



모델 2 기반의 MVC 패턴

이번 Chapter에서는 애플리케이션 작성 시 로직을 가지는 모델(Model), 화면에 내용을 표시하는 뷰(View), 그리고 프로그램의 흐름을 제어하는 컨트롤러(Controller)로 나뉜 작업 패턴을 제공하는 MVC(Model-View-Controller) 패턴에 대해 학습한다.



모델 2와 MVC 패턴의 개요

모델 1 vs. 모델 2

**MVC 패턴(Model-View-Controller
pattern)**



컨트롤러인 서블릿에 사용자의 요청을 명령어로 전달-커맨드 패턴

요청 파라미터로 명령어를 전달하는 방법

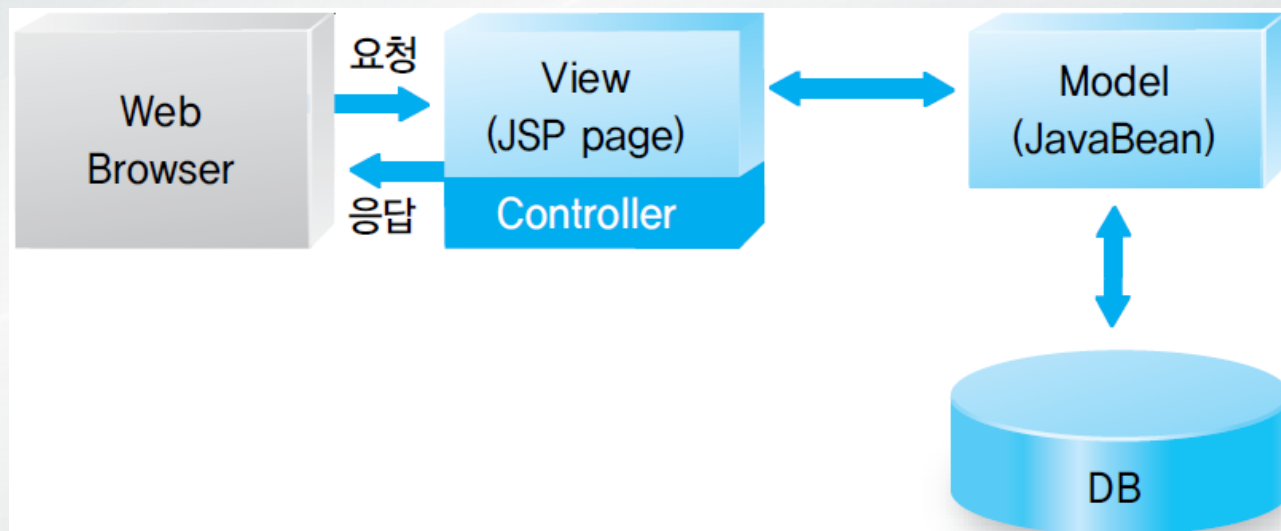
요청 URI 자체를 명령어로 전달하는 방법

참고 사항-UML 다이어그램

모델 1 vs. 모델 2

모델 1 구조

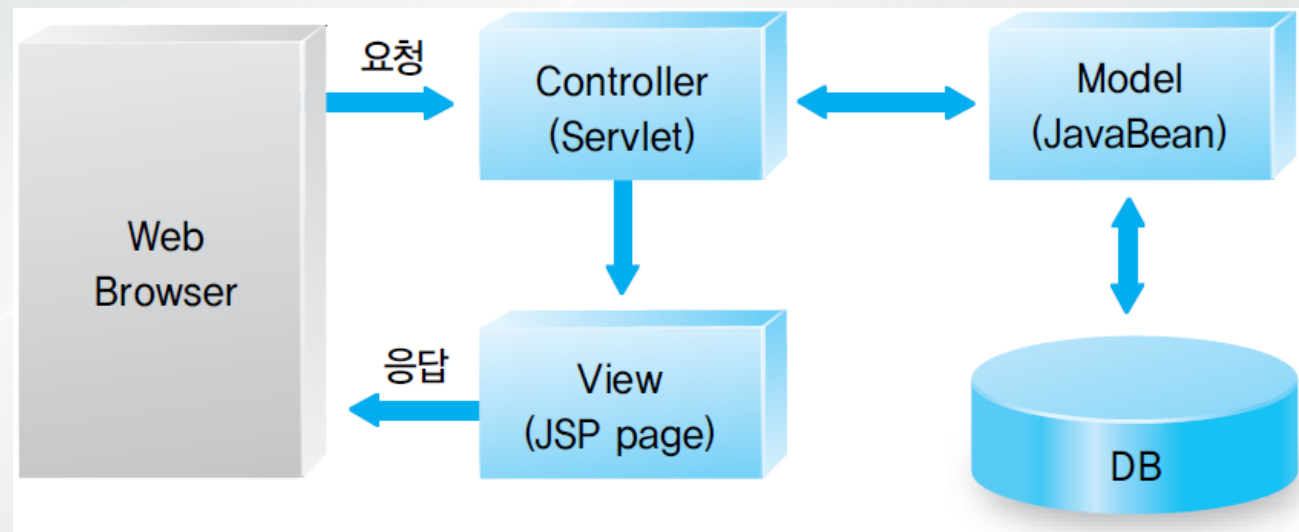
- 웹 브라우저의 요청(request)을 받아들이고, 웹 브라우저에 응답(response)하는 것을 JSP페이지가 단독으로 처리하는 구조



모델 1 vs. 모델 2

모델 2 구조

- 요청(request) 처리, 데이터 접근(data access), 비즈니스 로직(business logic)을 포함하고 있는 컨트롤러와 뷰는 엄격히 구분



모델 2 구조 ▶



모델 2와 MVC 패턴의 개요

🔄 MVC 패턴(Model-View-Controller pattern)

- 전통적인 GUI(Graphic User interface) 기반의 애플리케이션을 구현하기 위한 디자인 패턴
- MVC 구조는 사용자의 입력을 받아서 입력에 대한 처리를 하고, 그 결과를 다시 사용자에게 표시하기 위한 최적화된 설계를 제시



모델 2와 MVC 패턴의 개요

🔄 MVC 패턴(Model-View-Controller pattern)

▣ MVC 패턴의 요소

■ 모델(Model) : 로직을 가지는 부분

- DB와의 연동을 통해서 데이터를 가져와 어떤 작업을 처리
- 처리한 작업의 결과를 데이터로서 DB에 저장하는 일을 처리
- 자바빈(JavaBean), 처리 로직인 자바 클래스(Java class)가 이에 해당



모델 2와 MVC 패턴의 개요

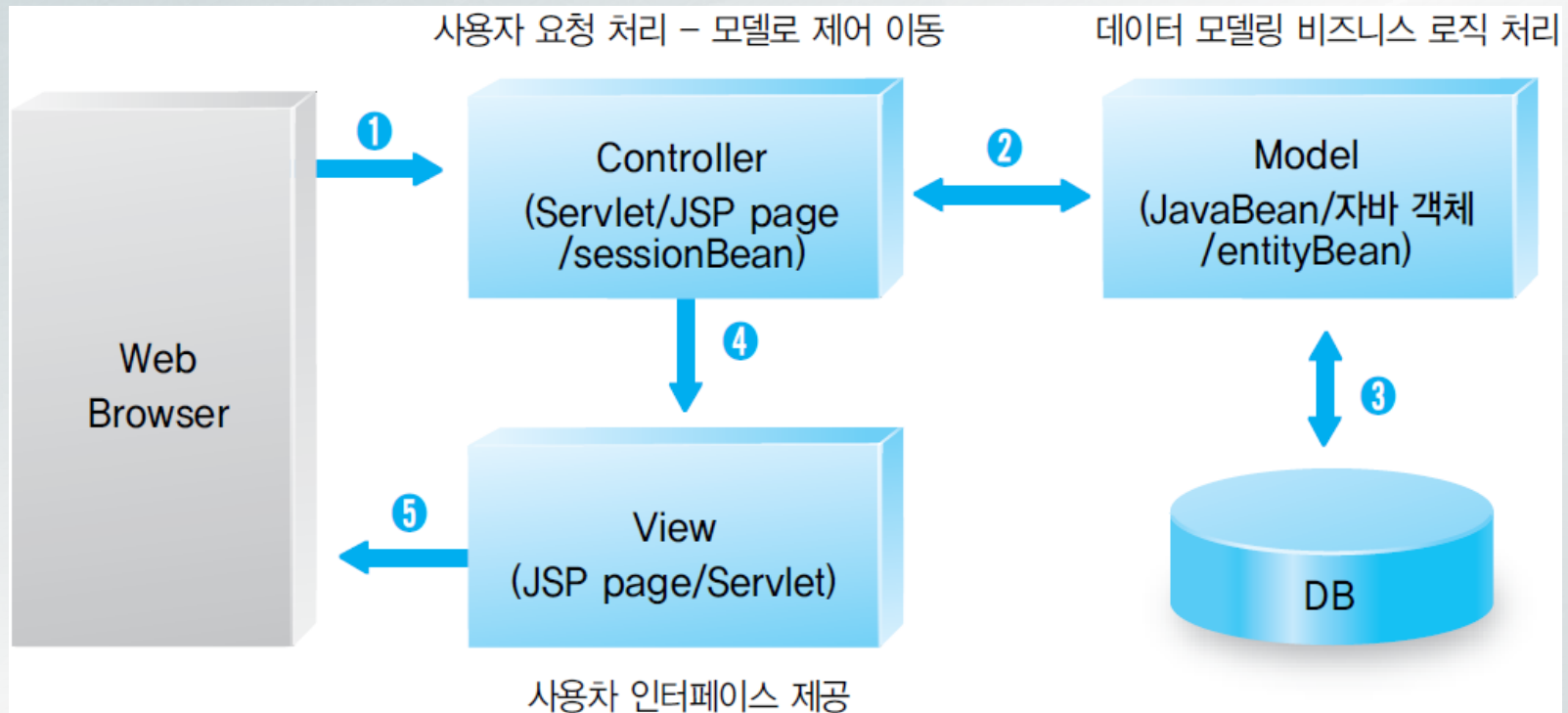
🔄 MVC 패턴(Model-View-Controller pattern)

▣ MVC 패턴의 요소

- 뷰(View) : 화면에 내용을 표시
 - 정보를 보여주는 역할만을 담당
 - JSP 페이지가 이에 해당
- 컨트롤러(Controller) : 어플리케이션의 흐름을 제어
 - 사용자의 요청을 받아서 모델(Model)에 넘겨줌
 - 모델(Model)이 처리한 작업의 결과를 뷰(View)로 보냄
 - 서블릿(Servlet)이 이에 해당

MVC 패턴(Model-View-Controller pattern)

MVC 패턴의 구조



▲ JSP 기반 웹 애플리케이션 구조에 MVC 패턴 적용






모델 2와 MVC 패턴의 개요

MVC 패턴(Model-View-Controller pattern)

MVC 패턴의 구성 요소

컨트롤러(Controller) : 서블릿(Servlet)

-  웹브라우저의 요청을 받는 진입점
-  사용자의 요청을 받아서 요구사항을 분석 후 로직 처리를 모델로 보냄
-  로직의 처리 결과를 모델로부터 받아서, 사용자에게 응답하기 위해 뷰로 보냄



모델 2와 MVC 패턴의 개요

MVC 패턴(Model-View-Controller pattern)

MVC 패턴의 구성 요소

컨트롤러(Controller)의 작업 처리 과정

- ① 웹 브라우저의 요청을 받음
 - 웹브라우저의 요청은 서블릿의 서비스 메소드인 doGet() 또는 doPost()메소드가 받음
- ② 웹 브라우저가 요구하는 작업을 분석
 - 사용자가 요구한 작업에 맞는 로직이 실행되도록, 웹브라우저의 요구 작업을 분석
- ③ 모델을 사용해서 요청한 작업을 처리
 - 요청한 작업에 해당하는 로직을 처리



모델 2와 MVC 패턴의 개요

🔄 MVC 패턴(Model-View-Controller pattern)

▣ MVC 패턴의 구성 요소

■ 컨트롤러(Controller)의 작업 처리 과정

- ④ 로직 처리 결과를 request객체의 속성에 저장
 - request 객체의 속성에 처리 결과를 저장
 - 처리 결과는 같은 request 객체 영역에서 공유
- ⑤ 적당한 뷰(JSP페이지)를 선택 후 해당 뷰로 포워딩(forwarding).
 - 처리 결과를 저장한 request객체를 뷰로 전달



모델 2와 MVC 패턴의 개요

🔄 MVC 패턴(Model-View-Controller pattern)

▣ MVC 패턴의 구성 요소

■ 뷰(View) : JSP 페이지

- 요청에 대한 응답 결과를 표시
- JSP 페이지의 request는 컨트롤러(Controller)인 서블릿(Servlet)과 같은 객체로 공유
 - `${requestScope.result}` 또는 `request.getAttribute("result")`와 같이 사용해서 결과를 화면에 표시



모델 2와 MVC 패턴의 개요

🔄 MVC 패턴(Model-View-Controller pattern)

▣ MVC 패턴의 구성 요소

■ 모델(Model) : 자바빈

- 컨트롤러(Controller)가 넘겨준 로직 처리
- 모델(Model)의 작업 처리과정
 - ① 컨트롤러(Controller)의 요청을 받음
 - ② 모델에서 로직을 처리
 - ③ 처리한 로직의 결과를 컨트롤러(Controller)로 반환