

●웹개발은 서블릿이나 JSP만으로도 가능하다

MVC2 패턴은 JSP라는 기술로만 개발을 할 경우 많은 단점을 극복한 개발패턴이다

처음에는 복잡하게 느껴질 수 있으나

JSP만으로 개발하는 것이 개발자에게 유리했다면 MVC2패턴과 스프링은 나오지 않았을 것이다.

그러므로 긍정 ^^ 의 마음으로 이 기술을 배우도록 하자

서블릿과 Jsp는 동적으로 페이지를 만들어 내는 자바의 기술이다

웹서버란 클라이언트의 요청이 있을 때 해당하는 페이지를 제공하는 것이다.

정적페이지 (미리 만들어놓은 html, 이미지 등)

동적페이지 (요청을 처리하는 하나의 작은 프로그램이다. 요청이 들어올 때 프로그램이 실행되고, 실행된 결과를 응답하게 되어있다)

서블릿과 JSP의 한계

서블릿으로 개발할 때는 뷰(View)화면을 위한 HTML을 만드는 작업이 자바 코드에 섞여서 지저분하고 복잡했다.(html만드는 것은 많은 단순작업이 필요함)  
JSP를 사용한 덕분에 뷰를 생성하는 HTML 작업을 깔끔하게 가져가고, 중간중간 동적으로 변경이 필요한 부분에만 자바 코드를 적용했다.

JSP개발로 HTML작업이 편하고 깔끔하게 되었으나

JSP가 너무 많은 역할을 하고 있다, JSP코드는 재사용할 수 없다.

( 요청/응답, 비즈니스 로직 (데이터베이스 처리 등) , HTML응답 만들기)

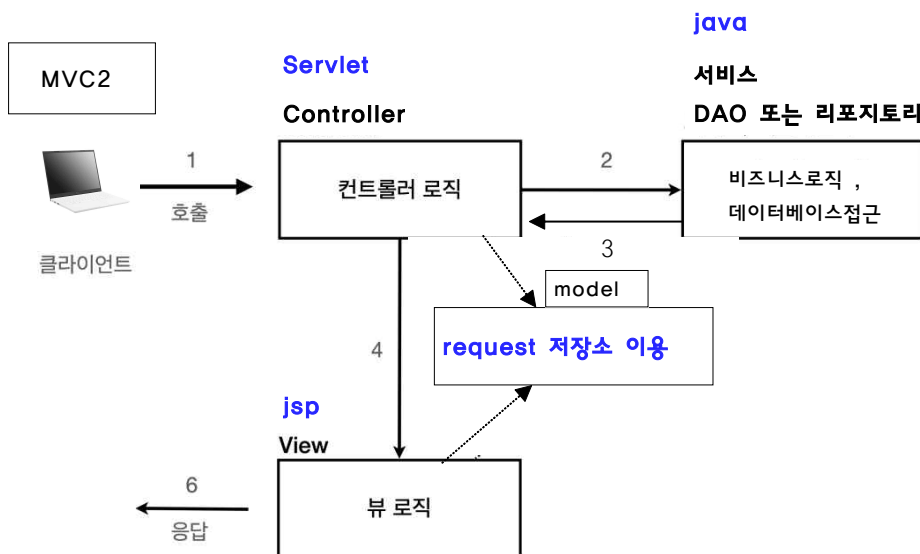
●스파게트 코드 : 자바코드와 html이 혼재한 코드는 재사용이 불가능하고 너무 많은 처리를 하고 있어 요구사항이 변경되거나 많은 양의 개발시 유지보수가 너무 힘든 상황을 만나게 된다.

MVC 패턴의 등장 ( 역할을 나눠서 코드를 작성함 )

서블릿 요청과 응답

비즈니스 로직은 자바로 사용함 Model을 만드는 역할을 함 (재사용 가능)

JSP는 목적에 맞게 HTML로 화면(View)을 그리는 일에 집중하도록 하자.



MVC 패턴은 지금까지 학습한 것 처럼 하나의 서블릿이나, JSP로 처리하던 것을 컨트롤러(Controller)와 뷰(View)라는 영역으로 서로 역할을 나눈 것을 말한다. 웹 애플리케이션은 보통 이 MVC 패턴을 사용함 !! 매우 중요

Controller(컨트롤러): HTTP 요청을 받아서 파라미터를 검증하고, 비즈니스 로직을 실행 ,  
뷰에 전달할 결과 데이터를 조회해서 모델에 저장함

Model (모델): 뷰에 출력할 데이터를 담아둔다. (request 객체를 저장소로 사용함)

View (뷰): 모델에 담겨있는 데이터를 사용해서 화면을 그리는 일에 집중한다. HTML을 생성하는 부분 담당

<< 과제 >>

학생조회jsp -> mvc패턴으로 개발해 보기

실기시험으로 볼 예정 ( 이 만큼은 손에 익힐 필요가 있음)