JSP

1. MVC2 패턴
2. URL맵핑의 변화
3. MVC 아키텍쳐

MVC2 전체적인 컴포넌트 설계

3개를 다 나눔

M:model

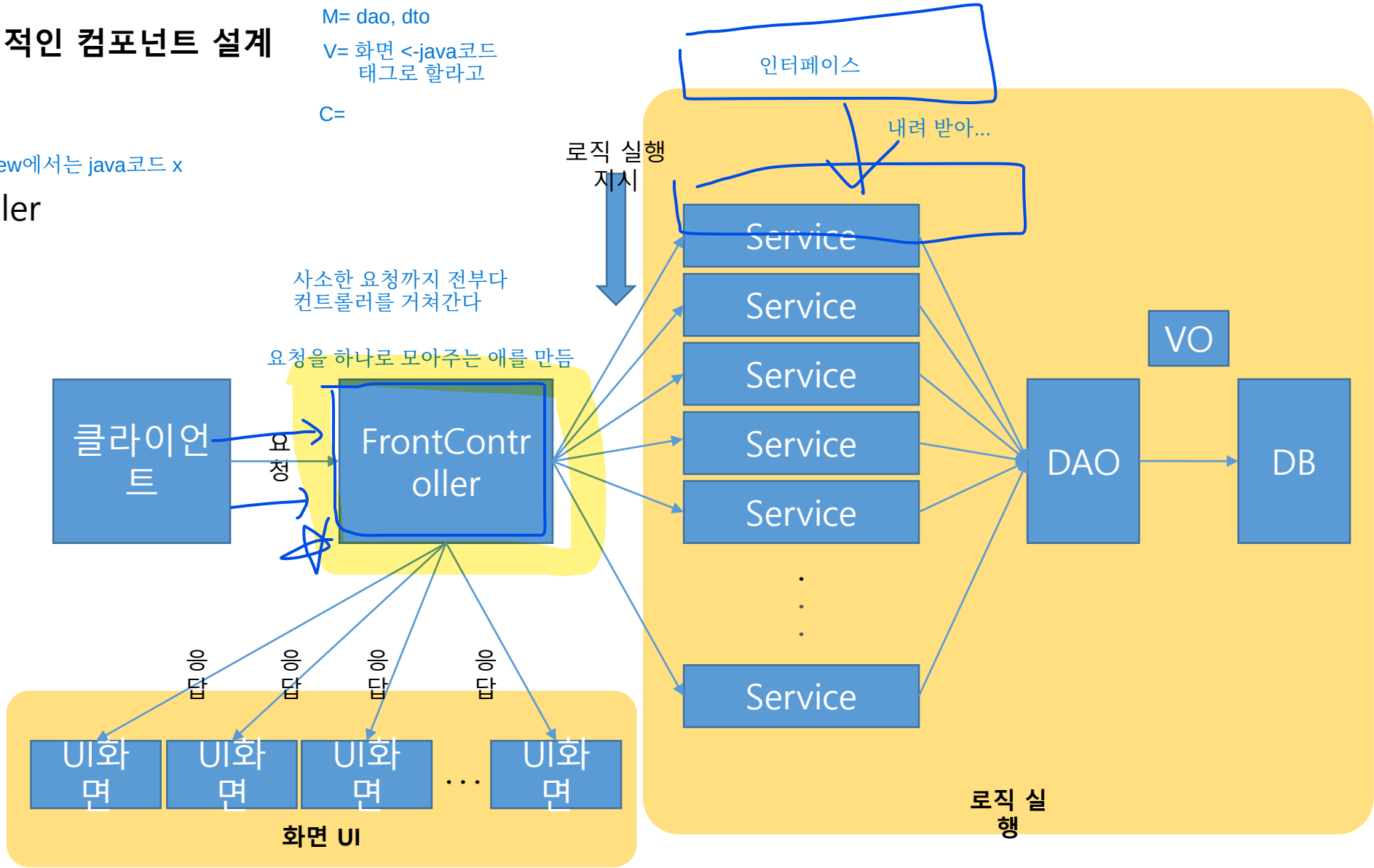
V: view view에서는 java코드 x

C: Controller

M= dao, dto

V= 화면 <-java코드 태그로 할라고

C=



*** URL-Pattern**

1. 디렉토리 패턴: 디렉토리 형태로 서버의 해당 컴포넌트를 찾아서 실행하는 구조입니다.

ex) http://localhost:8181/cr/Hello --> /Hello 서블릿
http://localhost:8181/cr/World --> /World 서블릿

2. 확장자 패턴: 확장자 형태로 서버의 해당 컴포넌트를 찾아서 실행하는 구조입니다.

ex) http://localhost:8181/cr/Hello.do --> *.do 서블릿
http://localhost:8181/cr/World.do --> *.do 서블릿

.board로 끝나면 컨트롤러로 들어온다..

```
@WebServlet("*.board")
public class BoardController extends HttpServlet {
```

* MVC Model 2 Architecture

- 모델2 구조는 웹 브라우저의 요청을 하나의 서블릿이 받으며 서블릿은 그 요청을 알맞게 처리한 후, 그 결과를 보여줄 JSP 페이지로 **포워딩**합니다.

- 이 구조의 특징은 웹 브라우저의 모든 요청을 단일 진입점, 즉 하나의 서블릿에서 처리한다는 점입니다. 하나의 서블릿이 모든 요청을 받기 때문에 서블릿은 웹 브라우저의 요청을 구분하는 방법이 필요합니다.

- **MVC 모델2 구조의 핵심은**
모델(Model)은 비즈니스와 관련된 로직만 처리하면 되며 사용자에게 보일 화면이나 요청의 흐름 제어에 대해서는 전혀 처리하지 않으며,

뷰(View)는 사용자에게 알맞은 화면을 보여주는 역할만 수행할 뿐, 비즈니스 로직이나 요청 흐름 제어 등을 처리하지 않습니다.

컨트롤러(Controller)는 사용자의 요청에 대해서 알맞은 모델을 사용하고 사용자에게 보여줄 뷰를 선택합니다.

- MVC 모델2 구조를 사용함으로써 코드의 유지보수가 쉬워지며 어플리케이션을 쉽게 확장할 수 있습니다.

* MVC의 컨트롤러: 서블릿

- 모델 2 구조에서 서블릿은 MVC 패턴의 컨트롤러 역할을 합니다. 서블릿은 웹 브라우저의 요청과 웹 어플리케이션의 전체적인 흐름을 제어합니다.

- 컨트롤러의 흐름 제어 처리 로직

1. 웹 브라우저가 전송한 HTTP 요청을 받아 요청방식에 맞게 doGet(), doPost()를 호출함.
2. 웹 브라우저가 어떤 기능을 요청했는지 분석함.
3. 모델을 사용하여 요청한 기능을 수행.
4. 모델로부터 전달받은 결과물을 알맞게 가공한 후, request나 session의 setAttribute() 메서드를 이용하여 결과값을 속성에 저장. 이렇게 저장한 결과값은 뷰인 JSP에서 사용함.
5. 웹 브라우저에 결과를 전송할 JSP 페이지를 선택한 후, 해당 JSP로 포워딩(혹은 리다이렉트)함.

* MVC의 뷰: JSP

- 모델 2 구조에서 JSP는 뷰 역할을 담당합니다. 뷰 역할을 하는 JSP는 컨트롤러에서 request 객체나 session객체에 저장된 데이터를 사용하여 웹 브라우저에 알맞을 화면을 출력합니다.

- 뷰 역할을 하는 JSP는 웹 브라우저가 요청한 결과를 보여주는 프레젠테이션의 역할을 할 뿐만 아니라 웹 브라우저의 요청을 컨트롤러에 전달해주는 매개체가 되기도 합니다.

- 즉, 뷰 역할을 하는 JSP는 웹 브라우저가 지속적으로 컨트롤러에 요청을 보낼 수 있는 링크나 폼을 제공해서 웹 브라우저가 업무 흐름에 따라 컨트롤러에게 알맞은 요청을 보낼 수 있도록 합니다.

* MVC의 모델

- 컨트롤러는 서블릿을 통해 구현하고 뷰는 JSP를 통해서 구현하는데 모델은 명확하게 어떤 것을 통해서 구현한다는 규칙은 없습니다. 단지 비즈니스 로직을 처리해주면 모델이 될 수 있습니다.

- 컨트롤러의 서블릿이 웹 브라우저의 요청을 분석하여 알맞은 모델을 호출하면서부터 모델의 기능이 시작됩니다. 모델은 컨트롤러가 요청한 작업을 처리한 후 알맞은 결과를 컨트롤러에게 전달하는데, 이때 처리한 결과값을 저장하는 객체로 보통 자바빈을 사용합니다.