

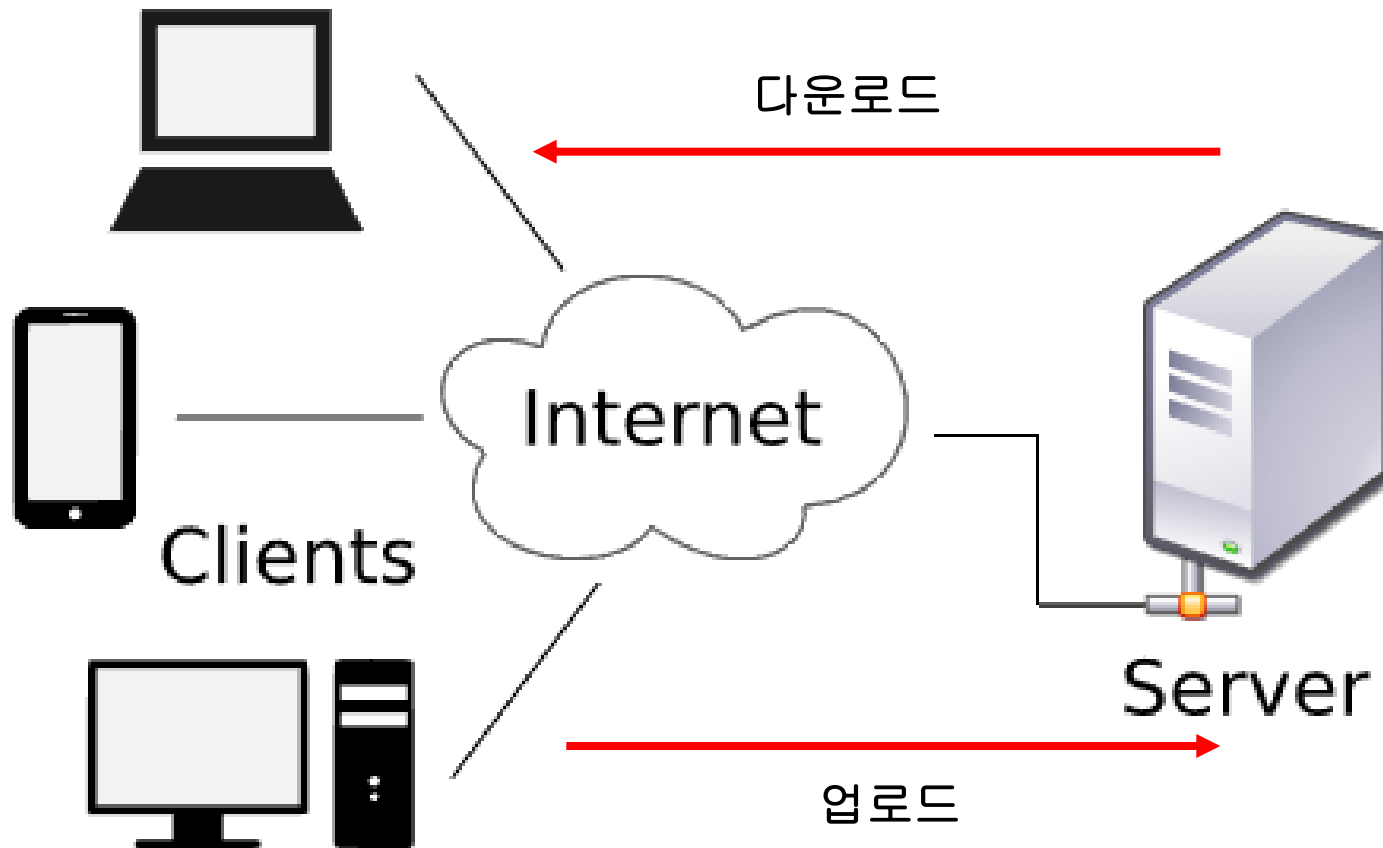


JSP 프로그래밍

15차시

- 자료실
- MVC 패턴의 이해

지난시간 복습





□ 파일 업로드는 POST만 이용 가능

- restful 형식에서도 put은 안됨. post만 가능함

□ form태그의 enctype 속성 추가

- enctype="multipart/form-data"

- enctype은 폼을 전송할 때 인코딩 방법을 지정
- 기본값은 application/x-www-form-urlencoded로
- 모든 문자를 인코딩하는 것

□ Upload를 도와주는 라이브러리 사용

- cos.jar 를 프로젝트에 포함시키자

- 업로드 속도, 중복 이름 처리 등



□ 업로드 한 파일을 다운로드 받을 수 있는 게시판

□ 파일을 업로드 하는 방법은 다양하고, 상황에 따라 선택지가 다르다. 각각의 장단점은 존재한다.

○ Ex) 파일 분류: 날짜/파일이름, 파일seq/파일이름 등

○ 파일 업로드: HTTP, FTP 이용 등

○ 파일 저장: 파일시스템, DB저장



□ 파일 저장 - 파일 시스템

- 파일 자체는 따로 저장하고 메타 정보만 DB에 기록
- 파일에 대한 관리가 편하고, 로직의 구현이 쉬움
- 가장 많이 사용됨

□ 파일 저장 - DB에 저장

- DB에 파일의 바이너리를 저장
- 보안처리가 쉽고, 트랜잭션에 포함 가능
- DB의 용량이 증가되고, DB에 병목지점이 생김



□ 파일 저장 - 파일 시스템

○ 스텝1: 임시 공간에 파일업로드(완료)

○ 스텝2: DB에 메타데이터 저장

- 업로더, 파일 저장 위치, 업로드 된 원본 파일명

○ 스텝3: 스텝2가 성공하면 파일 옮기기

○ 스텝4: 목록



□ 파일 저장 - 파일 시스템

○ 스텝1: 임시 공간에 파일업로드(완료)

○ 스텝2: DB에 메타데이터 저장

- 업로더, 파일 저장 위치, 업로드 된 원본 파일명

○ 스텝3: 스텝2가 성공하면 파일 옮기기

○ 스텝4: 목록



- 모델-뷰-컨트롤러(Model-View-Controller)
- SW공학에서 사용되는 디자인 패턴
- 어플리케이션의 역할을 모델, 뷰, 컨트롤러로 구분
- 사용자 인터페이스로부터 비즈니스 로직을 분리하여 개발하여 서로 영향 없이 쉽게 수정이 가능한 어플리케이션을 만들 수 있다.



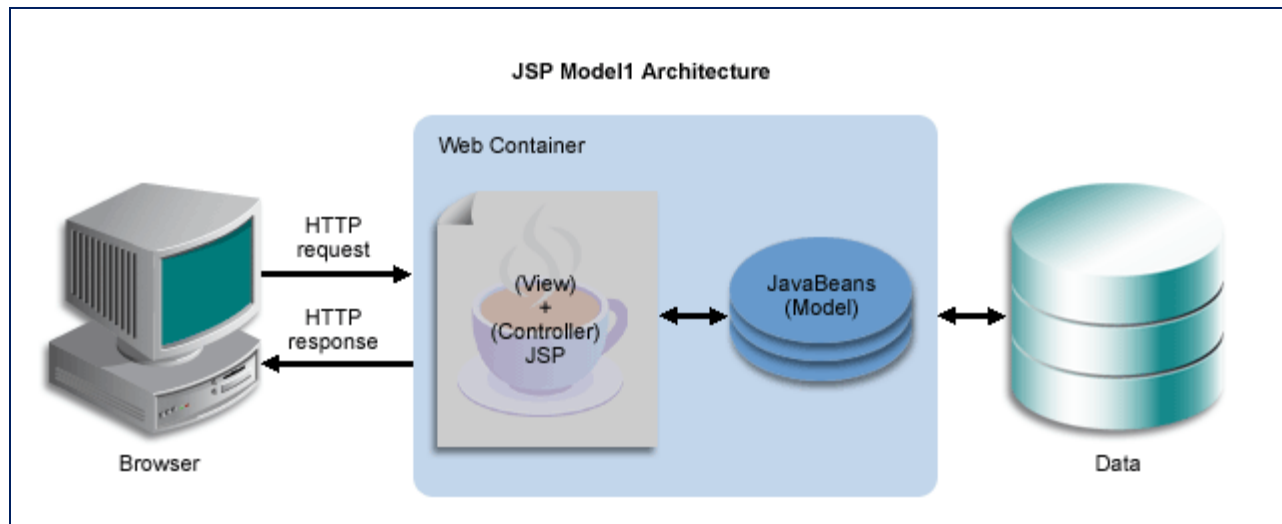
- 모델: 정보(데이터)
- 뷰: 텍스트, 체크박스 등과 같은 사용자 인터페이스
- 컨트롤러: 데이터와 비즈니스 로직 사이의 상호동작



- 모델1(Model1) VS 모델2(Model2)
- 여기서 말하는 모델1, 모델2는 MVC에서 나오는 Model과는 다른 의미라고 생각하는게 편하다.



□ 모델 1 (Model1)





□ 모델2(Model2)

