

19장. 모델2 방식 개발 패턴

1. 웹 애플리케이션 모델

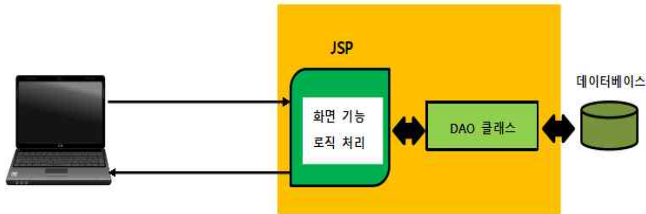
- 애플리케이션 개발시 일반적으로 많이 사용하는
표준화된 소스 구조
- 모델의 종류에는 모델1과 모델2가 있음

1. 웹 애플리케이션 모델

모델1 방식

- 데이터베이스 연동 같은 비즈니스 로직 작업과 그 작업 결과를 나타내주는 작업을 동일한 JSP에서 수행함
- 모든 클라이언트의 요청과 비즈니스 로직 처리를 JSP가 담당하는 구조
- 기능 구현이 쉽고 편리하지만 유지보수가 어려움

1. 웹 애플리케이션 모델



모델1로 구현한 애플리케이션 동작 방식

1. 웹 애플리케이션 모델

모델2 방식

- 웹 애플리케이션의 각 기능(클라이언트의 요청 처리, 응답 처리, 비즈니스 로직 처리)을 분리해서 구현
- 객체 지향 프로그래밍에서 각각의 기능을 모듈화해 개발하자는 원리

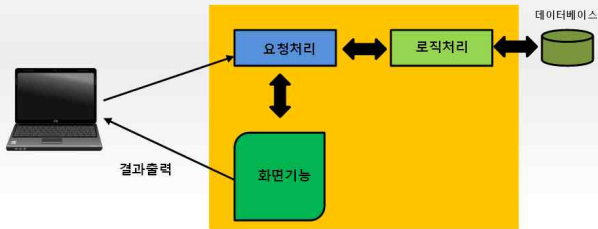
1. 웹 애플리케이션 모델

모델2 방식 특징

- 각 기능이 서로 분리되어 있어 개발 및 유지보수가 쉬움
- 각 기능(모듈)의 재사용성이 높음
- 디자이너와 개발자의 작업을 분업화해서 쉽게 개발할 수 있음
- 모델2 방식과 관련된 기능이나 개념의 학습이 필요

1. 웹 애플리케이션 모델

모델2 동작 방식



2. MVC 디자인 패턴

MVC란?

- Model-View-Controller(모델-뷰-컨트롤러)의 약자로, 일반 PC 프로그램 개발에 사용되는 디자인 패턴을 웹 애플리케이션에 도입한 것
- 즉, 웹 애플리케이션을 화면 부분, 요청 처리 부분, 로직 처리 부분으로 나누어 개발하는 방법

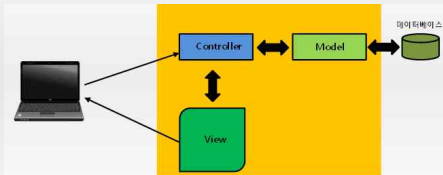
2. MVC 디자인 패턴

MVC 특징

- 각 기능이 분리되어 있어 개발 및 유지보수가 편리함
- 각 기능의 재사용성이 높아짐
- 디자이너와 개발자의 작업을 분업화해서 쉽게 개발할 수 있음

2. MVC 디자인 패턴

MVC 구조와 구성 요소



Controller: 사용자의 요청 및 흐름 제어를 담당

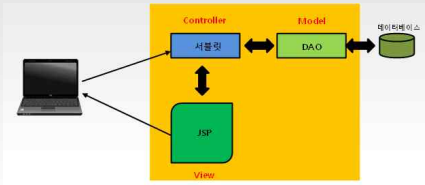
Model: 비즈니스 로직 처리

View: 사용자에게 보여줄 화면 담당

2. MVC 디자인 패턴

MVC 구성 요소와 기능

MVC로 구현된 웹 애플리케이션과 각 구성 요소의 기능



2. MVC 디자인 패턴

Controller

- 서블릿이 컨트롤러의 역할
- 클라이언트의 요청을 분석
- 요청에 대해서 필요한 모델을 호출
- Model에서 처리한 결과를 보여주기 위해 JSP를 선택

2. MVC 디자인 패턴

Model

- 데이터베이스 연동과 같은 비즈니스 로직을 수행
- 일반적으로 **DAO**와 **VO** 클래스로 이루어짐

View

- **JSP**가 화면 기능을 담당
- **Model**에서 처리한 결과를 화면에 표시

3. 모델2로 답변형 게시판 구현하기

답글 기능 게시판

3. 모델2로 답변형 게시판 구현하기

답변형 게시판 테이블(t_board) 구조

no	컬럼이름	속성	자료형	크기	유일키 여부	NULL 여부	키	기본값
1	articleNO	글 번호	number	10	Y	N	기본키	
2	parentNO	부모 글 번호	number	10	N	N		0
3	title	글 제목	varchar2	100	N	N		
4	content	글 내용	varchar2	4000	N	N		
5	imageFileName	이미지 파일 이름	varchar2	100	N	N		
6	writeDate	작성일	date		N	N		sysdate
7	id	작성자 ID	varchar2	20	N	N	외래키	

3. 모델2로 답변형 게시판 구현하기

테이블의 주요 속성

속성	컬럼	설명
글 번호	articleNO	글이 추가될 때마다 1씩 증가되면서 고유 값을 부여합니다.
부모 글 번호	parentNO	답변을 단 부모 글의 번호를 나타냅니다. 부모 글 번호가 0이면 자신이 부모 글입니다.
첨부 파일 이름	imageFileName	글 작성 시 첨부한 이미지 파일 이름입니다
작성자 ID	id	글을 작성한 작성자의 ID입니다.

3. 모델2로 답변형 게시판 구현하기

게시판 테이블 관계

T_MEMBER	
ID	varchar2(10)
PWD	varchar2(10)
NAME	varchar2(50)
EMAIL	varchar2(50)
JOINDATE	date

ID-10

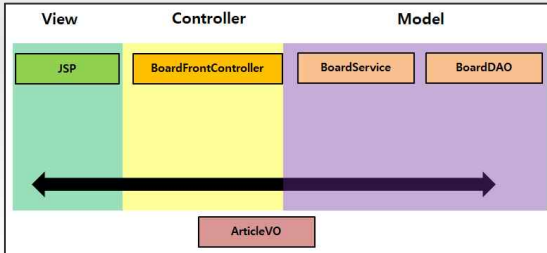
T_BOARD	
ARTICLENO	number(10)
PARENTNO	number(10)
TITLE	varchar2(500)
CONTENT	varchar2(4000)
IMAGEFILENAME	varchar2(30)
WRITEDATE	date
ID	varchar2(10)

3. 모델2로 답변형 게시판 구현하기

테이블 생성 및 데이터 입력하기

3. 모델2로 답변형 게시판 구현하기

- Service 클래스를 두는 이유



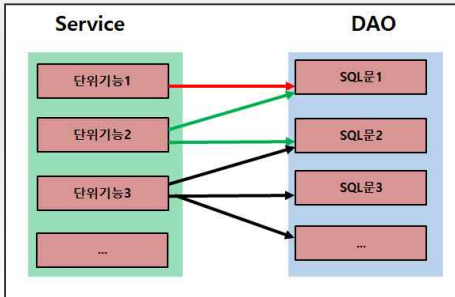
단위 기능

- 사용자 입장에서 하나의 논리적인 기능

웹 애플리케이션의 일반적인 단위 기능 예

- 게시판 글 조회 시 해당 글을 조회하는 기능과 조회 수를 갱신하는 기능
- 쇼핑몰에서 상품 주문 시 주문 상품을 테이블에 등록 후 주문자의 포인트를 갱신하는 기능
- 은행에서 송금 시 송금자의 잔고를 갱신하는 기능과 수신자의 잔고를 갱신하는 기능

일반적인 모델 구조



3. 모델2로 답변형 게시판 구현하기

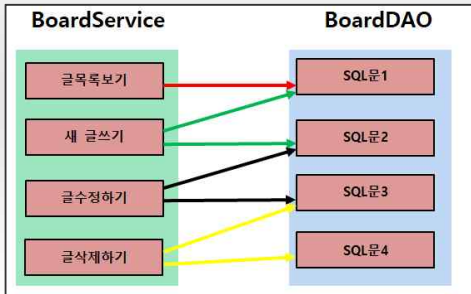
- Service 클래스의 메서드를 이용해 큰 기능을 단위 기능으로 나눈 후 Service 클래스의 각 메서드는 자신의 기능을 더 세부적인 기능을 하는 DAO의 SQL문들을 조합해서 구현
- 유지보수나 시스템의 확장성 면에서 훨씬 유리함

3. 모델2로 답변형 게시판 구현하기

게시판 단위 기능

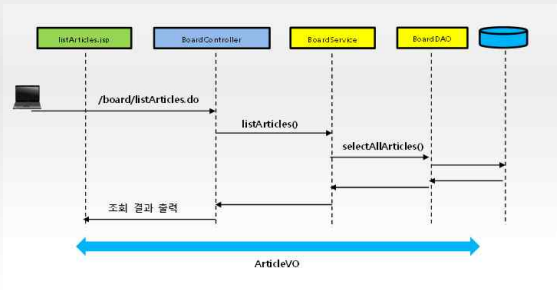
- 새 글 쓰기
- 글 보기
- 글 수정하기
- 글 삭제하기

답변형 게시판의 모델 구조



- 게시판 글 목록 보기 구현

글목록 보기 흐름도





감사합니다.