

# 예약: 메인페이지

## 프로젝트 개발과정 참고

### 1. maven 프로젝트를 생성합니다.

groupId 와 artifactId 는 임의로 지정합니다.

### 2. MySQL 에서 프로젝트에 사용할 database 와 사용자 계정을 생성합니다.

생성한 데이터베이스와 계정정보는 src/main/resources/application.properties 파일에 다음과 같이 설정합니다.

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://domain:port/dbName?
useUnicode=true&characterEncoding=utf8
spring.datasource.username=dbUserName
spring.datasource.password=dbPasswd
```

### 3. 생성한 데이터베이스에 접속하여 주어진 sql을 실행합니다.

먼저 ddl.sql의 내용을 실행하여 테이블을 생성하고, dml.sql의 내용을 실행하여 샘플 데이터를 추가합니다. [코드 바로가기](#)

### 4. sample이미지가 있는 압축파일인 img.zip을 webapp 폴더에 압축 해제합니다.

webapp폴더에 img와 img\_map 폴더 아래에 샘플 이미지가 위치합니다.

### 5. Spring MVC, Spring JDBC를 사용하기 위한 Spring설정 파일들을 작성합니다.

6. 샘플 데이터를 읽어 들여 메인화면을 출력하기 위한 DTO, Controller, Service, Repository를 맞게 작성합니다.

### 7. web API 스펙은 아래와 같습니다.

#### 카테고리 목록 구하기 API 스펙 & test

- GET (카테고리 목록 구하기): /api/categories

#### 상품 목록 구하기 API 스펙 & test

- GET (상품 목록 구하기): /api/products

#### 프로모션 정보 구하기 API 스펙 & test

- GET (프로모션 목록 구하기): /api/promotions

### 8. image type 설명

- th: thumbnail (썸네일 사진 - 상품리스트 혹은 프로모션 정보에서 보여주는 이미지)
- ma: main (메인 사진 - 상품 상세정보에서 보여주는 이미지)
- et: etc (기타 사진 - 상품 상세정보에서 추가적으로 보여주는 기타 이미지)

## BE

### 웹백엔드 기술요구사항

- 제공된 SQL을 이용해서 테이블을 생성하고, 샘플데이터를 입력합니다.
- maven을 이용해서 웹 어플리케이션 프로젝트를 작성합니다.
- 학습했던 것처럼 controller,service,dao로 레이어드 아키텍처를 구성합니다.
- springJDBC를 이용하여 주어진 테이블로부터 입력, 수정, 삭제, 조회하는 DAO와 DTO를 작성합니다.
- 서비스 인터페이스를 작성하고 해당 서비스 인터페이스에 비즈니스 메소드를 작성합니다.
- 서비스 인터페이스를 구현하는 클래스를 작성합니다.
- 해당 구현 클래스의 메소드에 적절한 트랜잭션에 관련된 애노테이션을 사용합니다.
- 클라이언트에게 Web API를 제공하기 위해 RestController 를 작성합니다.

## FE

### 웹프론트엔드 기술요구사항

- DOMContentLoaded 이후에 모든 자바스크립트 로직이 동작하게 합니다.
- 상단영역의 애니메이션은 CSS3의 transition과 transform 속성을 활용해서 구현해야 합니다.
- TABUI로 구성되는 카테고리 아이템이 노출되는 영역의 HTML은 카테고리별로 각각 만들지 않고 하나로 만들어 재사용합니다.
- 카테고리 탭을 선택할 때마다, Ajax 요청을 해서 데이터를 가져와야 합니다.
- 탭 메뉴 (전시/뮤지컬/콘서트 등)는 Event delegation으로 구현합니다.
- Template 코드를 javascript 함수 안에 보관하지 않고, 별도 분리시켜서 사용해야 합니다. (HTML에 script태그 안에 보관한다던가)
- 함수 하나에 여러 개의 기능을 넣지 않고, 함수를 여러 개로 분리합니다.

## 평가 기준표 - BE

### 기능

기준	세부항목
카테고리 노출 영역 (탭 UI)	- 전체리스트가 Ajax를 통해서 화면에 4개의 아이템이 노출된다. - 탭별로 전체갯수가 상단에 노출되어야 한다. - 각 아이템(전시상품)은 이미지/제목/장소/설명이 노출되어야 한다. - 탭을 누르면 다른 카테고리 콘텐츠 4개가 다시 노출된다. - 더보기를 누르면 4개씩 노출되어야 한다. 4개보다 적으면 적은 만큼 노출되어야 한다. - 더보기줄 데이터가 없다면 더보기는 사라진다. - TOP영역이 선택되면, 화면 맨 위로 이동한다

### 소스코드

기준	세부항목
JAVA - 이름규칙	- JAVA Naming Conventions 을 지킨다.(참고 1),(참고 2) - 클래스의 이름과 메소드의 이름은 직관적으로 작성하도록 한다. 클래스의 이름과 메소드의 이름만 보아도,어떤 기능을 가지고 있을지 어떤 내용이 구현되어 있을지 짐작할 수 있어야 한다. - 코드를 읽는 사람이 개념을 쉽게 파악할 수 있도록 읽기 쉬운 코드를 작성하도록 한다. 예를 들어 변수 이름을 구체적으로 작성하도록 한다.
JAVA - 중복된 코드 제거 및 코드 구조화	- 중복된 코드가 있다면, 별도의 메소드나 클래스로 분리하도록 한다. - 하나의 메소드가 너무 많은 코드를 담지 않는다. 코드의 양이 많을 경우 private한 메소드를 이용해서 메소드를 분리하거나 별도의 객체를 만들어 사용하도록 한다. - 클래스의 코드 길이가 너무 길어진다면, 해당 클래스가 몇개의 클래스로 분리될 수 있는지 고민한다. - 변수는 최대한 덜 사용하고, 최대한 가볍게 만들어 가독성을 높이도록 한다.
JAVA - 가독성	- 조건문의 경우 긍정적이고 흥미로운 (주 흐름에 해당하는) 경우가 앞쪽에 위치하도록 한다. - 삼항연산자, do-while문장은 코드 가독성을 떨어트리니 되도록 사용하지 않는다. - 블록이 너무 많이 중첩되면 코드를 읽기 어려워진다. 블록을 private메소드로 추출할 수 있는지 고민한다. -코드는 빈줄을 이용해 커다란 블록을 논리적인 문단으로 구분한다. - 코드는 들여쓰기를 잘 지키도록 한다. - 필요하지 않은 코드는 제거한다.
JAVA - 프로젝트 생성	- pom.xml 파일에 알맞은 의존성을 설정할 수 있어야 한다. - .gitignore파일이 주어진 요구사항대로 알맞게 작성되어 있어야 한다.
JAVA - 프로젝트 구조	- Controller, Service, Repository를 사용하여 구현되어 있어야 한다. - Controller에서 Service를 Service에서 Repository의 기능을 호출할 수 있지만, 그 반대는 허용되지 않는다.
JAVA - Web API	- 전체, 카테고리별 상품정보를 제공하는 Web Api를 제공한다. 해당 Web API는 카테고리 id를 파라미터로 받을 수 있다. 카테고리 id를 받지 않을 경우엔 전체 상품정보를 제공한다. 해당 Web API는 순번을 파라미터로 받을 수 있다. 파라미터로 순번을 못 받을 경우 첫 번째 데이터부터 4번째 데이터까지 제공한다. 순번을 받으면 순번 이후의 상품정보를 최대 4건을 제공한다. - 전체, 카테고리별 상품 수를 제공하는 Web Api를 제공한다. - 프로모션 정보를 알맞게 제공하는 Web Api를 제공한다. - SQL 문을 표준에 맞추어 작성하며 각 DBMS 별로 구현된 문법 사용은 최소화한다. - 전체, 카테고리별 상품 수는 GROUP BY 구문을 이용한다. - 데이터 수를 조회하는 경우 COUNT(*) 를 사용한다.

## 평가기준표 - FE

### 화면 레이아웃

기준	세부항목
메인 페이지 레이아웃	- HTML/CSS가 제공되는 것을 기본으로 작성해야 하며, 메인페이지의 화면구성은 아래 기획서대로 보여야 한다. <a href="#">기획서 보기</a> - GNB영역, 프로모션영역, 카테고리영역(6개의 메뉴구성). 상품리스트(기본 4개), 더보기 화면이 노출되어야 한다.

## 기능

기준	세부항목
프로모션영역	- 프로모션영역의 이미지는 1개보다 많으며, 자동으로 슬라이딩되어 넘어간다. - 슬라이딩 이미지는 애니메이션이 되면서 좌측으로 이동하는 것이 보여야한다. - 마지막 이미지에 다르면 처음것이 그 다음으로 노출되어야 한다. 마지막것에서 처음내용이 다시 보이는 부분은 끊겨서 노출되도 상관 없으며, 중요한 건 다시 처음부터 슬라이딩이 계속 되어야 한다는 것이다. (엄격하진 않지만 최대한 자연스럽게 동작하려고 해야 한다) - 이동되는 속도는 적절하게 처리되면 된다. - 슬라이딩 이미지는 마우스나 터치에 반응하지는 않는다.
카테고리 노출영역 (탭 UI)	- 전체리스트가 Ajax를 통해서 화면에 4개의 아이템이 노출된다. - 탭별로 전체갯수가 상단에 노출되어야 한다. - 각 아이템(전시상품)은 이미지/제목/장소/설명이 노출되어야 한다. - 탭을 누르면 다른 카테고리 콘텐츠 4개가 다시 노출된다. - 더보기를 누르면 4개씩 노출되어야 한다. 4개보다 적으면 적은 만큼 노출되어야 한다. - 더보기줄 데이터가 없다면 더보기는 사라진다. - TOP영역이 선택되면, 화면 맨 위로 이동한다

## 소스코드

기준	세부항목
HTML,CSS	- 제공되는 HTML/CSS를 수정할 수 있으며, 제공되는 코드의 형태의 구조를 지나치게 수정하지 않도록 한다. - 기존에 제공되는 class와 id규칙에 따라서 새로운 내용을 추가할 수 있다. - TABUI로 구성되는 카테고리 아이템이 노출 되는 영역의 HTML은 카테고리별로 각각 만들지 않고 하나로 만들어 재사용한다.
CSS3 를 사용한 애니메이션 구현	- 상단영역의 애니메이션은 CSS3의 transition과 transform 속성을 JavaScript로 제어하면서 구현해야 한다.
효과적인 브라우저 기반 개발 방식사용	- DOMContentLoaded 이후에 모든 자바스크립트 로직이 동작되게 한다. - 카테고리 탭을 선택할 때마다, Ajax요청을 해서 데이터를 가져와야 한다. - 탭메뉴 (전시/뮤지컬/콘서트 등)는 Event delegation으로 구현한다.
JavaScript 코드 개선하기	- Template 코드를 javascript 함수안에 보관하지 않고, 별도 분리시켜서 사용해야 한다. (HTML에 script태그 안에 보관한다던가) - 함수 하나에 여러개의 기능을 넣지 않고,함수를 여러개로 분리한다.

