



# BoostCourse

(Full-Stack Developer)

# 예약서비스



## 예약서비스?

- 예약서비스는 네이버웹서비스에서 사용됐던 서비스다.
- 실제 네이버 서비스의 동작과는 다르며, 본 기획서 내용을 토대로 개발해야 한다.
- 모바일웹과 PC웹을 하나의 UI로 구성한다. 따라서 PC에서도 동작하고, 모바일웹에서도 동작되는 UI이며, 예약서비스는 PC보다는 모바일웹에 어울리는 UI다.
- 크게 5가지 서비스 기능을 개발해야 한다.
  - 메인페이지
  - 각 항목의 상세페이지
  - 예약하기
  - 나의 예매 내역 확인
  - 한줄평등록

# 기능 요구사항 정의서

메인번호	기능	기능번호	기능명	설명	진행사항
1	타이틀	1.1	페이지 이동	- 클릭 시 예약 홈(메인페이지)으로 이동	
		1.2	페이지 이동	- 나의 예약페이지로 연결되어야 함	
		1.3	텍스트 출력	- 1.6번 버튼에 의해서 네이게이션을 하게 되면 해당 번호로 노출된다. - 최대로 노출해야 하는 개수는 1개이거나 2개이다. (기타이미지가 존재하면 2개)	
		1.4	이미지 출력	- 서버에서 생성한 메인이미지가 노출된다. 클릭 시 반응은 없다.	
		1.5	슬라이드 버튼	- 1.5는 메인이미지 이외에 기타이미지가 존재할 때만 노출된다. - 따라서 상세페이지가 로딩된 후 Ajax를 통해서 기타이미지가 있는지 확인해야 한다. - 기타이미지가 여러 개 API에 존재하더라도, 실제로는 한 개만 더 화면에 노출한다. - 터치이벤트 발생 시 슬라이딩 애니메이션 되며 다른 사진 노출	
		1.6	텍스트 출력	- 전시타이틀이 화면 가운데 텍스트로 노출 되어야 함	
2	상품설명	2.1	버튼	- 펼쳐보기를 누르면 펼쳐져서 보인다. - 다시 누르면 접기가 된다.	
3	이벤트 영역	3.1	텍스트 출력	- 클릭 시 반응 없음	
4	예매하기	4.1	페이지 이동	- 클릭 시 예매페이지로 이동	
5	상세설명 영역	5.1	텍스트 출력	- 5.0 만점의 평균을 표시, 총 건수도 표시한다.	
		5.2	텍스트 및 이미지 출력	- 이미지는 한참 노출됨 - 하단에 평점, 이메일, 방문날짜 표시 - 이미지 클릭시 반응없음	
		5.3	버튼	- 리스트는 3개까지만 노출하고 그 이상은 "더보기"버튼을 눌러서 전체 한줄평 노출 페이지로 이동함	
		5.4	페이지 이동	- 상세페이지 하단의 "더보기"를 누르면 전체 한줄평 리스트 화면이 노출된다. - 평점/개수 노출 - 한줄평 내용과 평점, 이메일, 방문날짜 노출	
6	상세설명 / 오시는 길	6.1	탭 버튼	- 탭으로 전환되며, 오시는 길을 누르면 6.5 컨텐츠가 노출	
		6.2	이미지 출력	- 공지사항은 이미지로 노출된다. (모든 아이템이 같은 이미지를 쓴다.)	
		6.3	이미지 및 텍스트 출력	- 전시타이틀, 새주소, 지번주소, 세부주소, 전화번호 노출 - 길찾기, 네비게이션은 반응 없음 - 지도는 동일한 내용의 지도이미지를 노출	
7	Top 이동영역	6.3	버튼	- 선택시 상단으로 이동한다.	

# 기능 요구사항 정의서

- Project 4 기능 요구사항
  1. 타이틀
  2. 상품설명
  3. 이벤트 영역
  4. 예매하기
  5. 상세설명 영역
  6. 상세설명 / 오시는 길
  7. Top 이동영역

# 기술 요구사항 정의서

# 기능 요구사항 정의서

- Project 4 기술 요구사항
  - 1. 웹 프론트 기술 요구사항
    - ① 객체 리터럴
    - ② 라이브러리 사용
    - ③ 클린코드
    - ④ 레이아웃
  - 2. 웹 백엔드 기술 요구사항
    - ① 레이어드 아키텍처
    - ② JDBC
    - ③ Web API

영역	기술명	규칙번호	규칙 설명
웹 프론트 기술 요구사항	객체 리터럴 패턴 사용	1.1	- 전체 코드는 객체리터럴 패턴으로 구현
		1.2	- 전역변수의 최소화
		1.3	- 한 개 이상의 객체리터럴을 사용할 수 있다.
	개발 규칙	2.1	- 상단 타이틀 이미지 영역의 애니메이션은 CSS3의 transition과 transform을 활용해서 구현해야 한다.
		2.2	- 상단에 추가로 노출해야 하는 기타 이미지는 Ajax 요청을 통해 가져온다.
		2.3	- DOMContentLoaded 이후에 모든 스크립트 로직이 동작하게 한다.
		2.4	- Event delegation으로 처리할 수 있는 영역은 최대한 delegation 방법으로 처리한다.
	라이브러리 사용	3.1	- Templating 작업은 handlebar 라이브러리를 사용해서 구현해야 한다.
		3.2	- 점기 / 펼치기 기능은 JQuery 라이브러리를 사용할 수 있다.
	클린코드	4.1	- 함수 하나에 여러 개의 기능을 넣지 않고, 함수를 여러 개로 분리
		4.2	- 중복 코드를 제거하고 공통 부분을 공통함수로 만든다.
		4.3	- 함수 이름은 동사 + 명사로 작성, 해당 함수에서는 함수명에 알맞은 동작을 한다.
웹 백엔드 기술 요구사항	Layered Architecture	1.1	- Controller, Service, Dao로 레이어드 아키텍처를 구성한다.
		1.2	- Controller에서 Service, Service에서 Repository의 기능을 호출할 수 있지만 그 반대는 허용되지 않는다.
		1.3	- Spring JDBC를 이용하여 주어진 테이블로부터 입력, 수정, 삭제, 조회하는 DAO와 DTO를 작성한다.
		1.4	- 서비스 인터페이스를 작성하고 해당 인터페이스에 비즈니스 메소드를 추가합니다.
		1.5	- 서비스 인터페이스를 구현하는 클래스를 작성하고 클래스 내 메소드에 적절한 트랜잭션에 관련된 어노테이션을 사용한다.
	JDBC	2.1	- SQL 문을 표준에 맞추어 작성하며 각 DBMS 별로 구현된 문법 사용은 최소화한다.
		2.2	- 상품 id를 전달 받아 해당 상품의 상품 정보, 예매자 리뷰 목록을 반환하는 Web API를 작성한다.
			- 예매자 리뷰정보의 경우 전체를 제공한다.
		2.3	- 데이터 수를 조회하는 경우 COUNT(*)를 사용한다.
	Web API	3.1	- 클라이언트에게 Web API를 제공하기 위해 RestController를 작성한다.

매인번호	기능	기능번호	기능명	설명	진행사항
1	화면 레이아웃	1.1	상세페이지 하단 영역	- 공지 사항과 지도는 일반 이미지를 사용해서 추가한다. 지도 이미지는 동일한 지도를 재사용해도 좋다.	
		1.2	타이틀영역 배경이미지 슬라이딩	- 서버에 기타(etc)로 존재하는 이미지가 있다면, 좌/우슬라이딩 버튼을 노출하고, 이미지가 슬라이딩 애니메이션 되어 한다.	
		1.3		- 무한 슬라이딩으로 계속 동작한다.	
		1.4	펼쳐보기	- 점기/펼치기를 선택하면 레이어의 높이가 변하면서 반응한다.	
		1.5	하단영역의 상세정보와 오시는길	- 탭으로 동작한다. 화면새로고침 없이 탭을 통해서 동작해야 한다.	

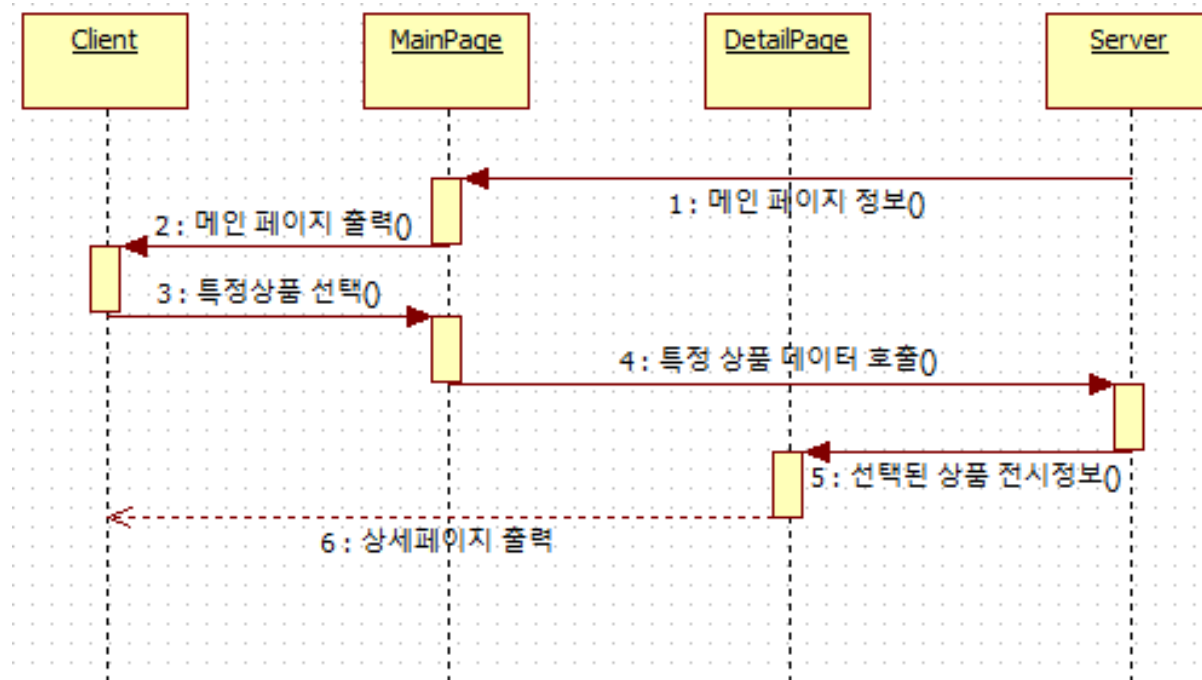
# 소스코드 규칙

제원번호	코드 작성 규칙	규칙번호	규칙명	설명	진행사항
1	화면 레이아웃	1.1	상세페이지 메인화면	- HTML/CSS가 제공되는 것을 기본으로 작성해야 하며, 메인 페이지의 화면 구성은 아래 기획서대로 보아야 한다.	
2		2.1	이름 규칙	- JAVA Naming Conventions 을 지킨다.	
		2.2		- 클래스의 이름과 메소드의 이름은 직관적으로 작성하도록 한다.	
		2.3		- 클래스의 이름과 메소드의 이름만 보아도, 어떤 기능을 가지고 있을지 어떤 내용이 구현되어 있을지 짐작할 수 있어야 한다.	
3		3.1	중복된 코드 제거 및 코드 구조화	- 코드를 읽는 사람이 개념을 쉽게 파악할 수 있도록 읽기 쉬운 코드를 작성하도록 한다.	
		3.2		- 예를 들어 변수 이름을 구체적으로 작성하도록 한다.	
		3.3		- 중복된 코드가 있다면, 별도의 메소드나 클래스로 분리하도록 한다.	
		3.4		- 하나의 메소드가 너무 많은 코드를 담지 않는다.	
4	Java	4.1	가독성	- 코드의 양이 많을 경우 private한 메소드를 이용해서 메소드를 분리하거나 별도의 객체를 만들어 사용하도록 한다.	
		4.2		- 클래스의 코드 길이가 너무 길어진다면, 해당 클래스가 몇개의 클래스로 분리될 수 있는지 고민한다.	
		4.3		- 변수는 최대한 덜 사용하고, 최대한 가볍게 만들어 가독성을 높이도록 한다.	
		4.4		- 조건문의 경우 긍정적이고 흥미로운 (주 흐름에 해당하는) 경우가 앞쪽에 위치하도록 한다.	
		4.5		- 상황연산자, do-while문정은 코드 가독성을 떨어트리니 되도록 사용하지 않는다.	
		4.6		- 블록이 너무 많이 중첩되면 코드를 읽기 어려워진다. 블록을 private메소드로 추출할 수 있는지 고민한다.	
5		5.1	프로젝트 생성	- 코드는 빈줄을 이용해 커다란 블록을 논리적인 단위로 구분한다.	
		5.2		- 코드는 들여쓰기를 잘 지키도록 한다.	
6		6.1	프로젝트 구조	- 필요하지 않은 코드는 제거한다.	
		6.2		- pom.xml 파일에 알맞은 의존성을 설정할 수 있어야 한다.	
7		7.1	Web API	- gitignore파일이 추가된 요구사항대로 알맞게 작성되어 있어야 한다.	
		7.2		- Controller, Service, Repository를 사용하여 구현되어 있어야 한다.	
		7.3		- Controller에서 Service를 Service에서 Repository의 기능을 호출할 수 있지만, 그 반대는 허용되지 않는다.	
		7.4		- 전체, 카테고리별 상품정보를 제공하는 Web Api를 제공한다.	
		7.5		- 해당 Web API는 카테고리 id를 파라미터로 받을 수 있다.	
		7.6		- 카테고리 id를 받지 않을 경우엔 전체 상품정보를 제공한다.	
		7.7		- 해당 Web API는 순번을 파라미터로 받을 수 있다.	
		7.8		- 파라미터로 순번을 못 받을 경우 첫번째 데이터부터 4번째 데이터까지 제공한다.	

제원번호	코드 작성 규칙	규칙번호	규칙명	설명	진행사항
1	HTML, CSS	1.1	HTML,CSS 규칙기반 코딩	- 제공되는 HTML/CSS를 수정할 수 있으며, 제공되는 코드의 형태의 구조를 지나치게 수정하지 않도록 한다.	
		1.2		- 기존에 제공되는 class와 id규칙에 따라서 새로운 내용을 추가할 수 있다.	
2	Browser	2.1	브라우저 개발 규칙	- 상단 타이틀 이미지 영역의 애니메이션은 CSS3의 transition과 transform을 활용해서 구현해야 한다.	
		2.2		- 상단에 추가로 노출해야 하는 기타이미지는 Ajax요청을 통해 가져온다.	
		2.3		- DOMContentLoaded 이후에 모든 자바스크립트 로직이 동작되게 한다.	
3	JavaScript	3.1	JavaScript 객체리터럴 패턴사용	- Event delegation으로 처리할 수 있는 영역은 최대한 delegation방법으로 처리한다.	
		3.2	JavaScript 라이브러리 사용	- 전체 코드는 객체리터럴 패턴으로 구현해야하고, 더불어 전역변수를 최소화 해야 한다. 한개이상 객체리터럴을 사용할 수 있다.	
		3.3		- Templating 작업은 handlebar 라이브러리를 사용해서 구현해야 한다.	
		3.4	JavaScript 클린코드	- 점기/펼치기 기능은 jQuery 라이브러리를 사용할 수 있다.	
		3.5		- 함수 하나에 여러개의 기능을 붙지 않고,함수를 여러개로 분리한다.	
		3.6		- 함수이름은 동사+명사이며 함수의 의도를 충분히 반영하고 있는가?	
		3.7		- 함수안의 내용은 이름에 어울리게 하나의 로직을 담고 있는가?	
		3.8		- 중복코드를 제거하고 공통부분을 공통함수로 만든다.	
		3.9		- 의도가 드러난 구현패턴, 의도가 표현되지 않은 코드를 최소한으로 줄이도록 한다.	

## Sequence Diagram

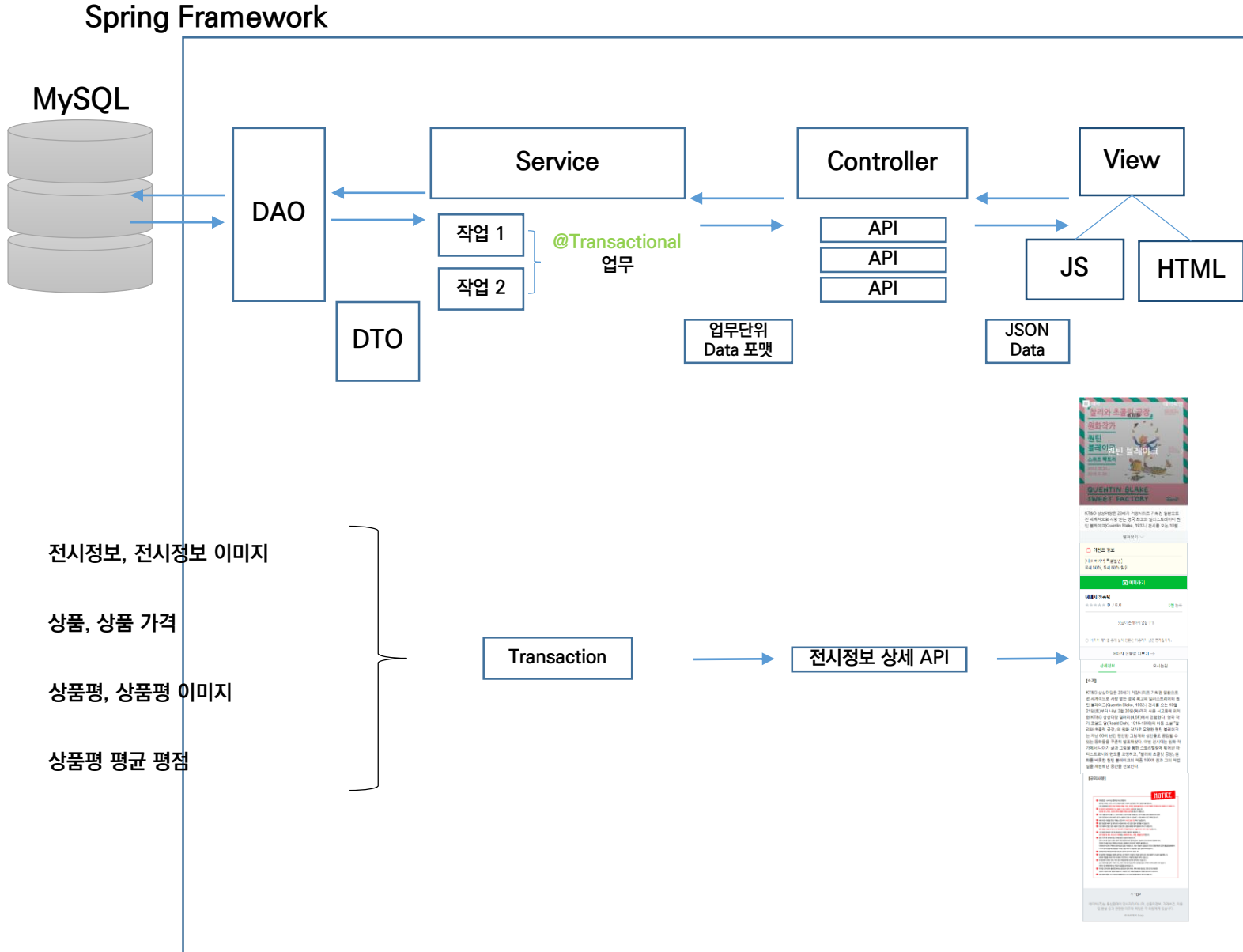
## Sequence Diagram



# 구현 순서

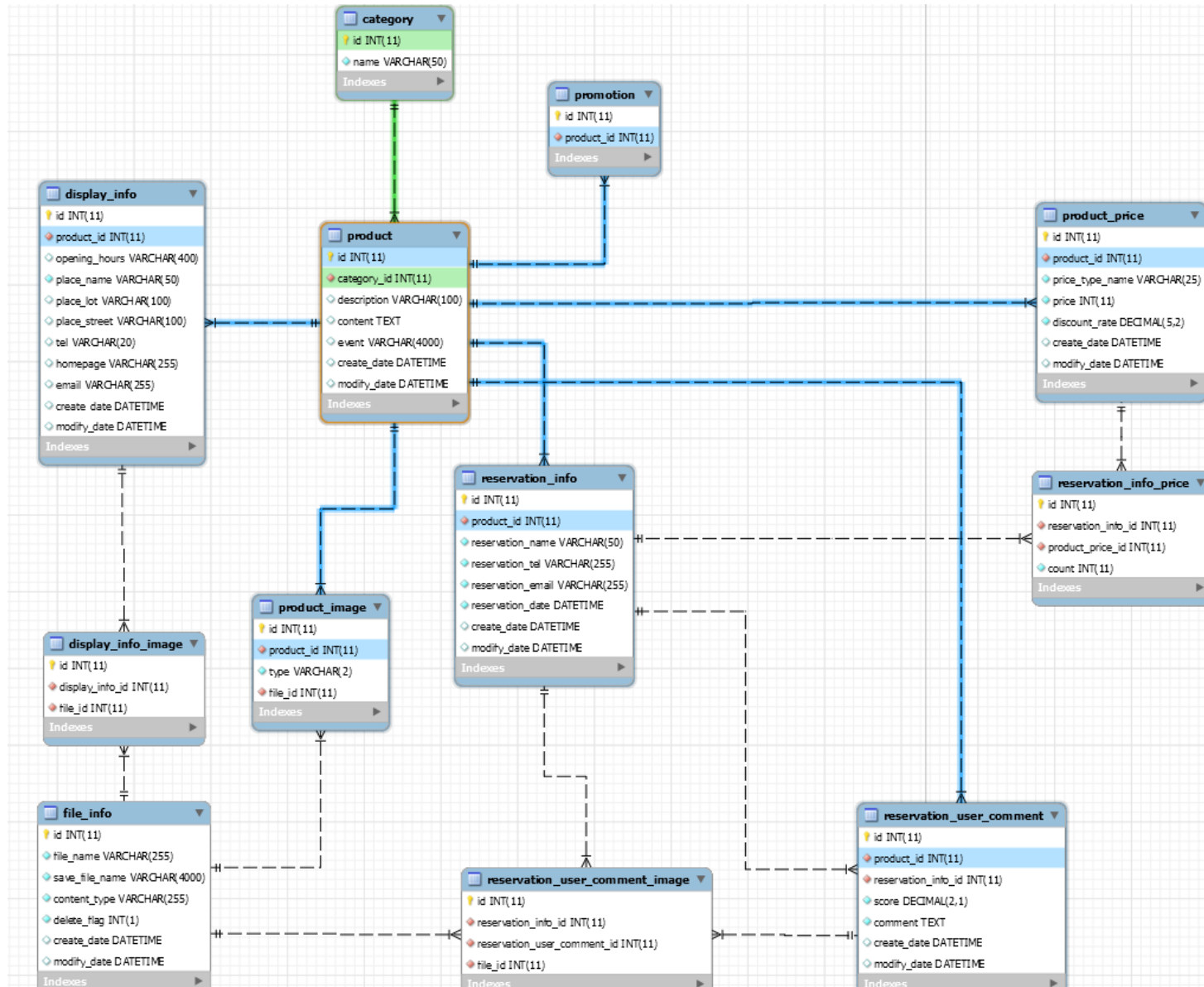
## Config

- DTO 작성
  - 전시정보
  - 전시정보 이미지
  - 상품이미지
  - 상품 가격
  - 상품 평
  - 상품 평 이미지
- DAO 작성
  - 전시정보
  - 전시정보 이미지
  - 상품 이미지
  - 상품 가격
  - 상품 평
  - 상품 평 이미지
  - 상품 평 평균 평점
- Service 작성
  - Transaction 처리
  - 전시정보 상세라는 업무로 처리
- Controller 작성
  - 업무 단위로 묶인 데이터를 반환
- JS 작성
  - 상세 페이지 로딩 시 상세 페이지의 데이터를 Ajax 방식으로 호출
- HTML 작성
  - 서버로부터 응답 받은 데이터를 템플릿을 이용하여 출력



# E-R Diagram

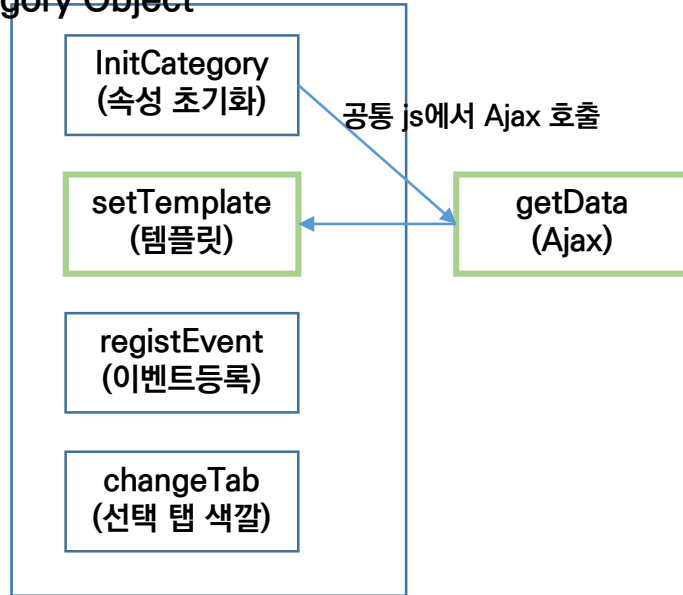
## Database Table





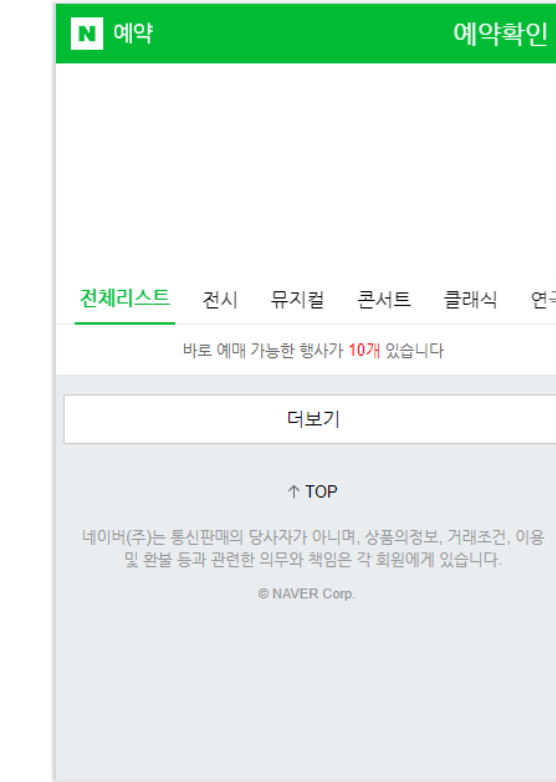
# 카테고리 UI 구현

## Category Object



```
▼ {items: Array(5)} ⓘ  
  ▼ items: Array(5)  
    ▶ 0: {id: 1, name: "전시", count: 10}  
    ▶ 1: {id: 2, name: "뮤지컬", count: 10}  
    ▶ 2: {id: 3, name: "콘서트", count: 10}  
    ▶ 3: {id: 4, name: "클래식", count: 10}  
    ▶ 4: {id: 5, name: "연극", count: 10}  
    length: 5  
    ▶ __proto__: Array(0)  
  ▶ __proto__: Object
```

> |



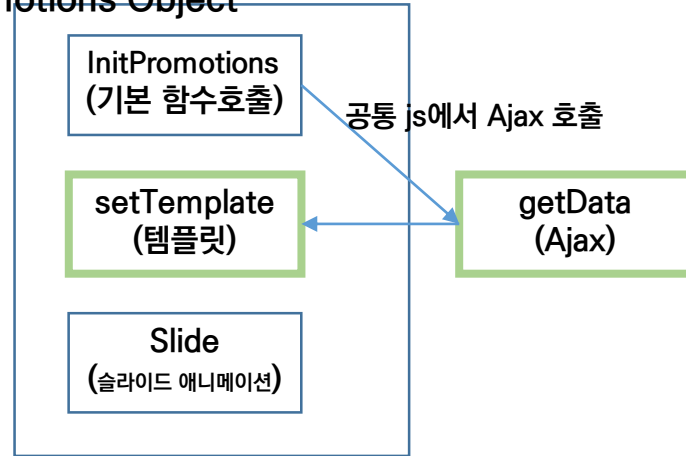
categories.js:37

## Categories.js

- Category Object
  1. 카테고리 UI 만들 때 필요한 속성을 초기화 함수
  2. 템플릿 만드는 함수  
(Ajax는 공통 js를 만들어 호출)
  3. 이벤트 등록 함수
  4. 탭을 클릭해서 selected된 TabUI 색 변화 함수
  5. 탭 클릭 시 Products 불러오는 함수

# 프로모션 UI 구현

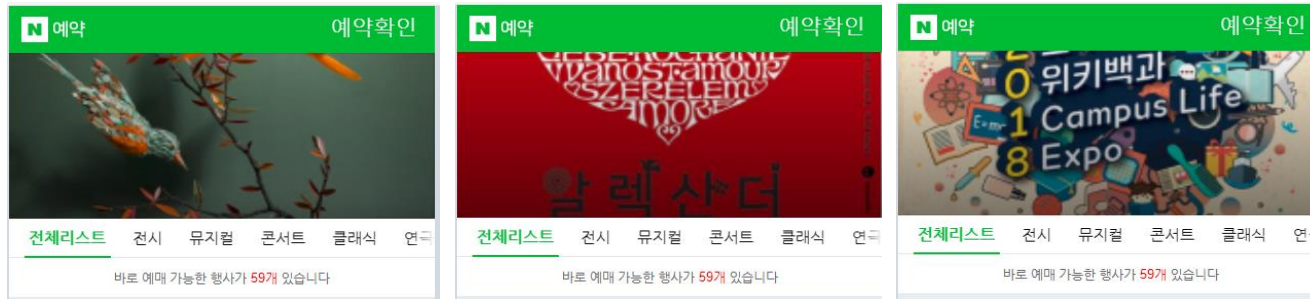
## Promotions Object



```

▼ (11) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]
  ▶ 0: {id: 1, productId: 1, categoryId: 1, categoryName: "전시", description: "Paper, Present:너를 위한 선물", ...}
  ▶ 1: {id: 2, productId: 5, categoryId: 1, categoryName: "전시", description: "알렉산더 지라드전", ...}
  ▶ 2: {id: 3, productId: 6, categoryId: 1, categoryName: "전시", description: "캠퍼스 라이프 엑스포", ...}
  ▶ 3: {id: 4, productId: 9, categoryId: 1, categoryName: "전시", description: "무인원화전", ...}
  ▶ 4: {id: 5, productId: 11, categoryId: 2, categoryName: "뮤지컬", description: "뮤지컬 모래시계", ...}
  ▶ 5: {id: 6, productId: 12, categoryId: 2, categoryName: "뮤지컬", description: "뮤지컬 출숙업", ...}
  ▶ 6: {id: 7, productId: 18, categoryId: 2, categoryName: "뮤지컬", description: "뮤지컬 - 카라마조프", ...}
  ▶ 7: {id: 8, productId: 22, categoryId: 3, categoryName: "콘서트", description: "노브레인의 연말 콘서트 [락이 먼저다]", ...}
  ▶ 8: {id: 9, productId: 34, categoryId: 4, categoryName: "클래식", description: "카마리타 크리스마스 특집 콘서트*", ...}
  ▶ 9: {id: 10, productId: 41, categoryId: 5, categoryName: "연극", description: "죽여주는이야기", ...}
  ▶ 10: {id: 11, productId: 44, categoryId: 5, categoryName: "연극", description: "여바웃 머니*", ...}
    length: 11
    __proto__: Array(0)
  
```

promotions.js:34

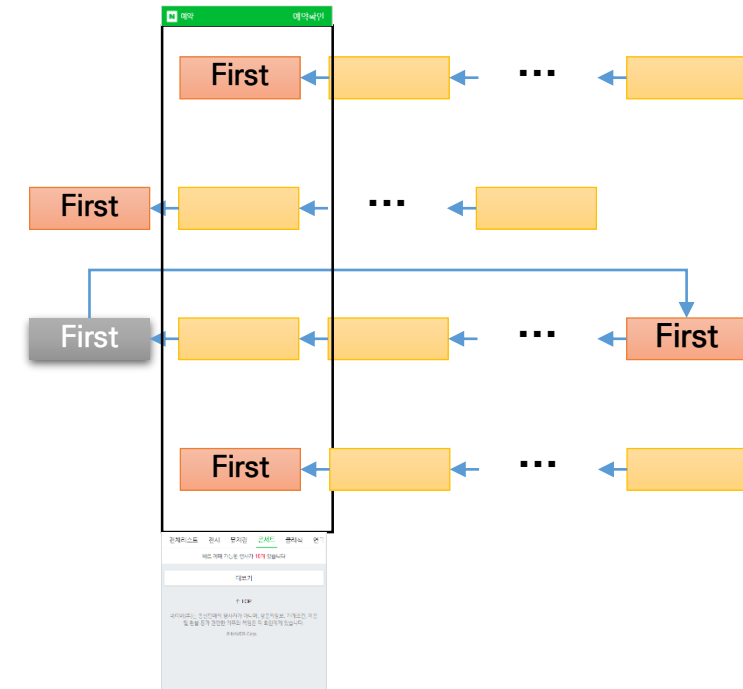


## Promotions.js

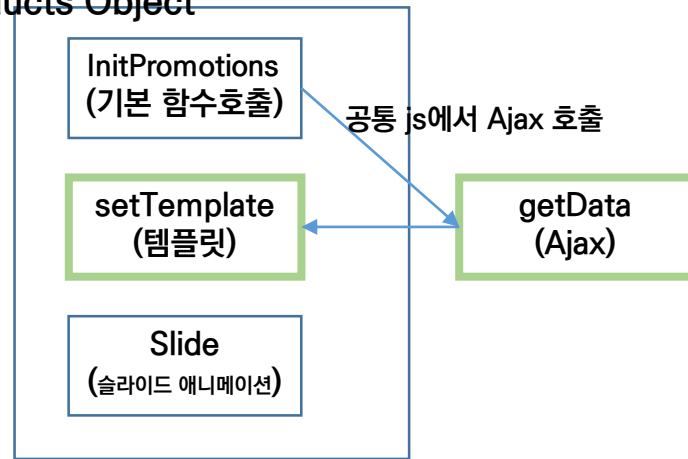
### • Promotions Object

1. 프로모션 UI 만들 때 필요한 속성을 초기화 함수
2. 템플릿 만드는 함수  
(Ajax는 공통 js를 만들어 호출)
3. 슬라이드 애니메이션 적용  
(캐러셀 슬라이드는 나중에..)

Slide 구현방식 Queue 구조로 동작  
맨 앞 노드가 이동된 뒤 마지막 노드 뒤로 붙음



## Products Object



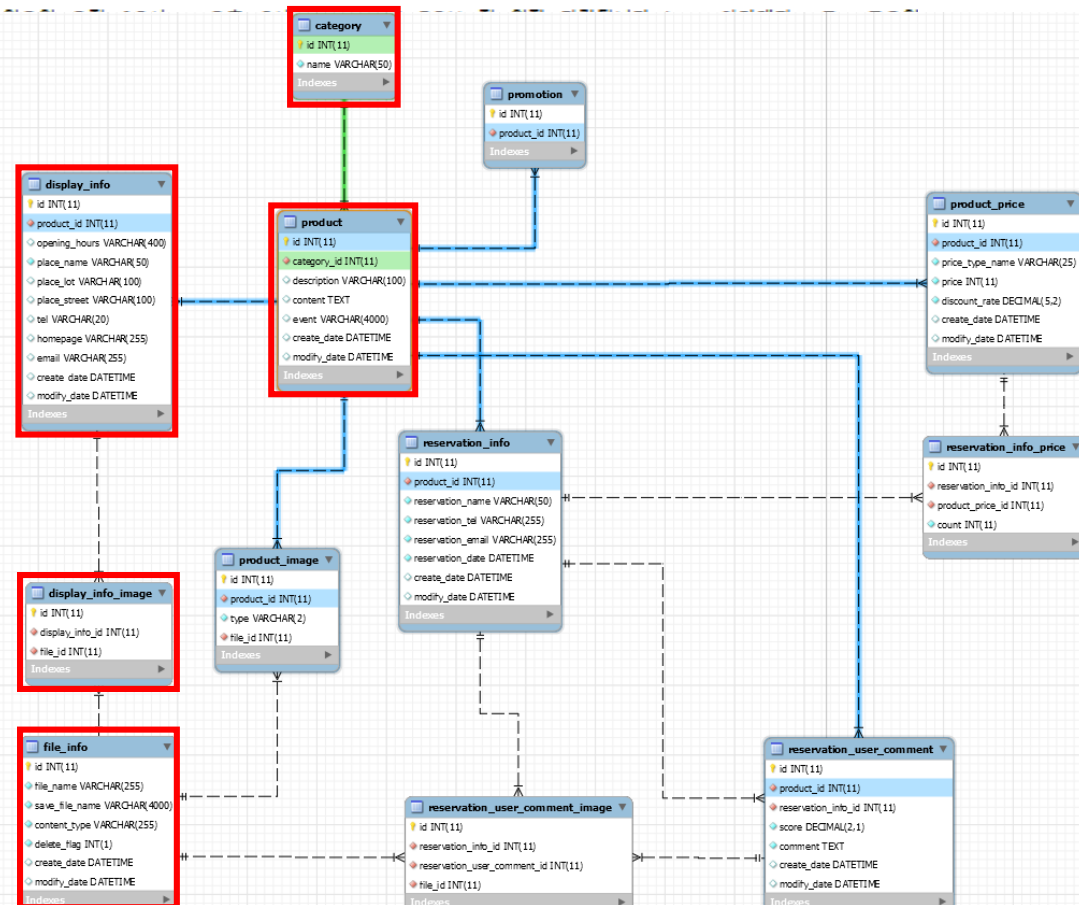
## Products.js

### • Products Object

1. 프로모션 UI 만들 때 필요한 속성을 초기화 함수
2. 템플릿 만드는 함수  
(Ajax는 공통 js를 만들어 호출)
3. 슬라이드 애니메이션 적용  
(캐러셀 슬라이드는 나중에..)

# DisplayInfo API 구현

```
{
  "displayInfo": {
    "productId": 1,
    "categoryId": 1,
    "displayInfoId": 1,
    "categoryName": "전시",
    "productDescription": "Paper, Present:너를 위한 선물",
    "productContent": "대림미술관은 오는 12월 7일부터 2018년 5월 27일까지 세계적인 아티스트들의 섬세한 감각과 아날로그적 소재인 종이와 만",
    "productEvent": "",
    "openingHours": "- 관람시간: 화, 수, 금",
    "placeName": "대림미술관",
    "placeLot": "서울특별시 종로구 동의동 35-1",
    "placeStreet": "서울특별시 종로구 자하문로4길",
    "telephone": "02-6403-9961",
    "homepage": "daelimuseum.org",
    "email": "",
    "createDate": "2018-11-14T11:31:42",
    "modifyDate": "2018-11-14T11:31:42"
  }
}
```

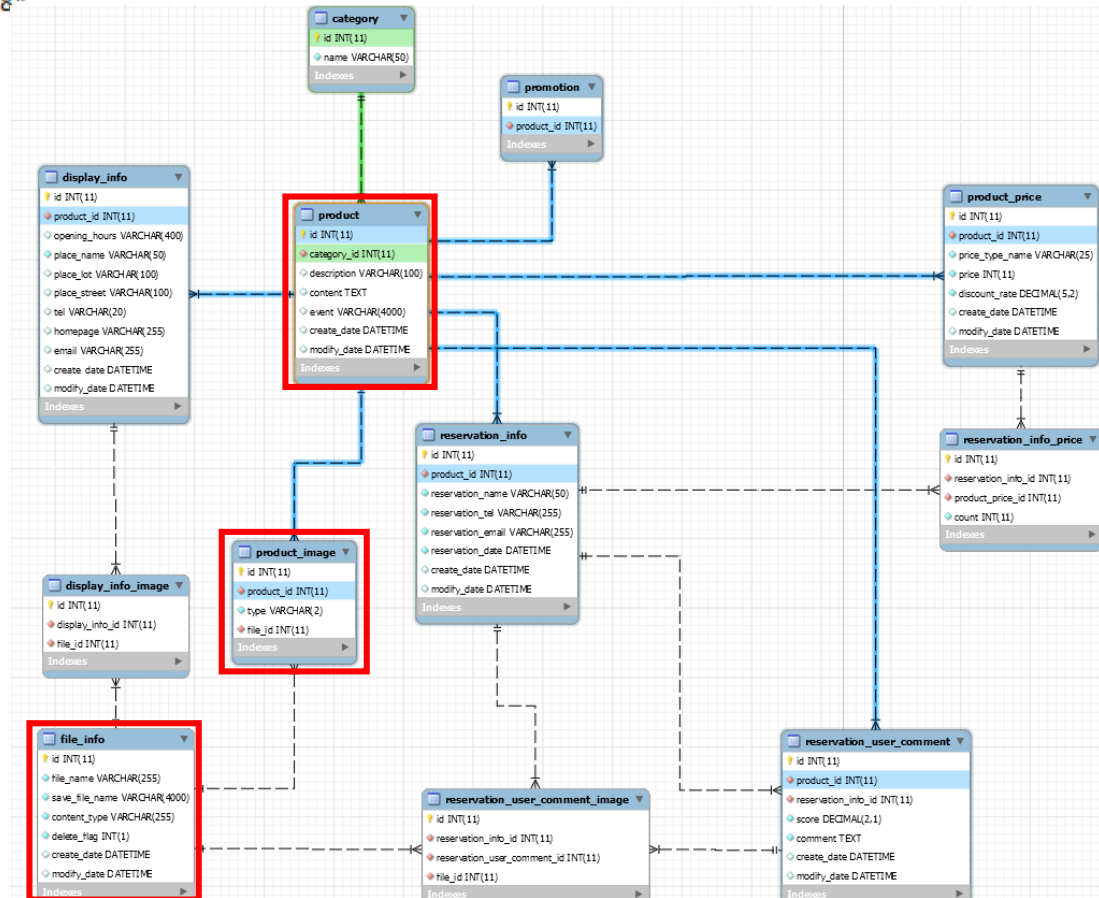


## ProductAPI

- displayInfo: **Object** (Product → DisplayInfo)
  - productId
  - categoryId
  - displayInfoId
  - categoryName
  - productDescription (속성 수정)
  - productContent (속성 수정)
  - productEvent
  - openingHours
  - placeName
  - placeLot
  - placeStreet
  - telephone
  - homepage
  - email
  - createDate
  - modifyDate
- productImages: **Array**
- displayInfoImages
- comments
- averageScore
- productPrices

# DisplayInfo API 구현

```
"productImages": [  
  {  
    "productId": 1,  
    "productImageId": 2,  
    "type": "ma",  
    "fileInfoId": 61,  
    "fileName": "1_ma_2.png",  
    "saveFileName": "img/1_ma_2.png",  
    "contentType": "image/png",  
    "deleteFlag": false,  
    "createDate": "2018-11-14T11:32:08",  
    "modifyDate": "2018-11-14T11:32:08"  
  },  
],
```

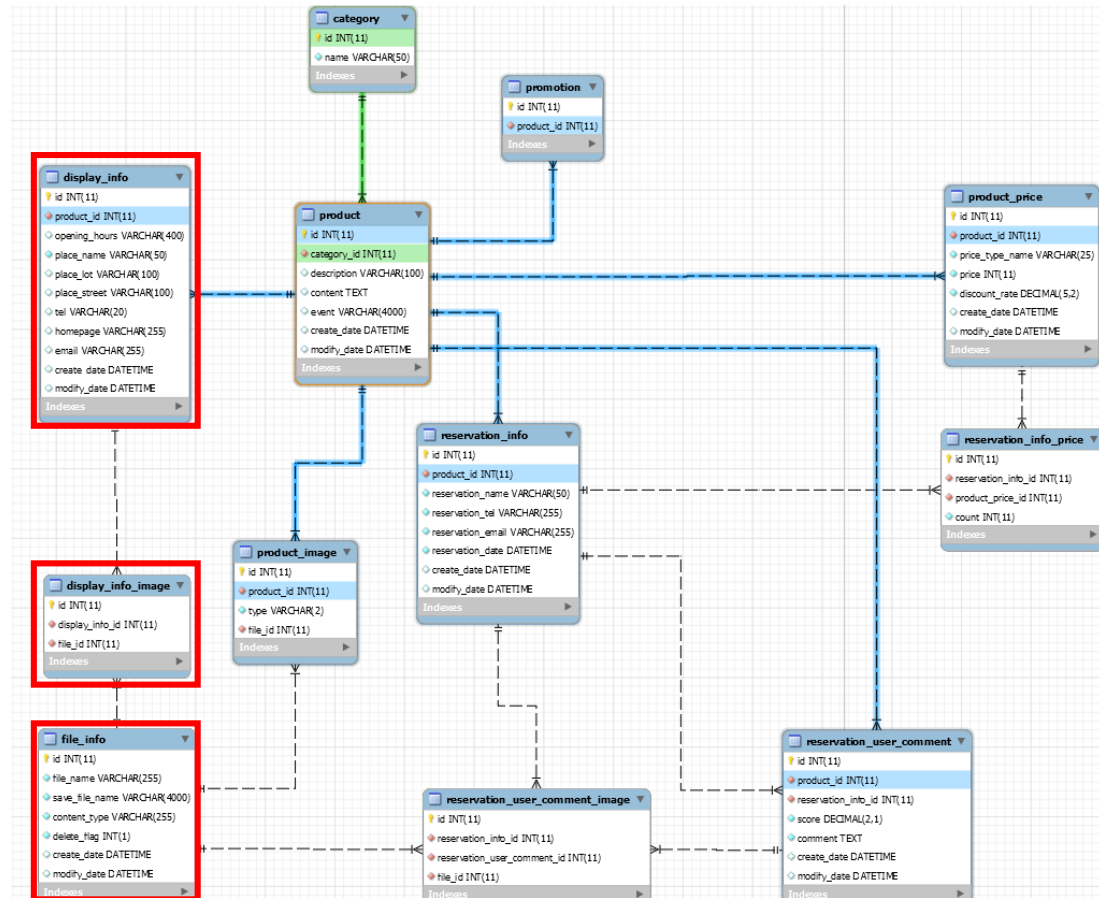


## ProductAPI

- displayInfo: **Object** (Product → DisplayInfo)
- productImages: **Array**
  - productId
  - productImageId
  - type
  - fileInfoId
  - fileName
  - saveFileName
  - contentType
  - deleteFlag (**int** → **boolean**)
  - createDate
  - modifyDate
- displayInfoImages
- comments
- averageScore
- productPrices

# DisplayInfo API 구현

```
"displayInfoImage": {  
  "displayInfoImageId": 1,  
  "displayInfoId": 1,  
  "fileId": 1,  
  "fileName": "1_map_1.png",  
  "saveFileName": "img_map/1_map_1.png",  
  "contentType": "image/png",  
  "deleteFlag": false,  
  "createDate": "2018-11-14T11:31:04",  
  "modifyDate": "2018-11-14T11:31:04"  
},
```



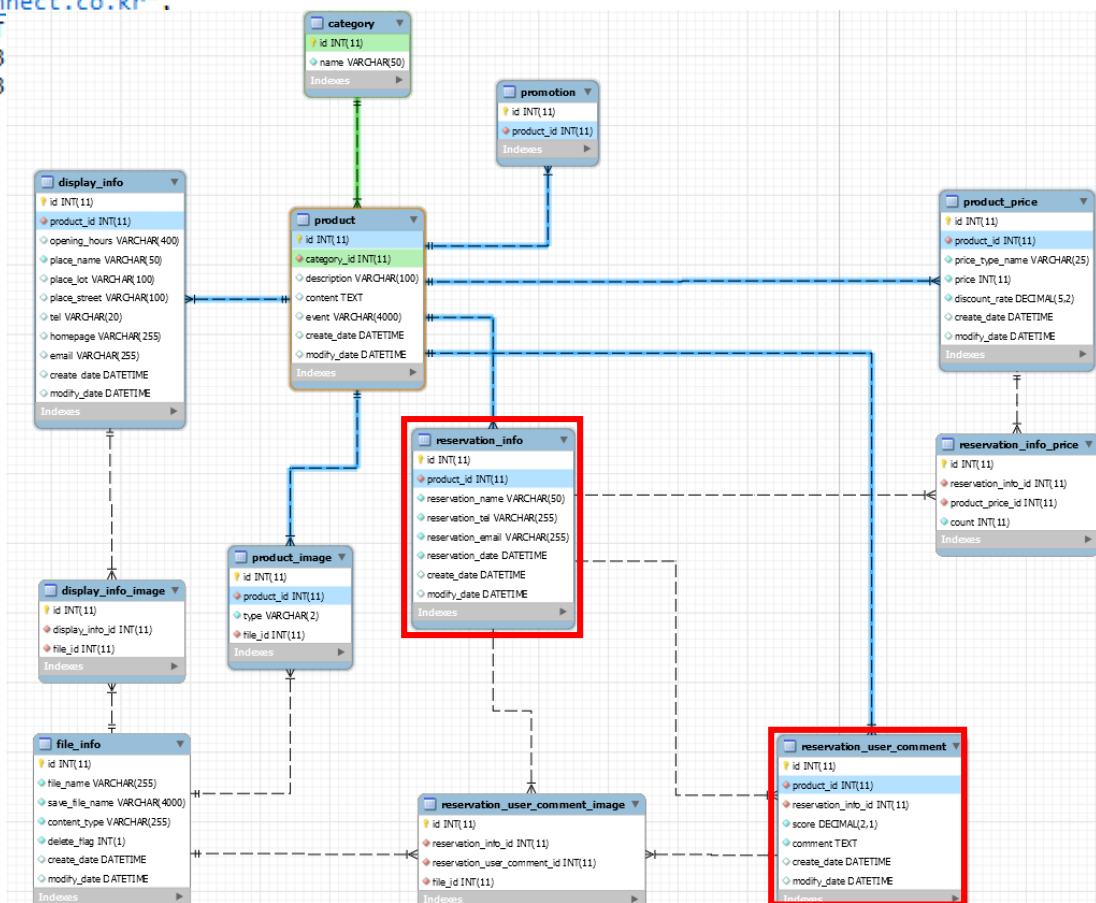
## ProductApiController.java

- displayInfo: **Object**
- productImages: **Array**
- displayInfoImage
  - displayInfoImageId (id → displayInfoImageId)
  - displayInfoId
  - fileId
  - fileName
  - saveFileName
  - contentType
  - deleteFlag
  - createDate
  - modifyDate
- comments
- averageScore
- productPrices



## DisplayInfo API 구현

```
{
  "commentId": 15,
  "productId": 1,
  "reservationInfoId": 15,
  "score": 5,
  "comment": "즐거움!!!",
  "reservationName": "도로시",
  "reservationTelephone": null,
  "reservationEmail": "dorosi@connect.co.kr",
  "reservationDate": "2018-11-14T
  "createDate": "2018-11-14T11:33
  "modifyDate": "2018-11-14T11:33
  "commentImages": []
},
```

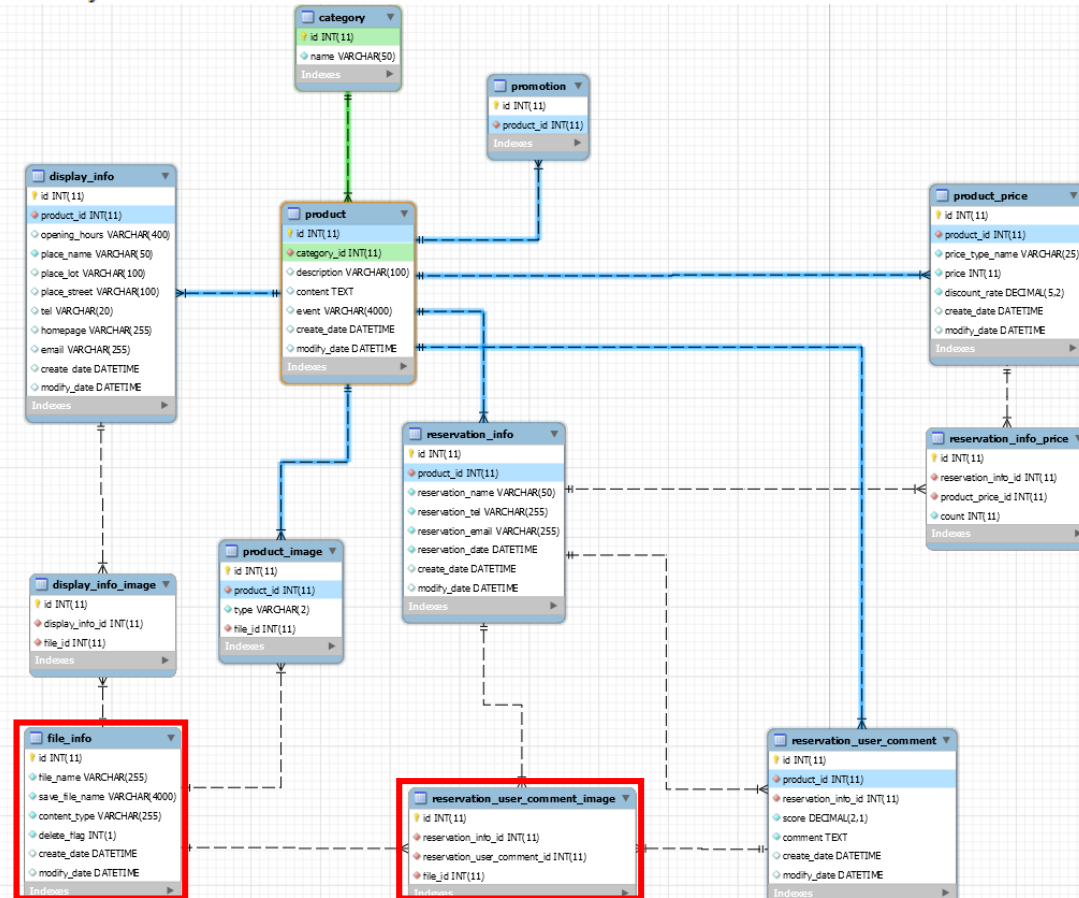


## ProductApiController.java

- displayInfo: **Object**
- productImages: **Array**
- displayInfolImages: **Object**
- comments: **Array**
  - comment: **Array**
    - commentId
    - productId
    - reservationInfol
    - score
    - comment
    - reservationName
    - reservationTelephone
    - reservationEmail
    - reservationDate
    - createDate
    - modifyDate
  - commentImages: **Array**
    - imageId
    - reservationInfol
    - reservationUserCommentId
    - fileId
    - fileName
    - saveFileName
    - contentType
    - deleteFlag
    - createDate
    - modifyDate
- averageScore
- productPrices

## DisplayInfo API 구현

```
{
  "commentId": 1,
  "productId": 1,
  "reservationInfoId": 1,
  "score": 4,
  "comment": "좋았어요.",
  "reservationName": "김진수",
  "reservationTelephone": null,
  "reservationEmail": "kimjinsu@connect.co.kr",
  "reservationDate": "2018-11-14T11:32:54",
  "createDate": "2018-11-14T11:33:11",
  "modifyDate": "2018-11-14T11:33:11",
  "commentImages": [
    {
      "imageId": 1,
      "reservationInfoId": 1,
      "reservationUserCommentId": 1,
      "fileId": 1,
      "fileName": "1_map_1.png",
      "saveFileName": "img_map/1_map",
      "contentType": "image/png",
      "deleteFlag": false,
      "createDate": "2018-11-14T11:33:11",
      "modifyDate": "2018-11-14T11:33:11"
    }
  ]
}
```



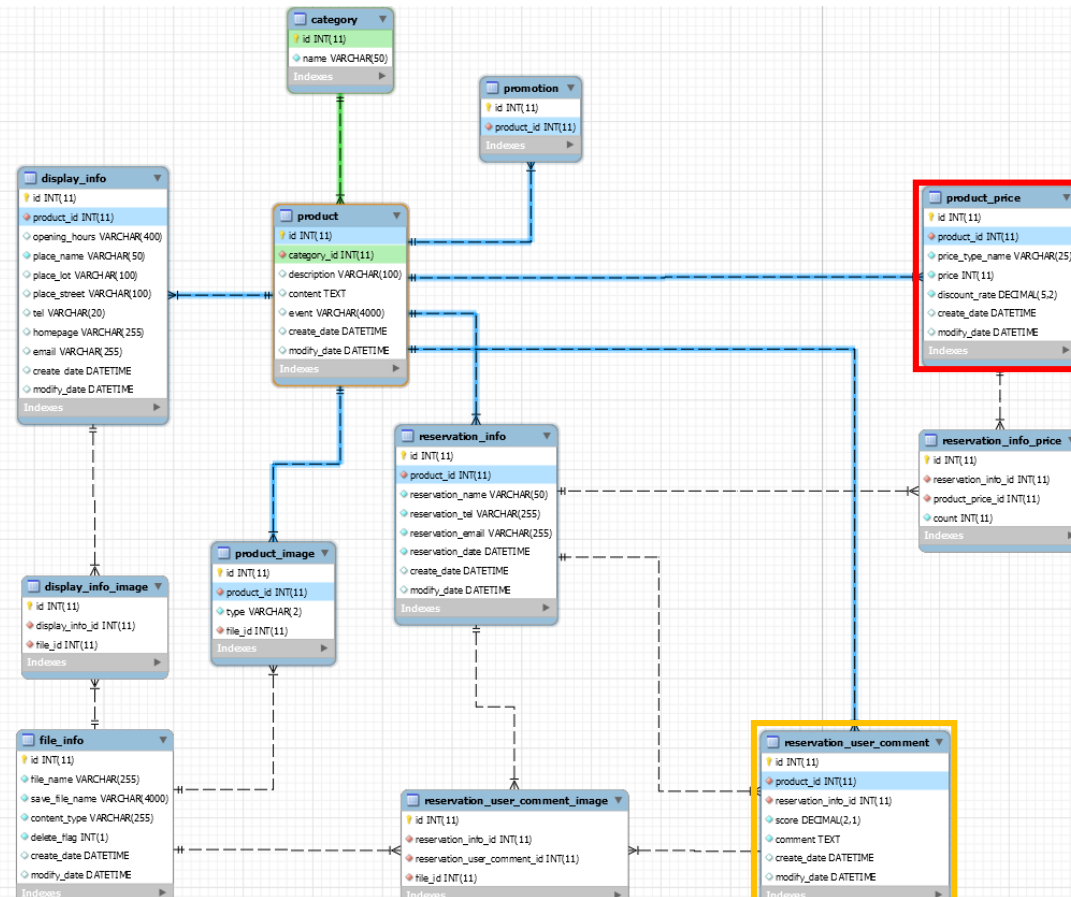
## CommentImages

- displayInfo: **Object**
- productImages: **Array**
- displayInfolImages: **Object**
- comments: **Array**
  - comment: **Array**
    - commentId
    - productId
    - reservationInfol
    - score
    - comment
    - reservationName
    - reservationTelephone
    - reservationEmail
    - reservationDate
    - createDate
    - modifyDate
  - commentImages: **Array**
    - imageId
    - reservationInfol: **FK**
    - reservationUserCommentId
    - fileId
    - fileName
    - saveFileName
    - contentType
    - deleteFlag
    - createDate
    - modifyDate
- averageScore
- productPrices



# DisplayInfo API 구현

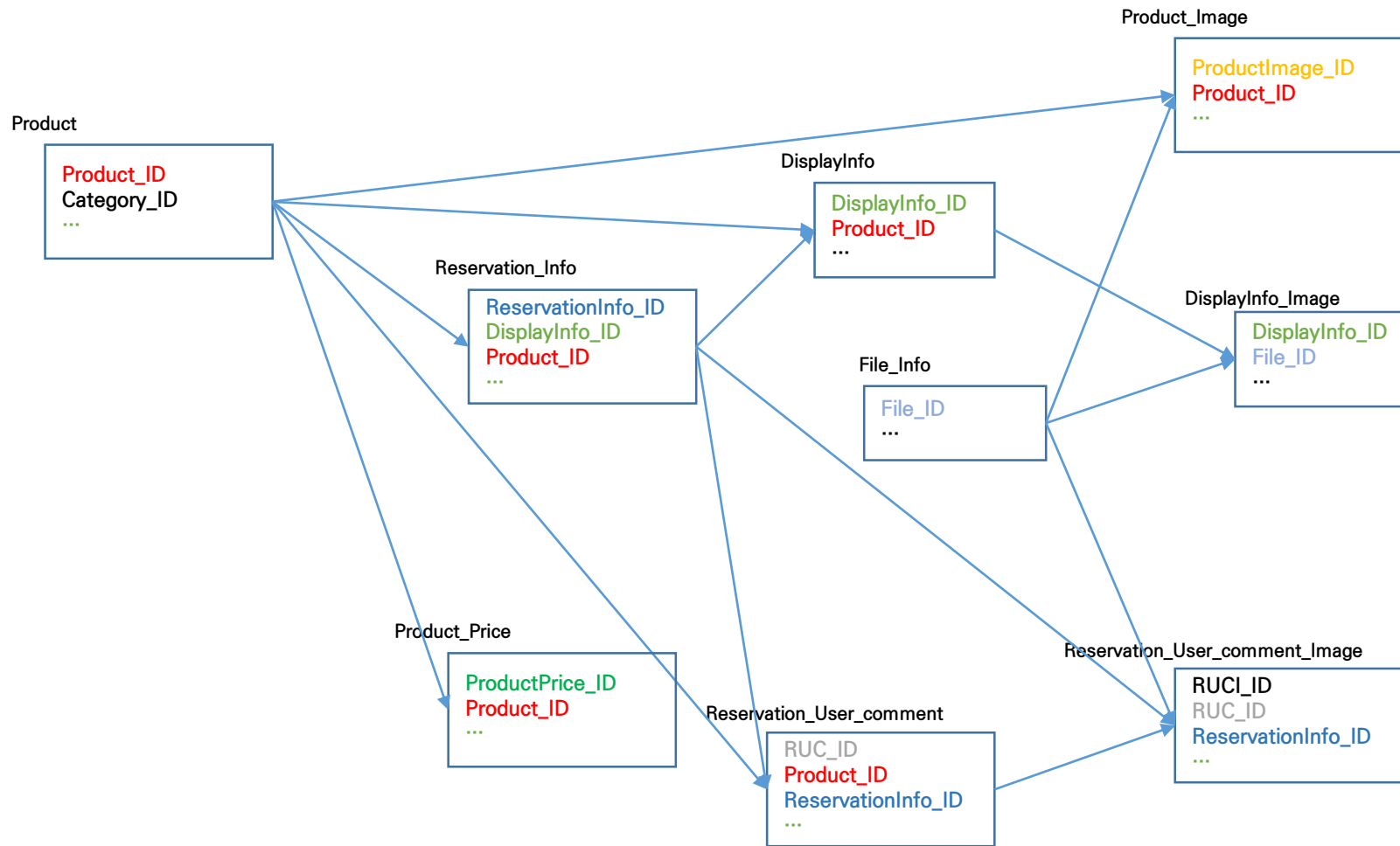
```
"averageScore": 3.66667,
"productPrices": [
  {
    "productPriceId": 3,
    "productId": 1,
    "priceTypeName": "B",
    "price": 2000,
    "discountRate": 50,
    "createDate": "2018-11-14T11:11:11",
    "modifyDate": "2018-11-14T11:11:11"
  },
  {
    "productPriceId": 2,
    "productId": 1,
    "priceTypeName": "Y",
    "price": 3000,
    "discountRate": 33,
    "createDate": "2018-11-14T11:11:11",
    "modifyDate": "2018-11-14T11:11:11"
  },
  {
    "productPriceId": 1,
    "productId": 1,
    "priceTypeName": "A",
    "price": 6000,
    "discountRate": 20,
    "createDate": "2018-11-14T11:11:11",
    "modifyDate": "2018-11-14T11:11:11"
  }
]
```



## ProductApiController.java

- displayInfo: **Object**
- productImages: **Array**
- displayInfoImages: **Object**
- comments: **Array**
- averageScore: **double**
- productPrices: **Array**
  - productPriceId
  - productId
  - priceTypeName
  - price
  - discountRate
  - createDate
  - modifyDate

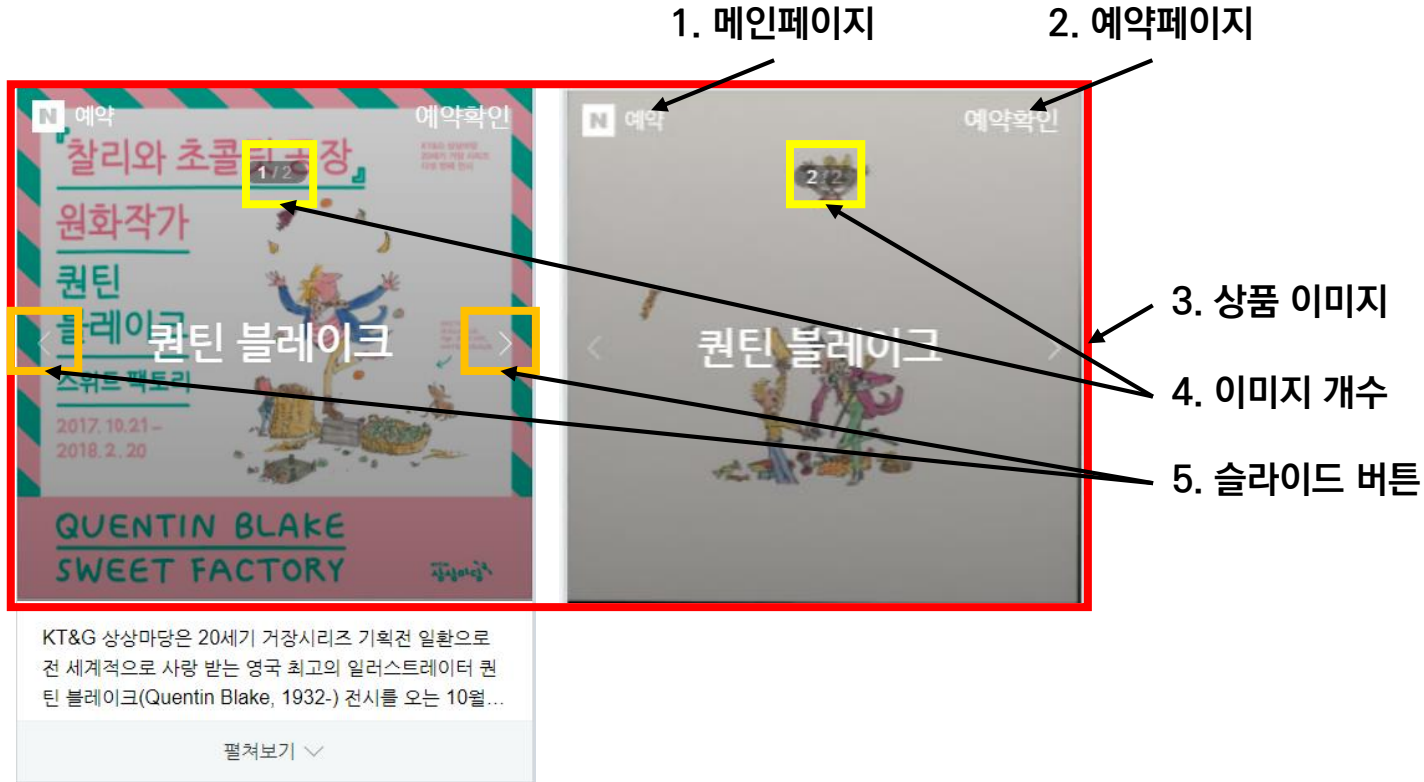
# DisplayInfo API 구현



## DisplayInfo

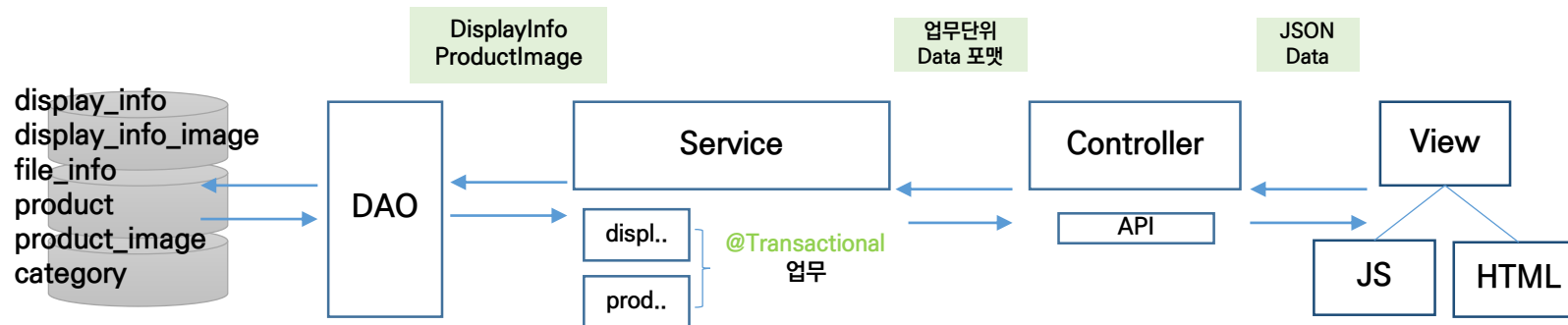
- displayInfo: **Object**
- productImages: **Array**
- displayInfolImages: **Object**
- comments: **Array**
- averageScore: **double**
- productPrices: **Array**
  - productPriceId
  - productId
  - priceTypeName
  - price
  - discountRate
  - createDate
  - modifyDate

## 상세 페이지

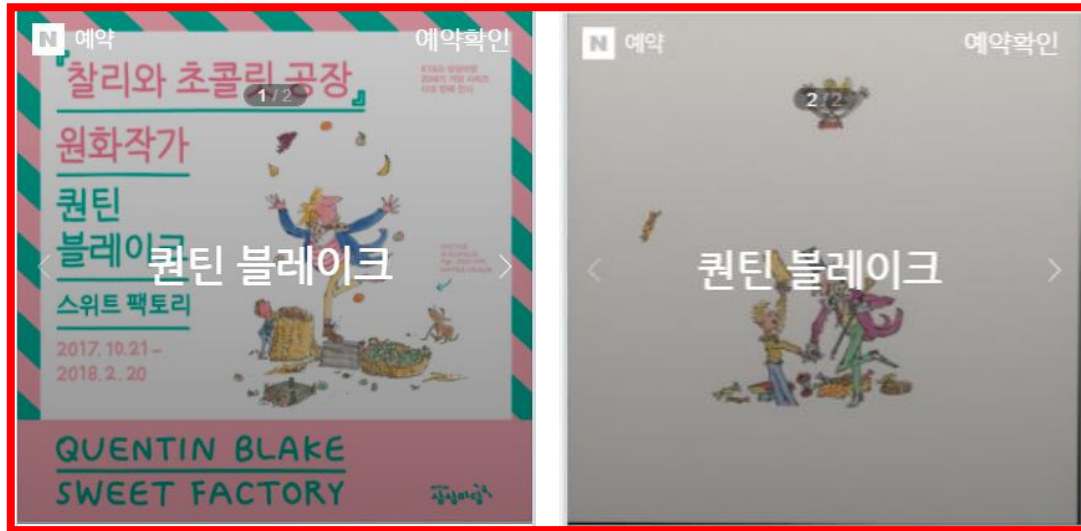


## 상단영역 (1)

- 페이지 이동
  1. 메인페이지 이동
  2. 예약페이지 이동
- 슬라이드
  1. 메인이미지(type = 'ma') 이외에 기타이미지(type = 'et')가 존재할 경우 이미지 2개 출력
  2. 출력된 이미지의 개수를 표시, 슬라이드 시 출력된 이미지의 번호를 출력
  3. 이미지가 2개일 경우 슬라이드 기능 활성화
- 펼쳐보기 & 접기



## 상세 페이지



KT&G 상상마당은 20세기 거장시리즈 기획전 일환으로 전 세계적으로 사랑 받는 영국 최고의 일러스트레이터 퀸틴 블레이크(Quentin Blake, 1932-) 전시를 오는 10월...

펼쳐보기 ▾

KT&G 상상마당은 20세기 거장시리즈 기획전 일환으로 전 세계적으로 사랑 받는 영국 최고의 일러스트레이터 퀸틴 블레이크(Quentin Blake, 1932-) 전시를 오는 10월 21일(토)부터 내년 2월 20일(화)까지 서울 서교동에 위치한 KT&G 상상마당 갤러리(4,5F)에서 진행한다. 영국 작가 로알드 달(Roald Dahl, 1916-1990)의 아동 소설 '찰리와 초콜릿 공장' 외 원화 작가로 유명한 퀸틴 블레이크는 지난 60여 년간 편안한 그림체와 성인들도 공감할 수 있는 동화들을 꾸준히 발표해왔다. 이번 전시에는 원화 작가에서 나아가 글과 그림을 통한 스토리텔링에 뛰어난 아티스트로서의 면모를 조명하고, '찰리와 초콜릿 공장' 원화를 비롯한 퀸틴 블레이크의 작품 180여 점과 그의 작업실을 재현해낸 공간을 선보인다.

접기 ▲

1. 상품 설명 펼쳐보기 & 접기

## 상단영역 (1)

- 페이지 이동
- 슬라이드
- 펼쳐보기 & 접기
  1. 상품 설명에 대한 내용을 펼쳐보기 & 접기

# 상세 페이지

## 2. 상품평의 평균 평점

예매자 한줄평

★★★★★ 3.7 / 5.0

15건 등록

**Paper, Present:**너를 위한 선물  
즐거움!!!  
5 | doro\*\*\*\* | 2018.12.31. 방문

**Paper, Present:**너를 위한 선물  
또 가고 싶어.  
5 | kimk\*\*\*\* | 2018.12.31. 방문

**Paper, Present:**너를 위한 선물  
재미남.  
4 | kimm\*\*\*\* | 2018.12.31. 방문

네이버 예약을 통해 실제 방문한 이용자가 남긴 평가입니다.

예매자 한줄평 더보기 →

## 1. 상품 설명 펼쳐보기 & 접기

🎁 이벤트 정보

[네이버예약 특별할인]  
R석 50%, S석 60% 할인

예매하기

예매자 한줄평

★★★★★ 0 / 5.0

0건 등록

댓글이 존재하지 않습니다.

네이버 예약을 통해 실제 방문한 이용자가 남긴 평가입니다.

예매자 한줄평 더보기 →

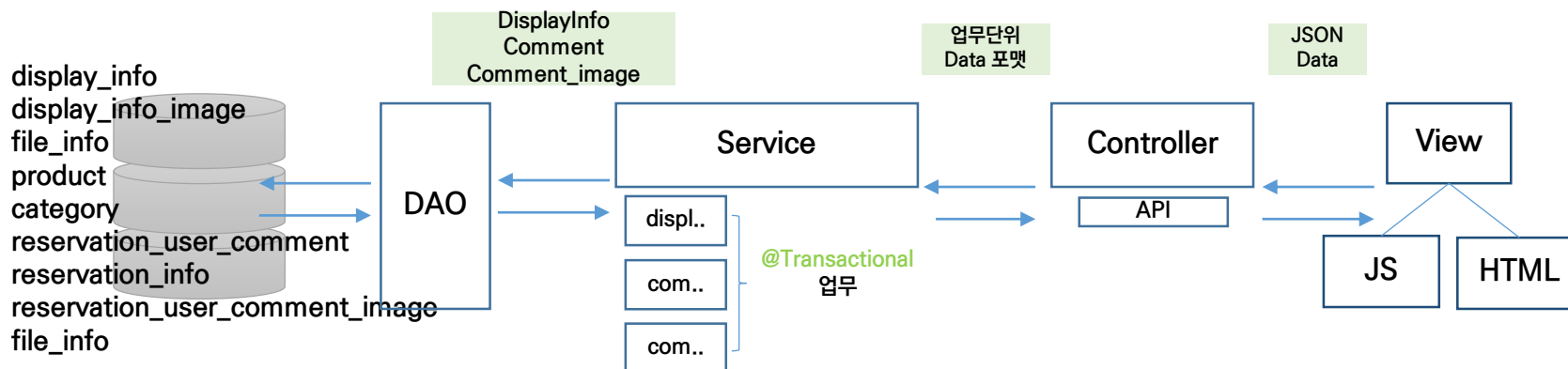
## 3. 해당 상품의 총 상품 평 개수

## 4. 해당 상품의 상품 평 (3개)

## 5. 해당 상품의 총 상품 평

## 중단 영역

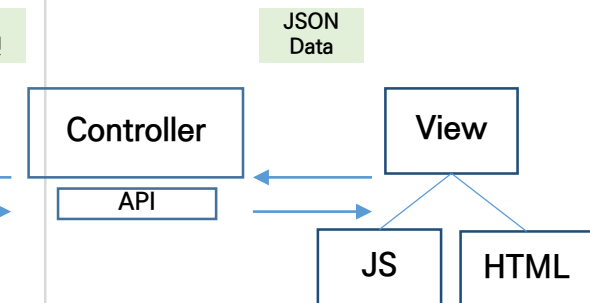
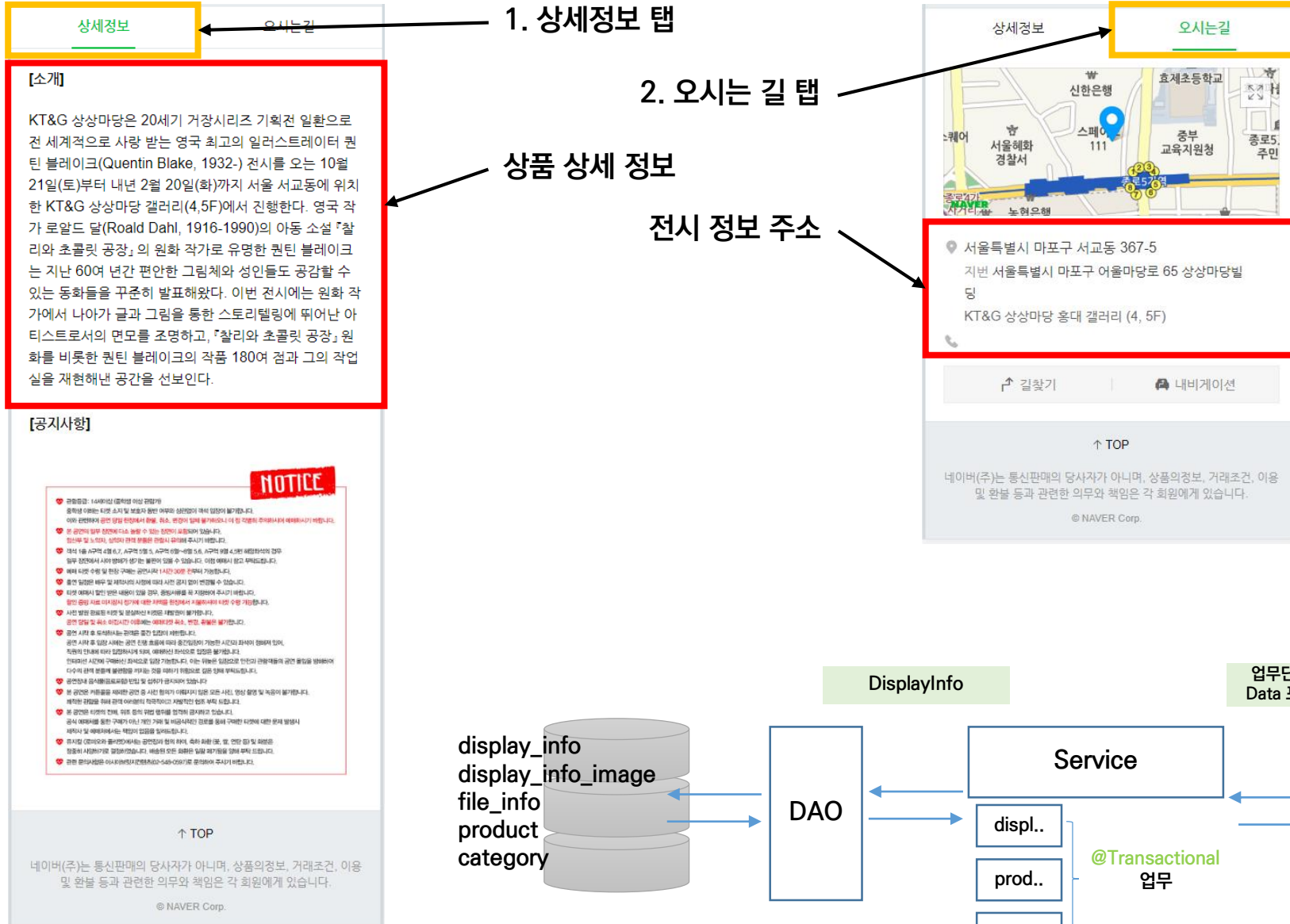
- 예매하기
  - 예약페이지 이동
- 예매자 평점
  - 상품평의 평균 평점
- 상품평 개수
  - 상품평의 총 개수
- 상품평
  - 현 상품의 예매 한 줄 평을 3개 출력
- 예매자 한줄평 더보기
  - 현 상품의 예매자 한 줄 평을 모두 출력하는 페이지로 이동



## 상세 페이지

## 하단 영역

1. 상세정보 탭
  - 상세정보
2. 오시는 길 탭
3. 상품 상세 정보
4. 전시정보 주소



- 예매자 한줄평 수정
  - 평점 소수점 표현
    - ① Javascript 함수 toFixed(1)
  - 시간 '연.월.일' 표현
    - ① 쿼리 수정 date\_format('date', '%Y-%m-%d')
  - 이미지
    - ① 쿼리 수정  
조인 테이블 수정

## 요구사항

예매자 한줄평

★★★★☆ 4.2 / 5.0

52건 등록

뮤지컬 로미오와 줄리엣

2층이어서 걱정했는데 꽤잘보여서 좋았습니  
다 고미오 너무 맛있었습니다 사진은 커튼콜  
때 찍었습니다 끝나고 퇴근길도 봐서 너무 좋  
았어요

4.0

dbfl\*\*\*\* | 2017.3.5. 방문

뮤지컬 로미오와 줄리엣

너무 재밌게봤구요~  
마지막공연 후 뒷풀이도 잘봤습니다

5.0

yyck\*\*\*\* | 2017.3.5. 방문

뮤지컬 로미오와 줄리엣

좋은 공연이었습니다.  
머큐쇼역할 하신분의 열창이 기억에 남는 반면에,,, 로미오는  
별로 기억에 남지 않네요..

4.0

xero\*\*\*\* | 2017.3.4. 방문

네이버 예약을 통해 실제 방문한 이용자가 남긴 평가입니다.

예매자 한줄평 더보기 →





## 상세 페이지

이미지 2개



이미지 1개



## 중단 영역

- 이미지 요구사항
  - 이미지 무한슬라이드로 변경
- 이미지 1개
  - 슬라이드 버튼 삭제
- 이미지 2개 ('ma', 'et')
  - 슬라이드 버튼
  - 무한슬라이드로 구현
  - default 'ma' 이미지, 'et' 추가 이미지