cate" property="pdtDate"/>
    <association property="address" javaType="com.hts.market.domain.member.dto.AddressDto$Read">
        <result column="address_no" property="addressNo"/>
        <result column="mem_no" property="memNo"/>
        <result column="address_si" property="addressSi"/>
    </association>
    <collection property="images" ofType="com.hts.market.domain.product.dto.PdtImgDto$Read">
        <result column="pdt_no" property="pdtNo"/>
        <result column="img_no" property="imgNo"/>
        <result column="img_path" property="imgPath"/>
    </collection>
</resultMap>
```

```
<result column="address_gu" property="addressGu"/>
<result column="address_dong" property="addressDong"/>
</association>
<association property="product" javaType="com.hts.market.domain.product.dto.PdtDto$Read">
    <result column="pdt_no" property="pdtNo"/>
    <result column="pdt_seller_no" property="pdtSellerNo"/>
    <result column="pdt_name" property="pdtName"/>
    <result column="pdt_desc" property="pdtDesc"/>
    <result column="pdt_status" property="pdtStatus"/>
    <result column="pdt_price" property="pdtPrice"/>
    <result column="pdt_views" property="pdtViews"/>
    <result column="pdt_created_at" property="pdtCreatedAt"/>
</association>
<collection property="images" ofType="com.hts.market.domain.product.dto.PdtImgDto$Read">
    <result column="pdt_no" property="pdtNo"/>
    <result column="img_no" property="imgNo"/>
    <result column="img_path" property="imgPath"/>
</collection>
</resultMap>
```

# MAPPER

## 상품 목록 조회 MAPPER

```
<select id="findAllByData" resultType="com.hts.market.domain.product.dto.PdtDto$ReadList">
    SELECT
        b.*
    FROM
        (
            SELECT /*+ index_desc(PRODUCT PK_PRODUCT) */
                ROWNUM AS rnum, a.*
            FROM
                (
                    SELECT
                        p.pdt_no, p.pdt_name, p.pdt_status, p.pdt_price, p.pdt_views, p.pdt_created_at, p.pdt_cate_no,
                        c.pdt_cate,
                        i.img_path,
                        d.address_si, d.address_gu, d.address_dong
                    FROM
                        product p left outer join address d on p.pdt_seller_no = d.mem_no
                        left outer join product_image i on p.pdt_no=i.pdt_no
                        left outer join product_category c on p.pdt_cate_no = c.pdt_cate_no
                    where
                        d.address_si = #{addressSi} and d.address_gu = #{addressGu} and d.address_dong = #{addressDong} and i.img_no=1
                ) a
            WHERE
                ROWNUM <= #{end}
        ) b
    ORDER BY
        p.pdt_no DESC
</select>
```

```
<if test="keyword != null">
    and (p.pdt_name LIKE '%'|| #{keyword} || '%' or p.pdt_desc Like '%'|| #{keyword} || '%')
</if>
<if test="pdtcateNo != null">
    and p.pdt_cate_no = #{pdtcateNo}
</if>
<if test="pdtDate != null and pdtNo !=null">
    and c.pdt_cate = #{pdtDate} and p.pdt_no != #{pdtNo}
</if>
<if test="pdtSellerNo != null and pdtNo !=null">
    and p.pdt_seller_no = #{pdtSellerNo} and p.pdt_no != #{pdtNo}
</if>
ORDER BY
p.pdt_no DESC
) a
WHERE
    ROWNUM <= #{end}
) b
WHERE
    b.rnum >= #{start}
</select>
```

# MAPPER

## 상품 판매자 번호로 조회 MAPPER

```
<select id="findSellerNoById" resultType="long">
    select pdt_seller_no from PRODUCT where pdt_no = #{pdtNo} and rownum <=1
</select>
```

## 상품 회원번호로 목록조회 MAPPER

```
<select id="findAllByMemNo" resultType="com.hts.market.domain.product.dto.PdtDto$ReadList">
    SELECT
        p.pdt_no, p.pdt_name, p.pdt_price, p.pdt_created_at,
        c.pdt_cate,
        i.img_path,
        d.address_si, d.address_gu, d.address_dong
    FROM
        product p left outer join address d on p.pdt_seller_no = d.mem_no
        left outer join product_image i on p.pdt_no=i.pdt_no
        left outer join product_category c on p.pdt_cate_no = c.pdt_cate_no
    where
        p.pdt_seller_no=#{memNo} and i.img_no=1
    ORDER BY
        p.pdt_no DESC
</select>
```

# MAPPER

## 상품 카테고리 조회 MAPPER

```
<select id="findAll" resultType="com.hts.market.domain.product.dto.PdtCateDto$Read">
    select pdt_cate_no, pdt_cate from PRODUCT_CATEGORY
</select>
```

## 상품 관심 등록 MAPPER

```
<insert id="save">
    insert into PRODUCT_FAVORITE(pdt_no, mem_no) values (#{pdtNo}, #{memNo})
</insert>
```

## 상품 관심 해제 MAPPER

```
<delete id="delete">
    delete from PRODUCT_FAVORITE where pdt_no=#{pdtNo} and mem_no=#{memNo}
</delete>
```

# MAPPER

## 상품 관심 목록 조회 MAPPER

```
<select id="findAllByMemNo" resultType="com.hts.market.domain.product.dto.PdtDto$ReadList">
    SELECT
        b.*
    FROM
        (
            SELECT /*+ index_desc(PRODUCT_FAVORITE PK_PRODUCT_FAVORITE) */
                ROWNUM AS rnum, a.*
            FROM
                (
                    select
                        f.pdt_no,
                        p.pdt_name, p.pdt_price, p.pdt_created_at,
                        i.img_path,
                        d.address_si, d.address_gu, d.address_dong
                    from
                        PRODUCT_FAVORITE f left outer join product p on f.pdt_no = p.pdt_no
```

```
        left outer join product_image i on f.pdt_no = i.pdt_no
        left join address d on f.mem_no = d.mem_no
        where
            f.mem_no=#{memNo} and d.address_no=(select max(dd.address_no)
            from address dd where dd.mem_no=#{memNo})
        ORDER BY
            pdt_no DESC
        ) a
        WHERE
            ROWNUM <= ${end}
        ) b
        WHERE
            b.rnum >= #{start}
</select>
```

# MAPPER

## 상품 관심수 확인 MAPPER

```
<select id="countByPdtNo" resultType="int">
    select count(mem_no) from PRODUCT_FAVORITE where pdt_no=#{pdtNo}
</select>
```

## 상품 관심여부 확인 MAPPER

```
<select id="active" resultType="boolean">
    select count(*) from PRODUCT_FAVORITE where pdt_no=#{pdtNo} and mem_no = #{memNo}
</select>
```

## 판매글 삭제시 관심 전체삭제 MAPPER

```
<delete id="deleteAllByMemNo">
    delete from PRODUCT_FAVORITE where mem_no=#{memNo}
</delete>
```

# MAPPER

## 상품 이미지 등록 MAPPER

```
<insert id="save">
    insert into
        PRODUCT_IMAGE(pdt_no, img_no, img_path )
    values
        ( #{pdtNo}, #{imgNo}, #{imgPath} )
</insert>
```

## 상품 이미지 수정 MAPPER

```
<update id="update">
    update
        PRODUCT_IMAGE
    set
        img_path=#{imgPath}
    where
        pdt_no = #{pdtNo} and img_no = #{imgNo}
</update>
```

# MAPPER

## 상품별 이미지 전체조회 MAPPER

```
<select id="searchOfPdtNo" resultType="com.hts.market.domain.product.dto.PdtImgDto$Read">
    select * from PRODUCT_IMAGE where pdt_no=#{pdtNo}
</select>
```

## 상품 이미지 삭제 MAPPER

```
<delete id="deleteByImgNo">
    delete from
    PRODUCT_IMAGE
    where
    pdt_no=#{pdtNo} and img_no=#{imgNo}
</delete>
```

# MAPPER

## 상품별 이미지 전체삭제 MAPPER

```
<delete id="deleteAll">
    delete from
    PRODUCT_IMAGE
    where
    pdt_no=#{pdtNo}
</delete>
```

## 상품 메인이미지 조회 MAPPER

```
<select id="findMainImg" resultType="string">
    select
    img_path
    from
    PRODUCT_IMAGE
    where
    pdt_no=#{pdtNo} and img_no=1
</select>
```

# MAPPER

## 상품 신고작성 MAPPER

```
<insert id="save">
    <selectKey resultType="long" order="BEFORE" keyProperty="rptNo">
        select nvl(max(rpt_no),0)+1 from PRODUCT_REPORT
    </selectKey>
    insert into
        PRODUCT_REPORT(rpt_no, rpt_pdt_no, rpt_mem_no, rpt_msg_no)
    values
        ( #{rptNo}, #{rptPdtNo}, #{rptMemNo}, #{rptMsgNo})
</insert>
```

## 상품 신고자 번호 조회 MAPPER

```
<select id="findIdBySearch" resultType="int">
    select count(*) from PRODUCT_REPORT where rpt_mem_no = #{rptMemNo} and rpt_pdt_no = #{rptPdtNo}
</select>
```

# MAPPER

## 상품 신고작성 MAPPER

```
<insert id="save">
    <selectKey resultType="long" order="BEFORE" keyProperty="rptNo">
        select nvl(max(rpt_no),0)+1 from PRODUCT_REPORT
    </selectKey>
    insert into
        PRODUCT_REPORT(rpt_no, rpt_pdt_no, rpt_mem_no, rpt_msg_no)
    values
        ( #{rptNo}, #{rptPdtNo}, #{rptMemNo}, #{rptMsgNo})
</insert>
```

## 상품 이미 신고등록 상품 조회 MAPPER

```
<select id="findIdBySearch" resultType="int">
    select count(*) from PRODUCT_REPORT where rpt_mem_no = #{rptMemNo} and rpt_pdt_no = #{rptPdtNo}
</select>
```

# MAPPER

## 상품 신고 일일누적 확인 MAPPER

```
<select id="countOfMemNo" resultType="integer">
    select count(rpt_mem_no) from PRODUCT_REPORT where rpt_mem_no= #{rptMemNo}
</select>
```

## 상품 누적 신고 횟수 조회 MAPPER

```
<select id="countOfRPptPdtNo" resultType="integer">
    select count(rpt_pdt_no) from PRODUCT_REPORT where rpt_pdt_no = #{rptPdtNo}
</select>
```

# MAPPER

## 5회이상 신고된 상품조회 MAPPER

```
<select id="findPdtNoByCount" resultType="long">
    select
        rpt_pdt_no
    from
        PRODUCT_REPORT
    group by
        rpt_pdt_no
    having
        count(rpt_pdt_no) >= 5
</select>
```

## 상품 신고 전체 삭제 MAPPER

```
<delete id="deleteAll">
    delete from PRODUCT_REPORT
</delete>
```