



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

기본 제공 페이지

N 예약

예약확인

전체리스트


전시

뮤지컬

콘서트

클래식

바로 예약 가능한 별사가 10개 있습니다




오리지널 소울에
압도당하다!
당신만의 스타를 지금 만나

DREAMGIRLS

2017.04.04-05.25차 두 번째까지

뮤지컬 드림걸즈
(DREAMGIRLS) 최초 내
한
샤롯데 씨어터

뮤지컬 드림걸즈
(DREAMGIRLS) 최초...




ROMEO
AND JULIET

뮤지컬

종로구 두산아트센터 연...

웹메이드 창작 뮤지컬의
대표 브랜드 '김수로 ...



인더하이스

IN THE HEIGHTS

N 예약

예약확인

전체

이용예정

이용완료

취소/환불

6

2

2

2

예약 확정

No.0000000

서비스명/상품명

일정 2000.0.00.(월)2000.0.00.(일)

내역 내역이 없습니다.

장소 내역이 없습니다.

업체 업체명이 없습니다.

결제 예정금액 000,000,000 원

취소

No.0000000

서비스명/상품명

일정 2000.0.00.(월)2000.0.00.(일)

내역 내역이 없습니다.

장소 내역이 없습니다.

업체 업체명이 없습니다.

결제 예정금액 000,000,000 원

취소

N 예약

비회원 예약확인

예약자 이메일 입력

crong@naver.com

내 예약 확인

TOP

로그인

전체서비스

네이버(주)는 통신판매의 당사자가 아니며, 상품외정보, 거래조건, 이용 및 환불 등과 관련한 의무와 책임은 각 회원에게 있습니다.

(주)네이버 사업자정보

이용약관 | 개인정보처리방침 | 네이버 예약 고객센터

© NAVER Corp.

N 예약

예약확인

1 / 2

펼쳐보기

이벤트 정보

[네이버예약 특별할인]

R석 50%, S석 60% 할인

예약하기

예약자 한줄평

★★★★☆ 4.2 / 5.0

52건 등록

2층이어서 걱정했는데 꽤잘보여서 좋

KLIMT

2017.2.17.(금)~2017.4.18.(화)

잔여티켓 2769매

₩12,000

장소 :

기간 : 2017.2.17.(금)~2017.4.18.(화)

관람시간

화, 목, 금 일요일 10:00am~06:00pm(입장마감 05:30pm)

'문화가 있는 날' 매월 마지막 주 수요일은 오후 8시까지 연장

요금

성인(만 19~64세) 5,000원 / 청소년(만 13~18세) 4,000원

어린이(만 4~12세) 3,000원 / 20인 이상 단체 20% 할인

국가유공자, 장애인, 65세 이상 4,000원

성인 10,200 원

10,200원 (15% 할인가)

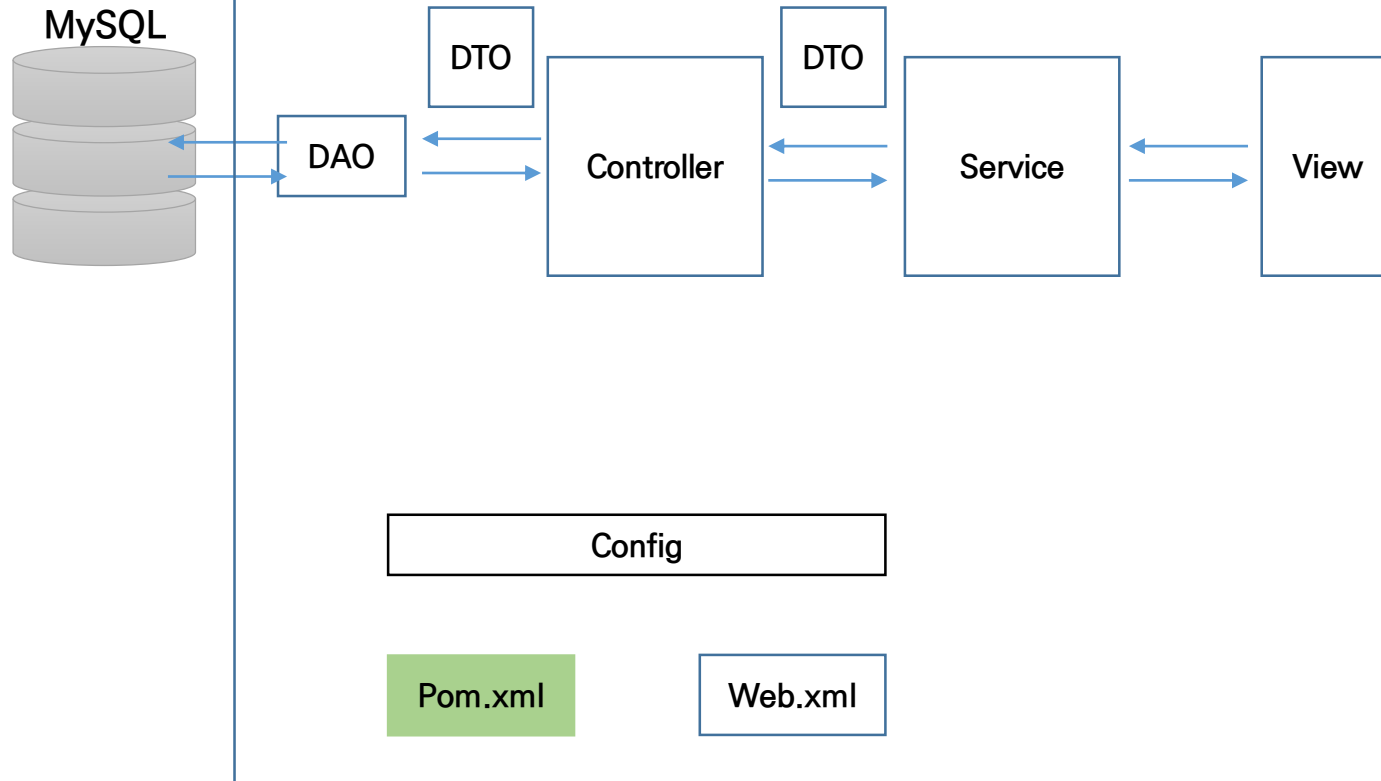
유아 5,800 원

시스템 설정 (pom.xml)

Pom.xml

- Spring
- DB
- Servlet
- JSON
 - 아직 확실히 필요한지 알 수 없음
- dbcp

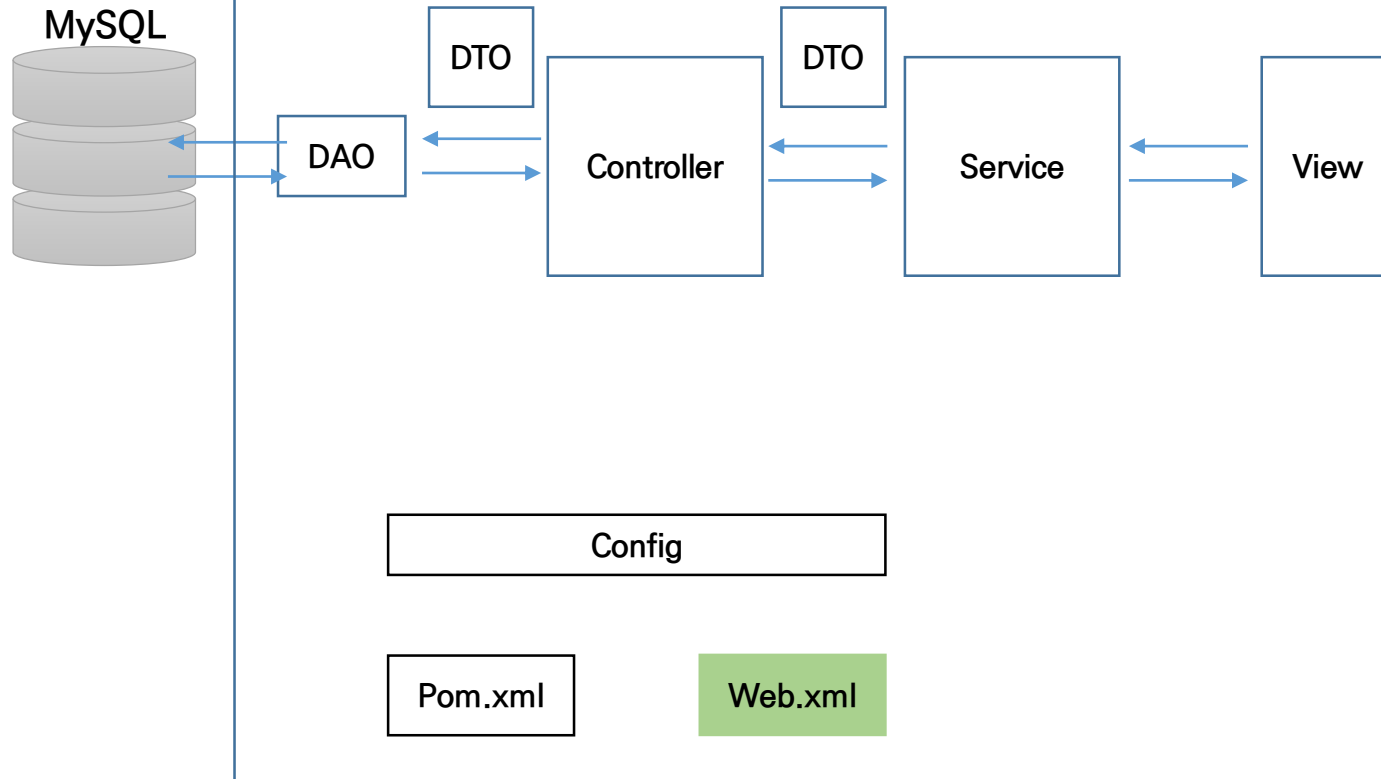
Spring Framework



Web.xml

- context-param
 - contextConfigLocation
 - AppConfig (자바 코드로 설정)
 - ContextClass
 - AnnotationConfigWebApplicationContext (어노테이션 설정 사용)
- listener
 - ContextLoaderListener (어떤 특정 이벤트가 발생 했을 때 동작)
- Servlet
 - DispatcherServlet (Controller의 주요 역할)
 - Init-param
 - ContextClass
 - AnnotationConfigWebApplicationContext (어노테이션 설정 사용으로 dispatcherServlet에 알림)
 - ContextConfigLocation
 - WebMvcContextConfiguration (controller에 resource 자원 어떻게 읽을 지 알리는 설정)
- EncodingFilter
 - CharacterEncodingFilter

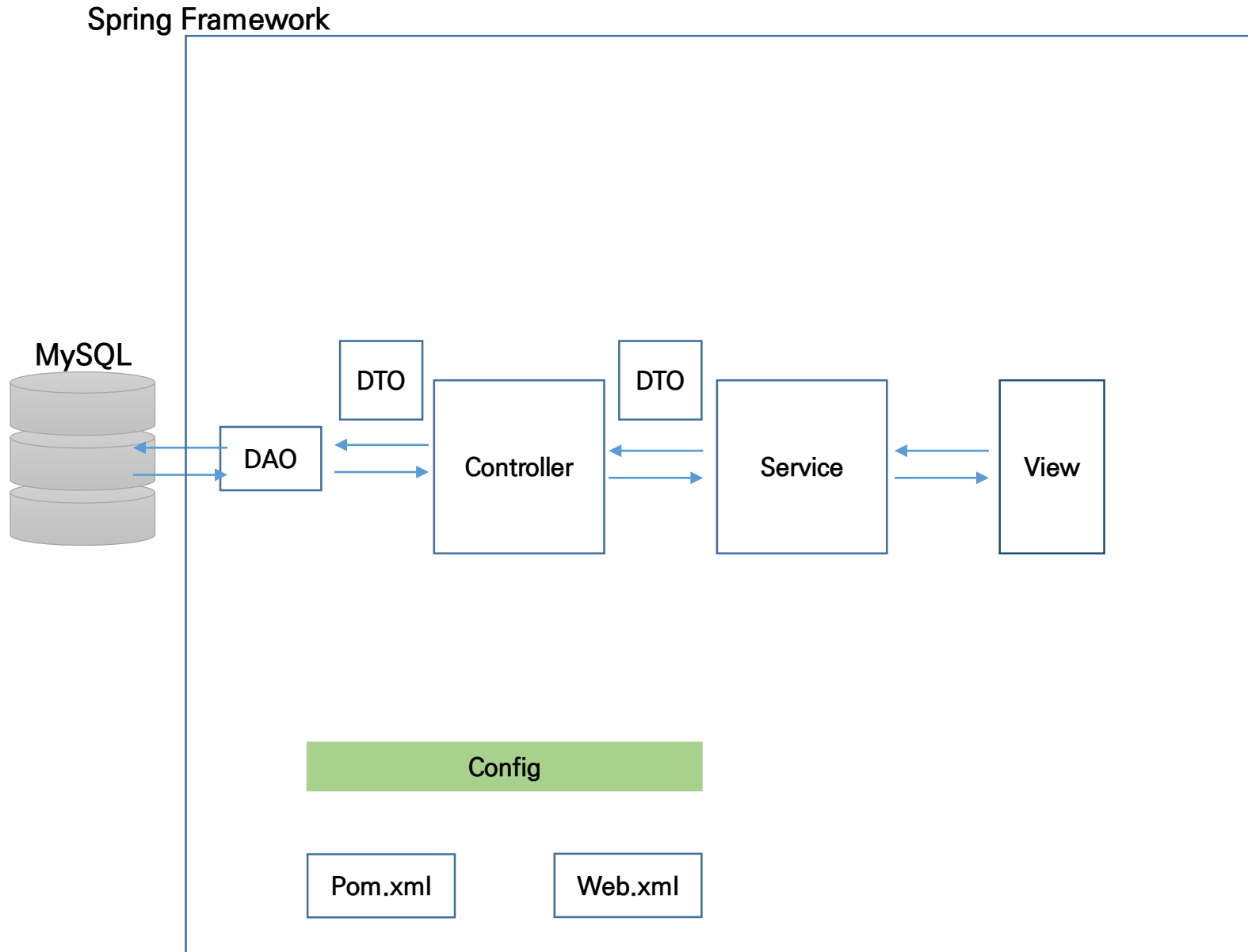
Spring Framework



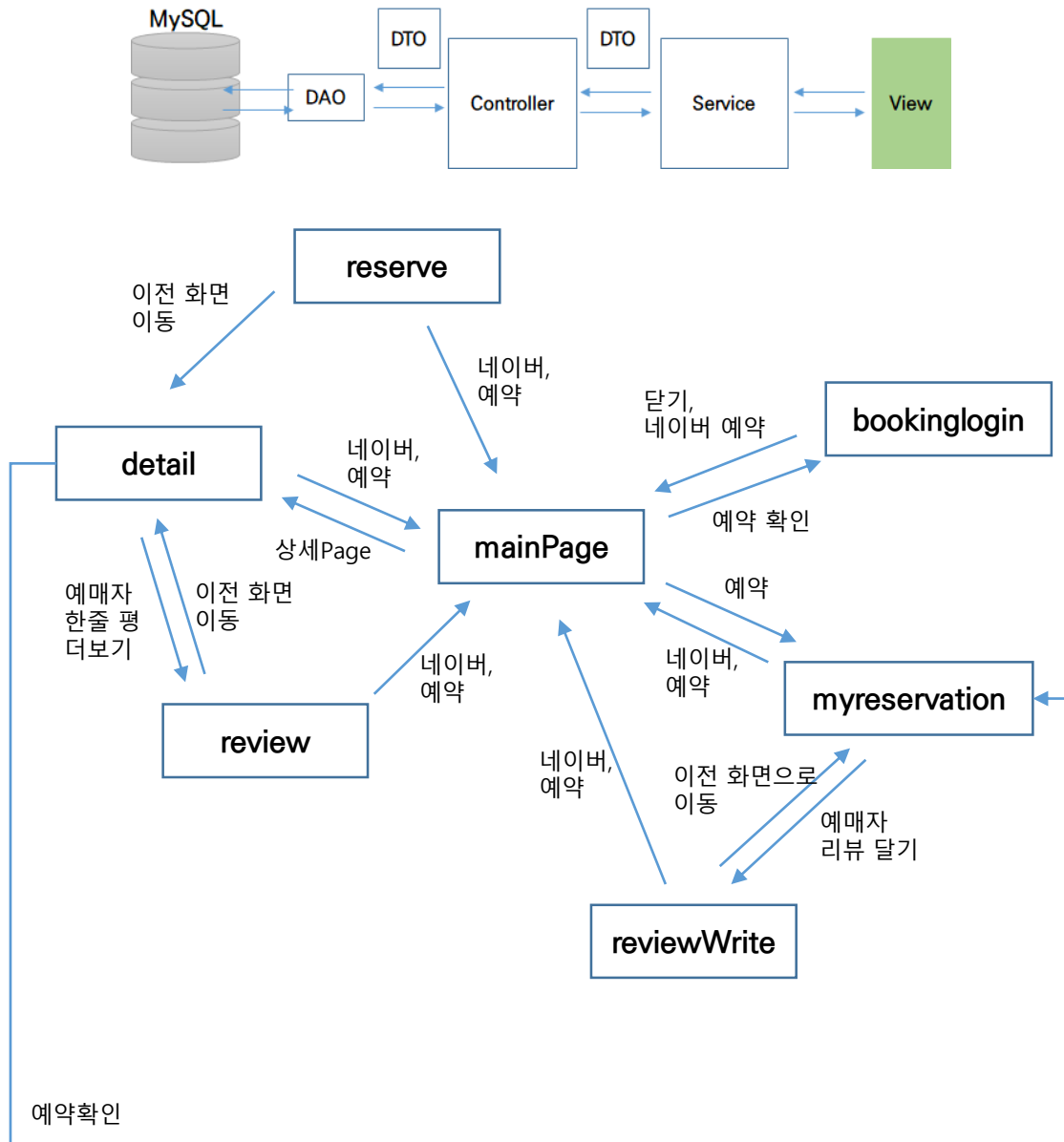
시스템 설계 (config)

Config

- DBConfig 설정
 - DataSource
 - @PlatformTransactionManager
 - @PropertySource
 - Application.properties
- AppConfig 설정
 - @ComponentScan (Dao, Service)
 - Import – DBConfig
- WebMvcContextConfiguration 설정
 - addResourceHandler()
(resource 자원 경로 설정)
 - addViewControllers()
(특정 경로에 대한 View처리)
 - configureDefaultServletHandling()
(?)
 - getInternalResourceViewResolver()
(View Resolver 설정)



View Flow



View

• JSP 페이지 이동에 관한 문서

- mainPage

- 1) 예약확인 (bookinglogin)
- 2) 예약 (myreservation)
- 3) 상세페이지 (detail)

- bookinglogin

- 1) 네이버 예약, 달기 (mainPage)

- myreservation

- 1) 네이버, 예약 (mainPage)
- 2) 예매자 리뷰 달기 (reviewWrite)

- reserve

- 1) 네이버, 예약 (mainPage)
- 2) 이전 화면 이동 (detail)

- detail

- 1) 네이버, 예약 (mainPage)
- 2) 예매자 한줄 평 더보기 (review)
- 3) 예약확인 (myreservation)

- review

- 1) 네이버, 예약 (mainPage)
- 2) 이전화면 (detail)

- reviewWrite

- 1) 네이버, 예약 (mainPage)
- 2) 이전화면 (myreservation)

MainPage (Promotion)

Promotion

JavaScript

1. 프로모션 Data를 Ajax통신으로 가져오기
2. 가져온 데이터를 HTML Templating

CSS & JS

1. 프로모션 이미지를 Transition, Transform을 이용, 어떤 방식으로 움직일 지 정의
2. JS로 동적인 부분처리

Category Tab UI

JavaScript

1. 카테고리 Data를 Ajax통신으로 가져오기
2. 가져온 Data를 카테고리 템플릿과 HTML Templating

Product Item

JavaScript

1. 기본적으로 페이지 로딩시 /api/products로 API 호출
 - 전체 데이터를 출력
2. 카테고리 클릭 시 카테고리에 해당하는 Product 출력
 - id, index 데이터를 포함하여 호출
 - product 개수, product 데이터 출력

Category Event

JS

1. 일일이 li에 이벤트 등록하지 않고 ul에 등록하기
2. li 이상 범위 클릭시 이벤트 X
3. ClassList에서 active제거
4. Click된 부분만 active 추가

View

- 메인페이지 Loading 시 필요한 데이터 출력

✓ Promotion

✓ Tab UI (Category)

✓ 특정 Category의 item을 4개씩 출력

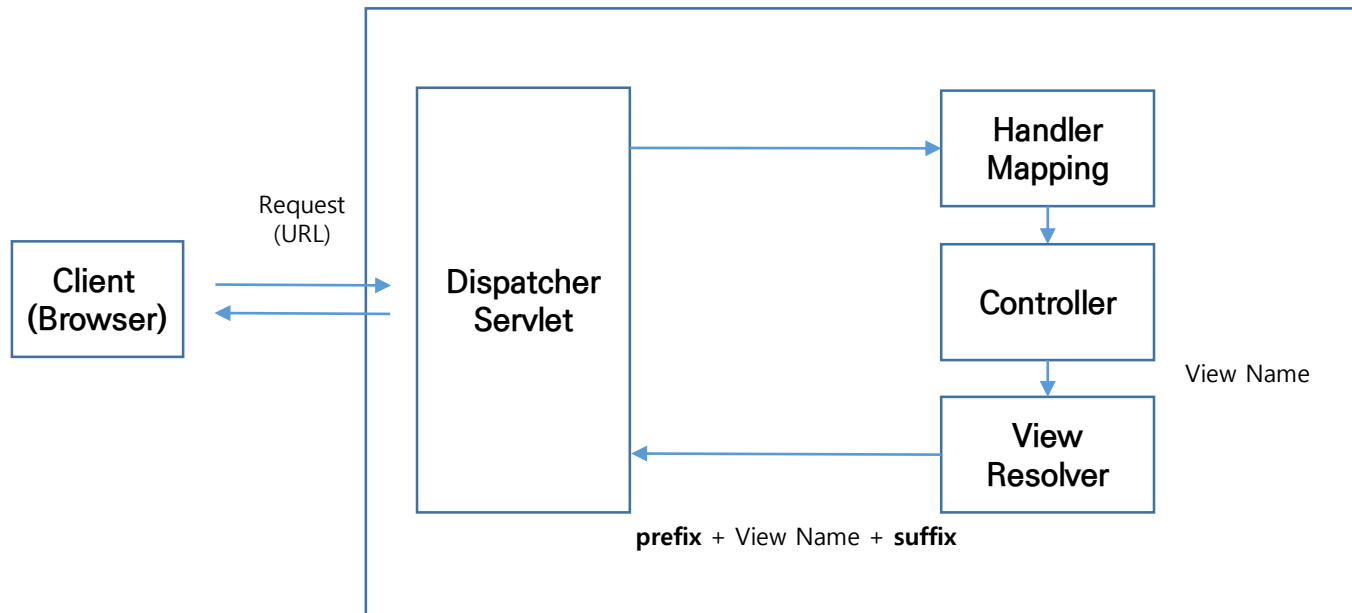
@Controller vs @Controller + @ResponseBody vs @RestController

Spring MVC

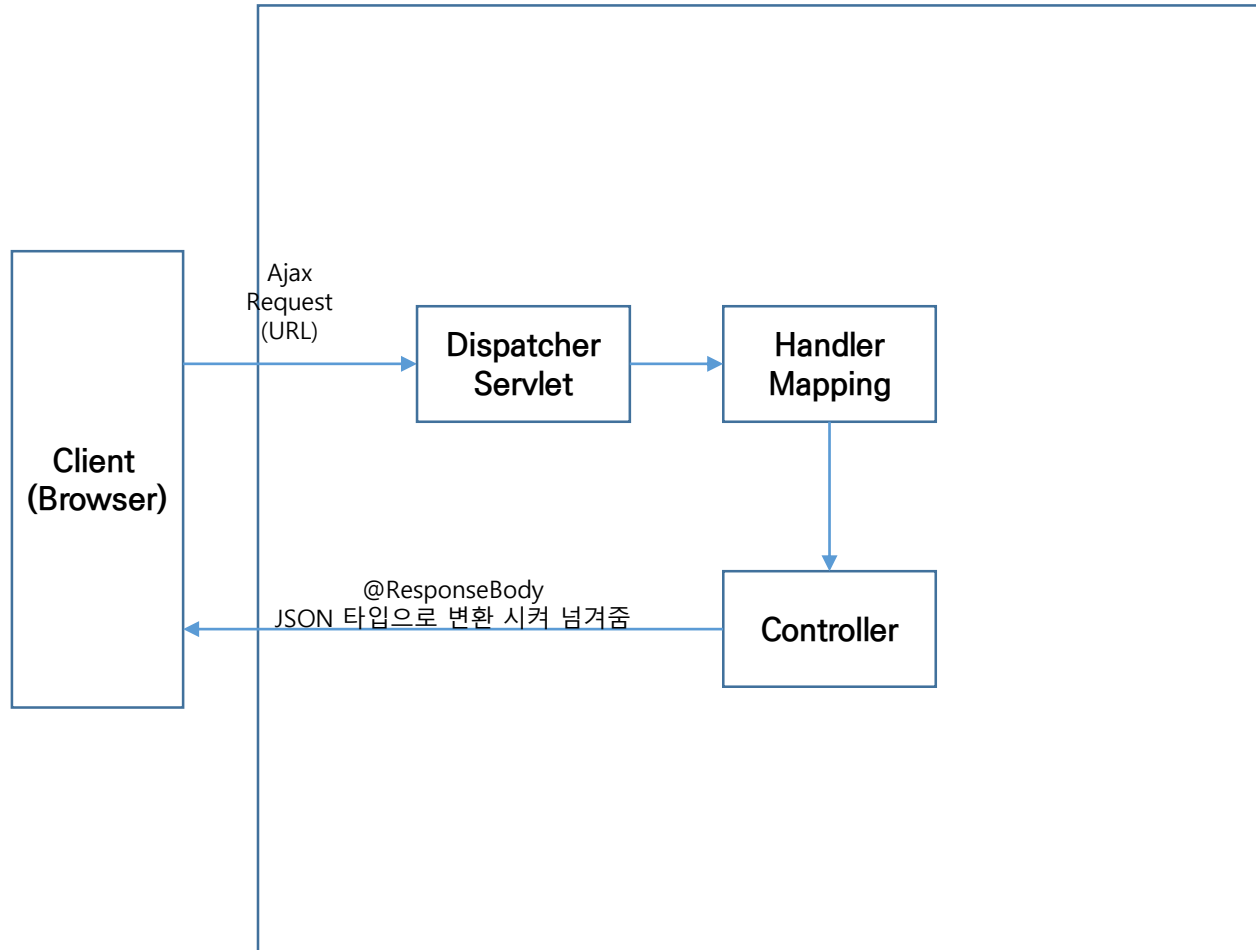
- @Controller
- @Controller + @ResponseBody
- @RestController

✓ @Controller

- ① Client의 요청이 발생
- ② Dispatcher Servlet은 Handler Mapping에게 클라이언트가 요청한 작업을 수행할 수 있는 컨트롤러가 무엇이 있는지 정보를 받는다.
- ③ Dispatcher Servlet가 2번의 정보를 받은 후에 Handler Adapter에게 작업 수행이 가능한 Controller에게 실행을 요청
- ④ 그 결과를 Model에 받아 Dispatcher Servlet에게 전달
- ⑤ 컨트롤러가 리턴한 View name은 View Resolver가 적절한 View를 찾아 실행



@Controller vs @Controller + @ResponseBody vs @RestController



Spring MVC

- @Controller
- @Controller + @ResponseBody
- @RestController

✓ @Controller + @ResponseBody

- Restful 방식

- ① Client의 요청이 발생
- ② Dispatcher Servlet은 Handler Mapping에게 클라이언트가 요청한 작업을 수행할 수 있는 컨트롤러가 무엇이 있는지 정보를 받는다.
- ③ Dispatcher Servlet가 2번의 정보를 받은 후에 Handler Adapter에게 작업 수행이 가능한 Controller에게 실행을 요청
- ④ Controller에서 작업 처리 후 리턴 (이때, 컨트롤러 클래스의 메서드 리턴 타입에 @ResponseBody를 포함)
- ⑤ Dispatcher Servlet으로 리턴 된 후 Dispatcher Servlet은 View로 전달
- ⑥ @ResponseBody 어노테이션을 사용하여 View를 리턴하지 않고 컨트롤러에서 직접 데이터를 리턴 (Spring은 Http 응답에 리턴 값을 자동 변환)

@Controller vs @Controller + @ResponseBody vs @RestController

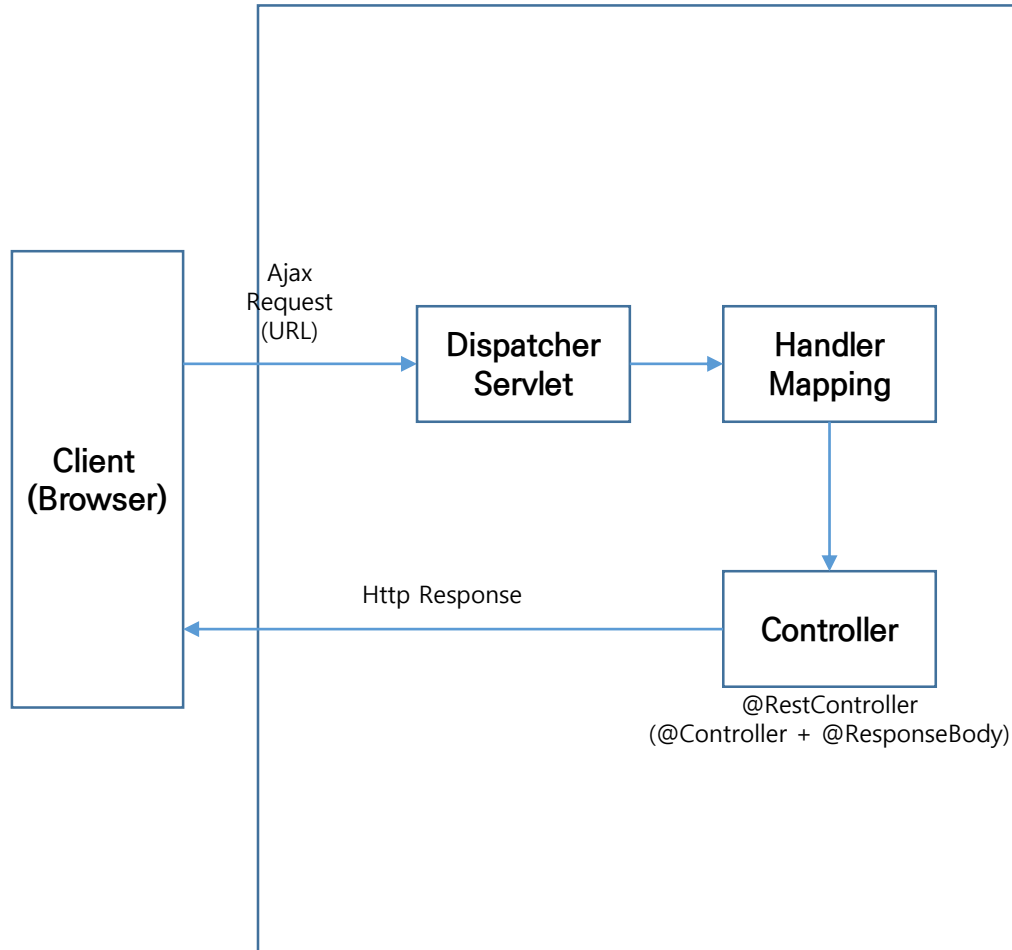
Spring MVC

- @Controller
- @Controller + @ResponseBody
- @RestController

✓ @RestController

- Spring 4.x MVC Restful Web Service

- ① Client의 요청이 발생
- ② Dispatcher Servlet은 Handler Mapping에게 클라이언트가 요청한 작업을 수행할 수 있는 컨트롤러가 무엇이 있는지 정보를 받는다.
- ③ Dispatcher Servlet가 2번의 정보를 받은 후에 Handler Adapter에게 작업 수행이 가능한 Controller에게 실행을 요청
- ④ Controller에서 작업 처리 후 리턴 (@RestController가 적용된 클래스는 메서드에 @ResponseBody가 기본으로 적용)
- ⑤ 데이터를 넘길 때 String으로 전달 할 지 Object로 전달할 지는 선택



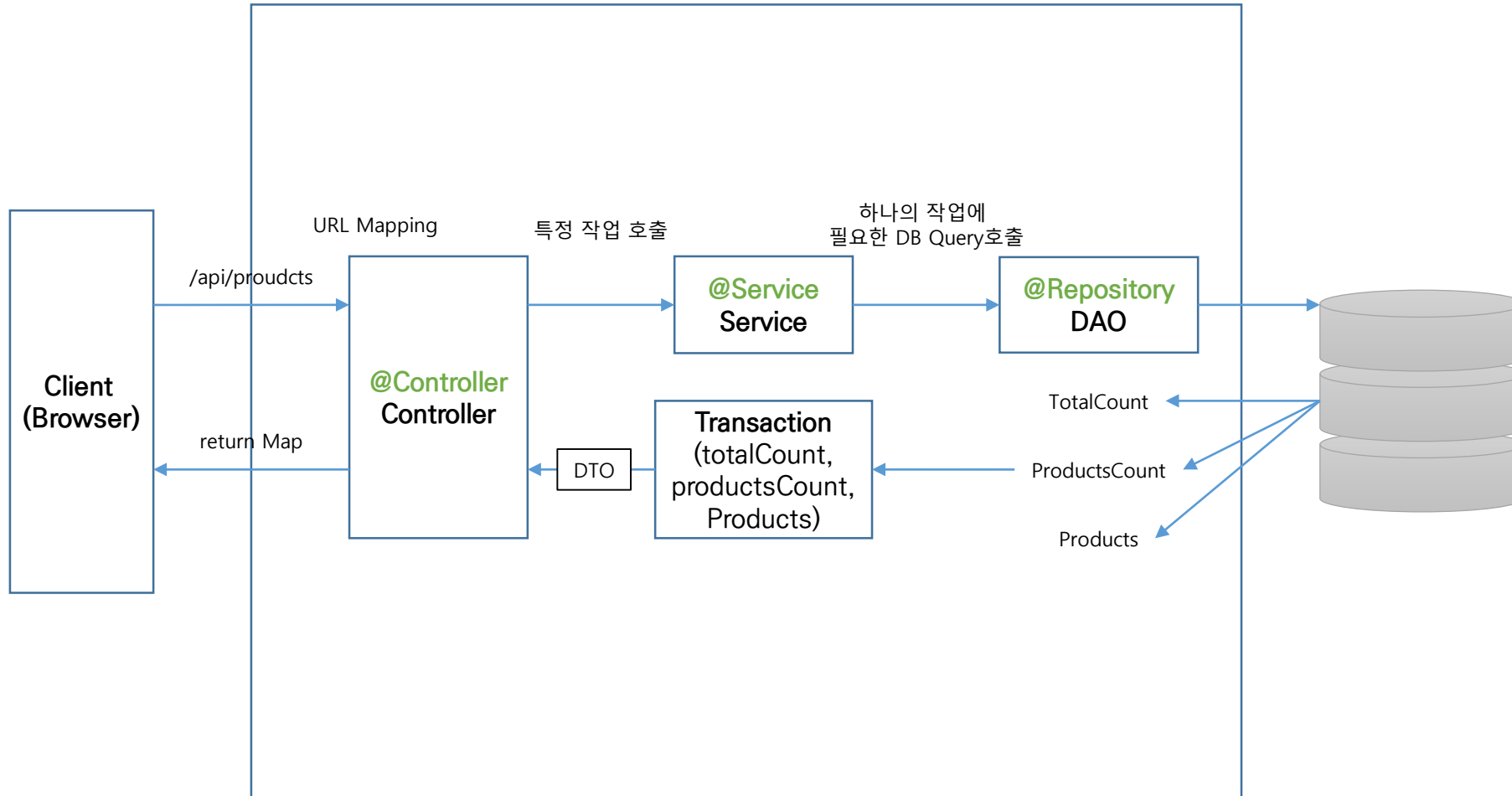
API Controller

Spring MVC

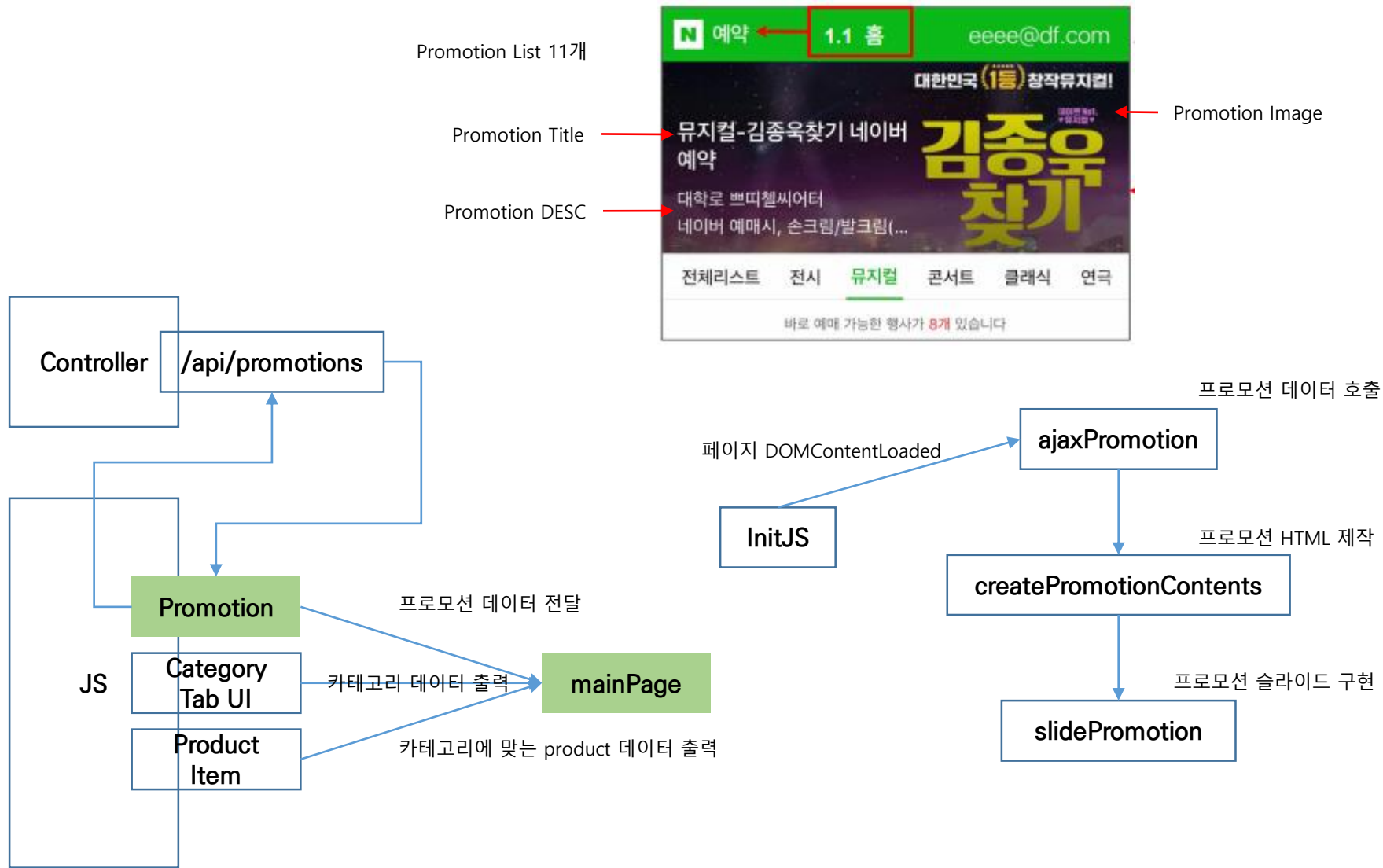
- API 만들기

- ✓ API 호출 Flow

- ① 사용자의 URL 요청
- ② @RestController 에서 @Service를 호출
- ③ @Service에서 필요한 DAO (@Repository) 메서드들을 호출
- ④ DAO에서 호출한 Query의 결과 값을 받아 핸들링할 타입으로 저장
- ⑤ Service에서 작업단위에 해당하는 결과 값들을 취합
- ⑥ Controller에서 Service 메서드를 호출한 값을 저장, View에 전달하기 위해 Map<String, Object> 형식으로 저장
- ⑦ View에서 Key 값으로 필요한 Object를 핸들링



MainPage (Promotion)



View

- 메인페이지 Loading 시 필요한 데이터 출력

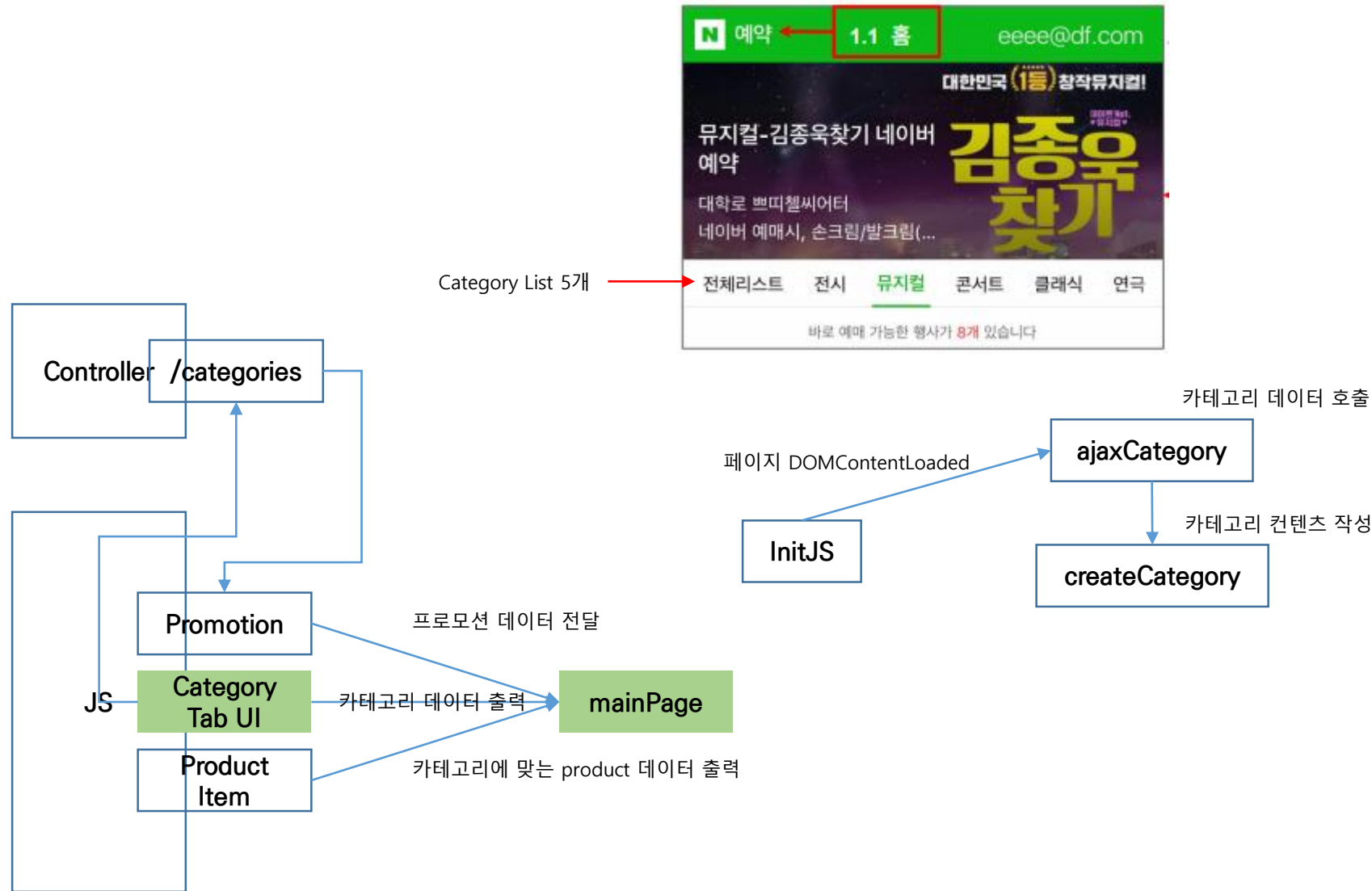
✓ Promotion

- MainPage에 DOMContentLoaded 시 Ajax 통신으로 DB의 데이터를 가져와 배열에 저장
- 배열에 저장된 HTML Template에 필요한 파라미터를 채우기
- 페이지 출력
- CSS3 Transition, Transform 속성을 활용해서 구현

✓ JS Flow

- 페이지 DOMContentLoaded 시 JS 시작
- Ajax로 데이터 호출
- 호출된 데이터로 Promotion HTML 작성
- 프로모션 Slide

MainPage (Category)



View

- 메인페이지 Loading 시 필요한 데이터 출력

✓ Promotion

✓ Tab UI (Category)

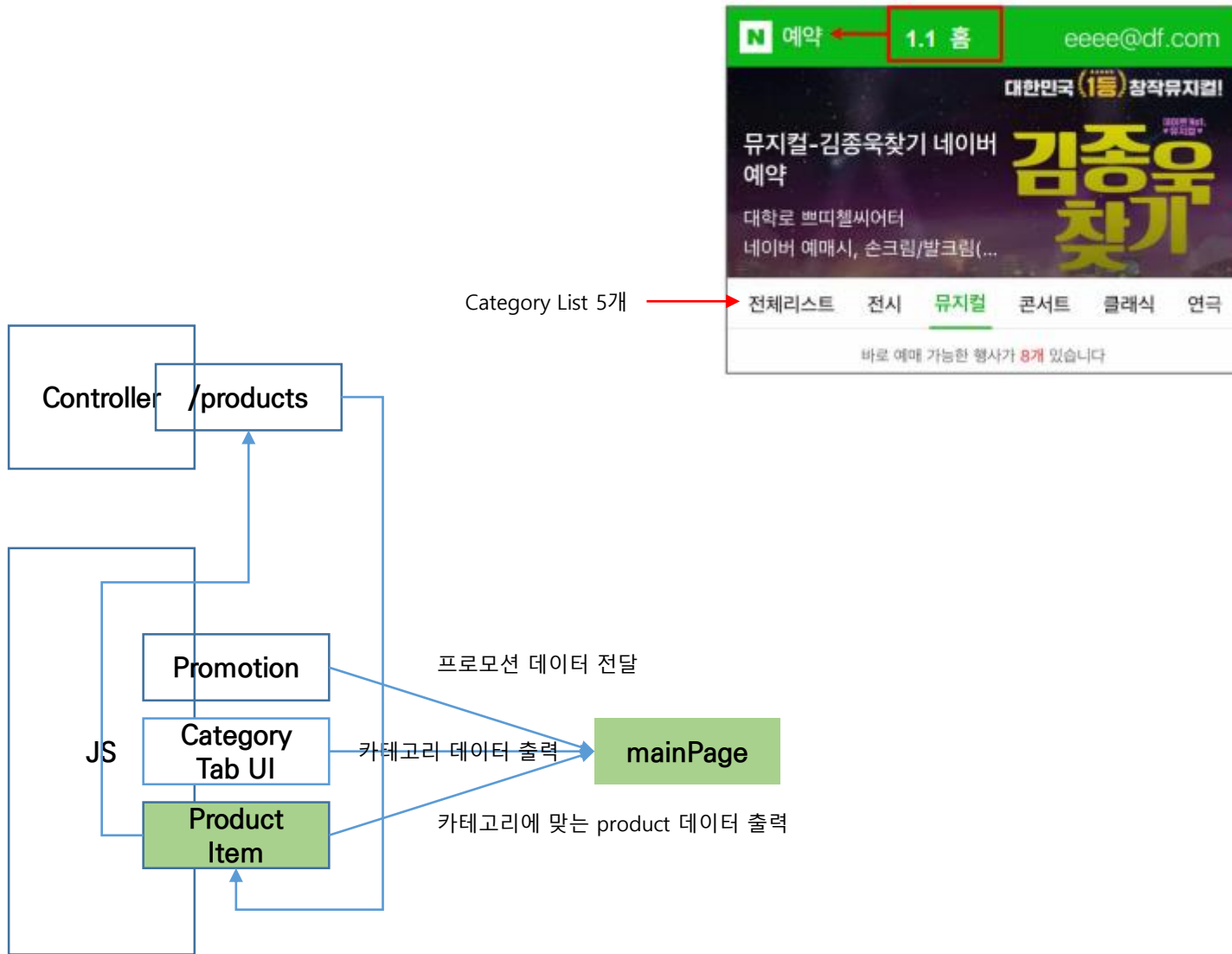
1. Controller에서 ModelMap으로 넘겨준 데이터를 JSTL로 출력하기
2. Ajax로 Controller에 "api/categories" URL을 호출하여 DB에 접근, DB에서 가져온 데이터를 JSON type으로 변환하여 HTML Templating
 - ① 페이지 로드 (DOMContentLoaded)
 - ② Ajax로 "api/categories"에 API 호출
 - ③ JSON 데이터 타입으로 변환하여 DB category 데이터를 반환
 - ④ HTML Template 방식으로 출력하여 하드코딩 하지 않도록 함
3. 211.249.62.123/api/categories 에서 반환되는 JSON 타입 데이터를 가져와 처리하기

✓ 특정 Category의 item을 4개씩 출력

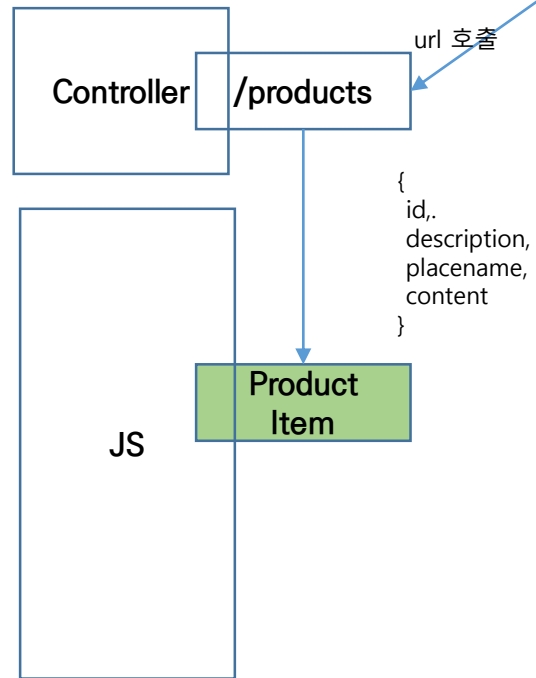
MainPage (Product)

View

- 메인페이지 Loading 시 필요한 데이터 출력
- ✓ Promotion
- ✓ Tab UI (Category)
- ✓ 특정 Category의 item을 4개씩 출력



MainPage (Product)



필요한 정보

: 카테고리 구분 키워드

출력될 정보

: 카테고리에 해당하는 아이템 개수 및 아이템 내용

1. 클릭 이벤트로 target의 category 명을 받아옴
2. Ajax로 해당 카테고리의 아이템 개수, 아이템을 출력
 - 2-1 아이템 목록을 출력 시 전체 아이템을 가져와 "더보기" 클릭 시 미리 가져왔던 데이터를 출력
 - 2-2 아이템 목록 출력 시 4개씩 출력 "더보기"클릭 시 다음 4개 아이템을 출력

View

- 메인페이지 Loading 시 필요한 데이터 출력

✓ Promotion

✓ Tab UI (Category)

1. Category UI에 Event delegation 방식으로 등록

✓ 특정 Category의 item을 4개씩 출력

E-R Diagram

Database Table

