신한DS 금융 SW 아카데미 MVC2 프로젝트

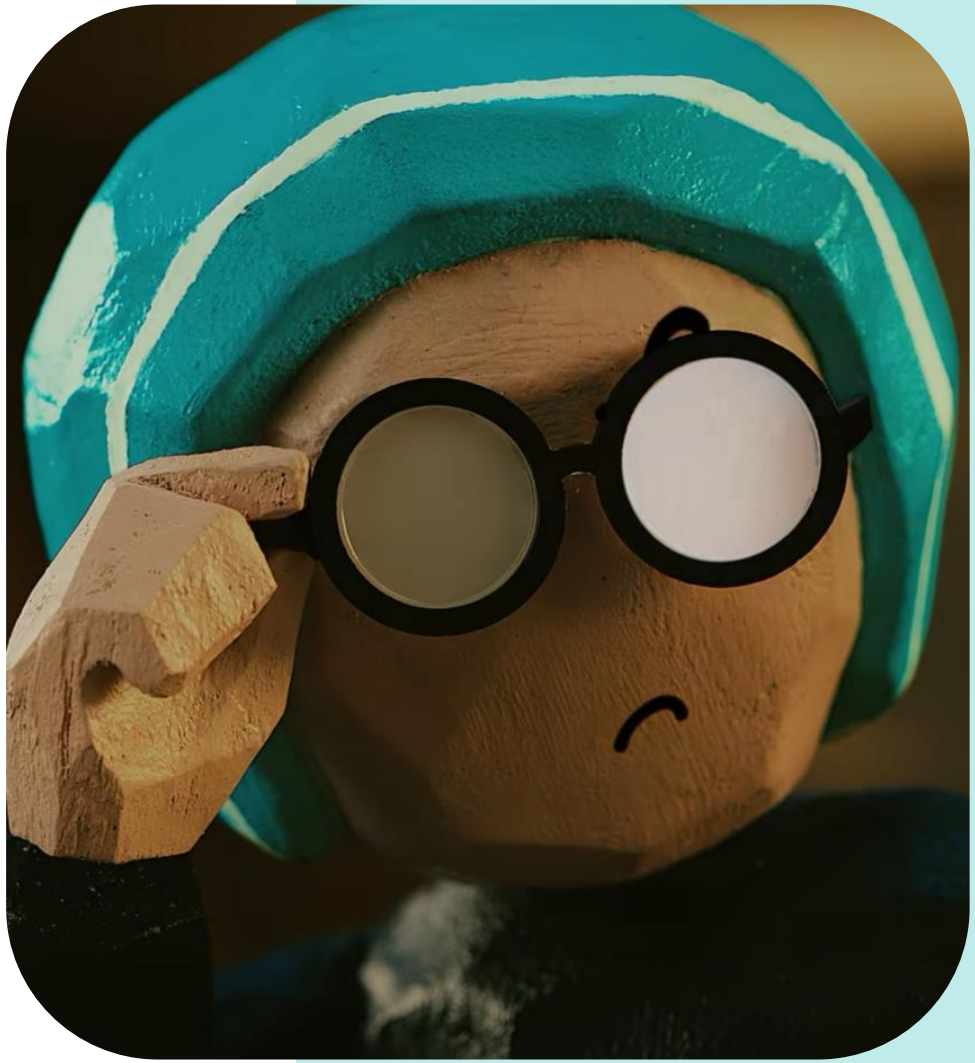
MVC2 모델 배달의민족 프로그램

송재명

CONTENTS

개 발 개 요

MVC2 모델을 이용한
요청과 응답, 비즈니스 로직을
구분한 시스템개발



MVC2 모델을 적용한 어플리케이션 개발

기존 개발한 **Model**에 **Controller, View을 결합**하여
MVC2모델의 시스템 구조를 설계합니다.

Servlet과 **View**를 통해 **요청과 응답을 구분**하고
비동기 작업 수행이 가능한 어플리케이션을 개발합니다.

개발 환경

View

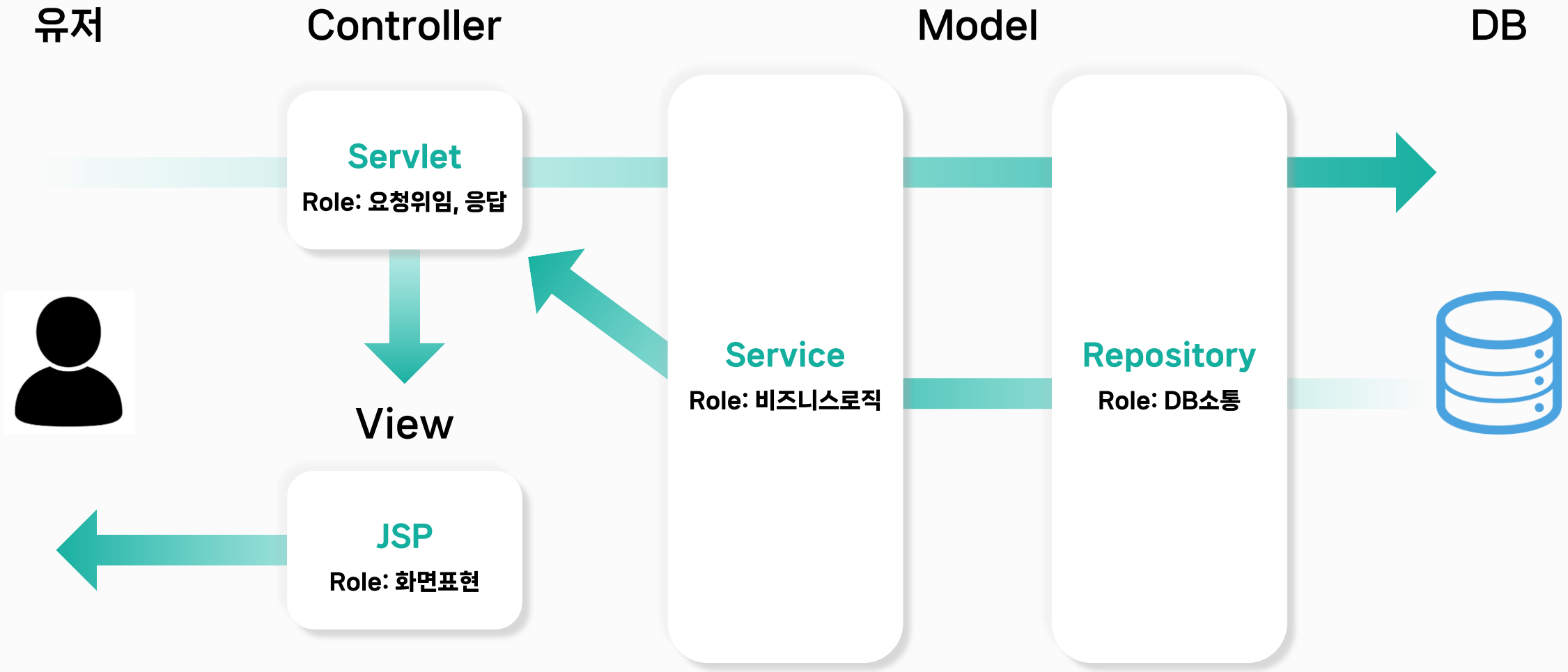
Controller, Model

데이터베이스

시스템 개요

MVC2 패턴을 이용한
시스템 구성

MVC2 패턴을 이용한 구조 설계



개 선 사 항

명확한 역할구분과
비동기 코드의 캡슐화와 정보은닉

MVC2 패턴, View 모델 구축, 비동기 통신

MVC2 패턴 동작 방식 이해

HTML, CSS, JavaScript를 이용한 View 모델 구축

Ajax를 이용한 비동기 통신의 개념 이해

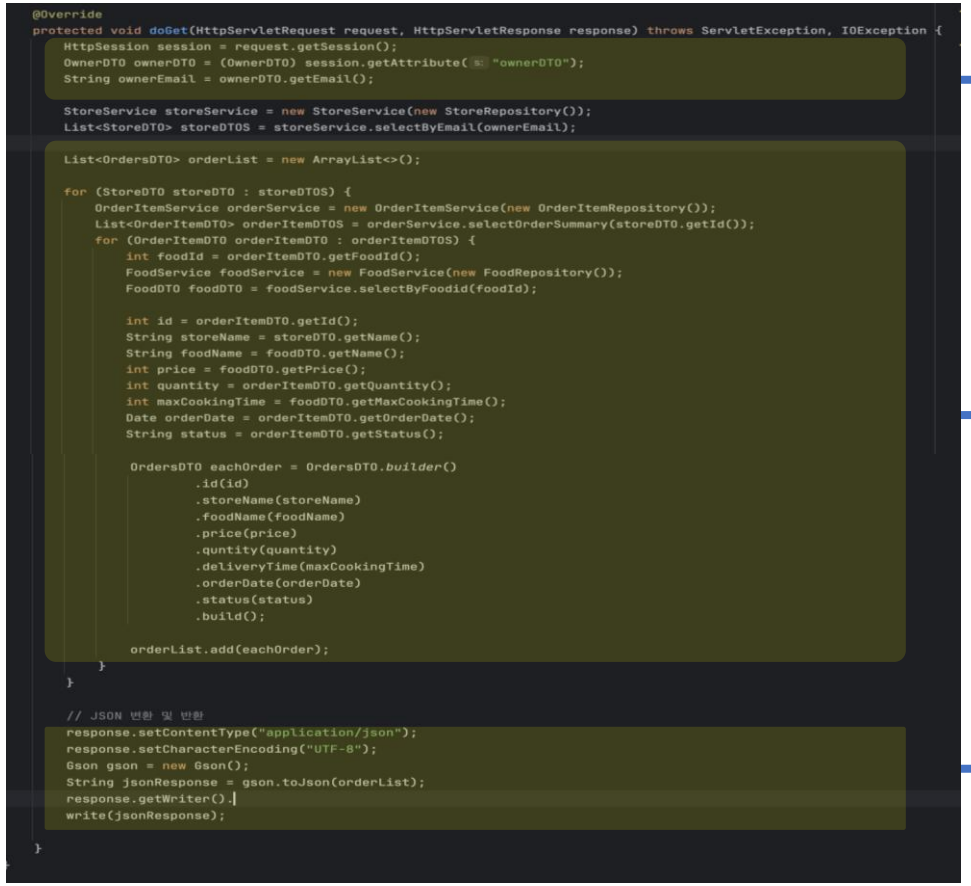
소프트웨어 아키텍쳐 부족과 클린 코드

명확하지 않은 역할 구분 SIP(Single Responsibility Principle)

비동기 코드의 캡슐화, 정보은닉

나열식 코드 작성으로 인해 유지보수의 어려움

주문서 업데이트 서블릿



```java
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    HttpSession session = request.getSession();
    OwnerDTO ownerDTO = (OwnerDTO) session.getAttribute("ownerDTO");
    String ownerEmail = ownerDTO.getEmail();

    StoreService storeService = new StoreService(new StoreRepository());
    List<StoreDTO> storeDTOS = storeService.selectByEmail(ownerEmail);

    List<OrdersDTO> orderList = new ArrayList<>();

    for (StoreDTO storeDTO : storeDTOS) {
        OrderItemService orderService = new OrderItemService(new OrderItemRepository());
        List<OrderItemDTO> orderItemDTOS = orderService.selectOrderSummary(storeDTO.getId());
        for (OrderItemDTO orderItemDTO : orderItemDTOS) {
            int foodId = orderItemDTO.getFoodId();
            FoodService foodService = new FoodService(new FoodRepository());
            FoodDTO foodDTO = foodService.selectByFoodid(foodId);

            int id = orderItemDTO.getId();
            String storeName = storeDTO.getName();
            String foodName = foodDTO.getName();
            int price = foodDTO.getPrice();
            int quantity = orderItemDTO.getQuantity();
            int maxCookingTime = foodDTO.getMaxCookingTime();
            Date orderDate = orderItemDTO.getOrderDate();
            String status = orderItemDTO.getStatus();

            OrdersDTO eachOrder = OrdersDTO.builder()
                    .id(id)
                    .storeName(storeName)
                    .foodName(foodName)
                    .price(price)
                    .quntity(quantity)
                    .deliveryTime(maxCookingTime)
                    .orderDate(orderDate)
                    .status(status)
                    .build();

            orderList.add(eachOrder);
        }
    }

    // JSON 변환 및 반환
    response.setContentType("application/json");
    response.setCharacterEncoding("UTF-8");
    Gson gson = new Gson();
    String jsonResponse = gson.toJson(orderList);
    response.getWriter().
    write(jsonResponse);
}
```

Parameter 받기

Request에 담을
Attribute를 만드는 로직

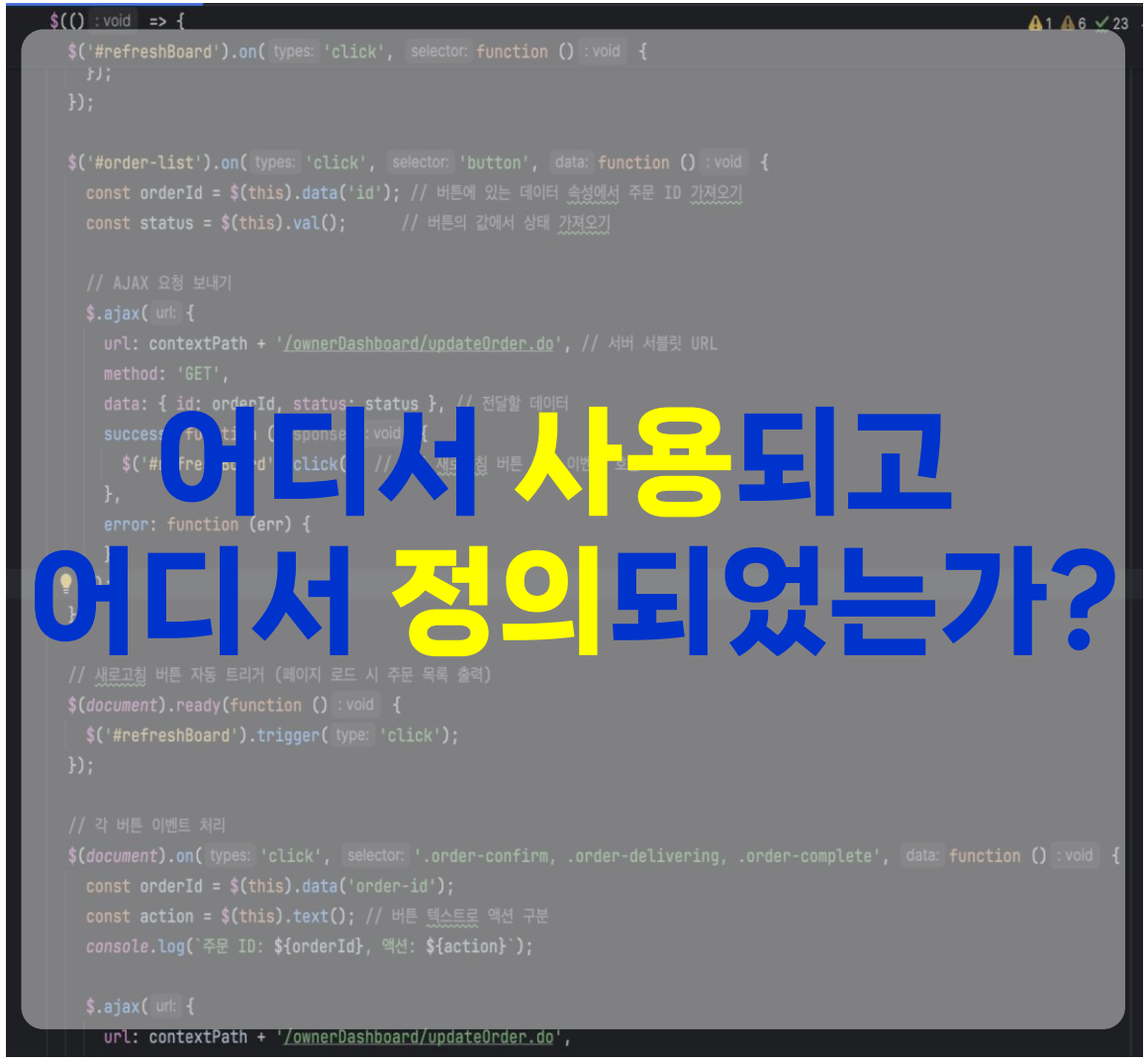
요청 처리와 데이터 처리 모두 수행

Service 계층에서 처리했다면...

JSON으로 데이터 변환

역할이 구분되지 않아 유지보수 시 어디서 코드를 찾아야하는지 혼동

판매자 메인보드 자바스크립트: 나열식 코드



역할과 책임을 분리

```
/src
  /components
    - Header.js
    - Footer.js
  /services
    - api.js
    - auth.js
  /styles
    - main.css
  index.js
```

감사합니다

THE END

Thank You
for reading