KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

요청 포워딩, 리다이렉트

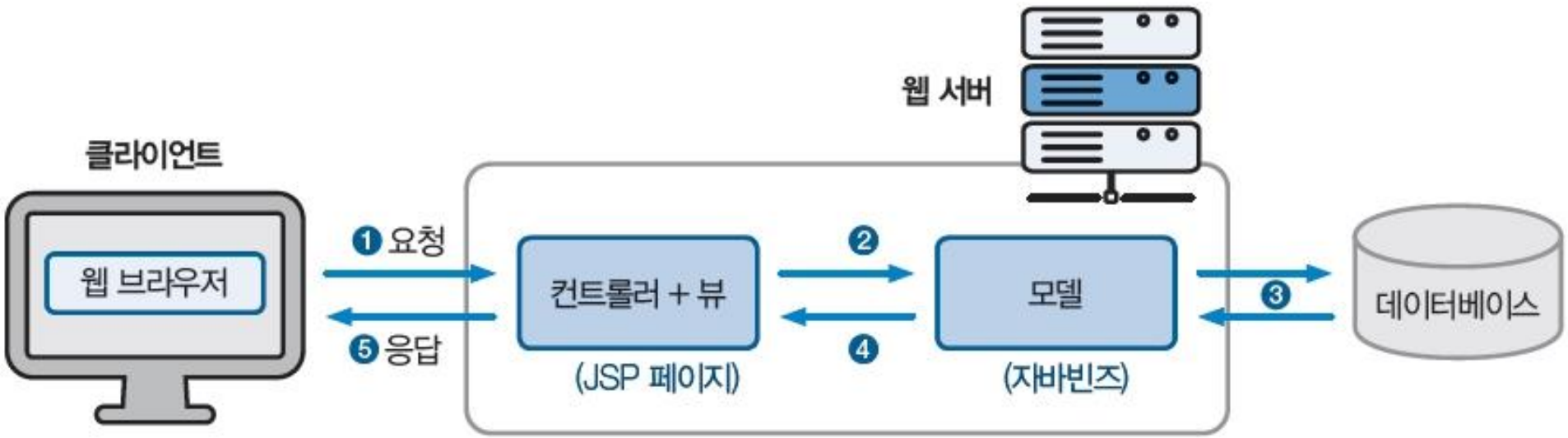
[KB] IT's Your Life

KB 국민은행

모델 1

- 기존의 JSP로만 구현한 웹 애플리케이션
- 웹 브라우저의 요청을 JSP 페이지가 받아서 처리하는 구조
- JSP 페이지에 비즈니스 로직 처리 코드와 웹 브라우저에 결과를 출력하는 코드가 섞이는 것
- 모델 1에서는 JSP가 핵심 역할을 수행함

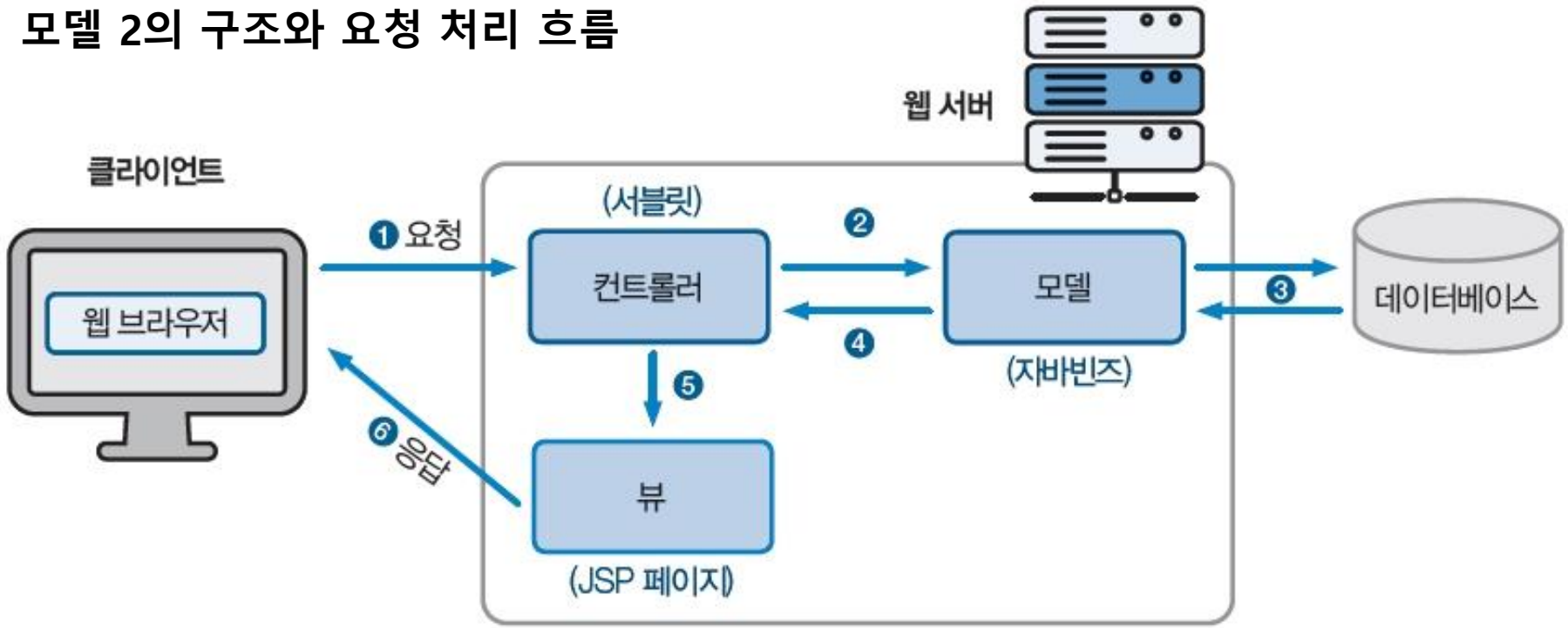
- 모델 1의 구조와 요청 처리 흐름



모델2

- 클라이언트의 요청 처리, 응답 처리, 비즈니스 로직 처리 부분을 모듈화한 구조
- 요청에 대한 로직을 처리할 자바빈즈나 자바 클래스인 모델,
  요청 결과를 출력하는 JSP 페이지인 뷰,
  모든 흐름을 제어는 서블릿인 컨트롤러로 나뉘어 웹 브라우저가 요청한 작업을 처리함
- 모델 2에서는 서블릿이 중요한 역할을 함

모델 2의 구조와 요청 처리 흐름



MVC

- Model, View, Controller의 약자
- 웹 애플리케이션을 비즈니스 로직, 프레젠테이션로직, 데이터로 분리하는 디자인 패턴
- 웹 애플리케이션에서는 일반적으로 애플리케이션을 비즈니스 로직, 프레젠테이션, 요청 처리 데이터로 분류
  - 비즈니스 로직: 애플리케이션의 데이터, 즉 고객, 제품, 주문 정보의 조작에 사용됨
  - 프레젠테이션: 애플리케이션이 사용자에게 어떻게 표시되는지, 즉 위치, 폰트, 크기
  - 요청 처리 데이터: 비즈니스 로직과 프레젠테이션 파트를 함께 묶음

MVC 패턴의 구성 요소

모델(model)

- 애플리케이션의 데이터와 비즈니스 로직을 담는 객체

컨트롤러(controller)

- 모델과 뷰 사이에 어떤 동작이 있을 때 조정하는 역할
- 웹으로부터 받은 요청에 가장 적합한 모델을 생성하는 것을 처리하는 역할
- 사용자에게 응답하는 적절한 뷰를 선택하여 해당 모델을 전달하는 역할

→ Servlet

뷰(view)

- 사용자에게 모델의 정보(데이터)를 보여주는 역할
- 비즈니스 로직을 포함하지 않으며, 하나의 모델을 다양한 뷰에서 사용

→ JSP

2 요청 포워딩(Request Forwarding)

요청 포워딩의 필요성

- 요청 처리 작업의 모듈화
- 모듈의 재사용성 증가 및 유지 보수의 편이
- MVC 모델(Model2)의 기본 기능

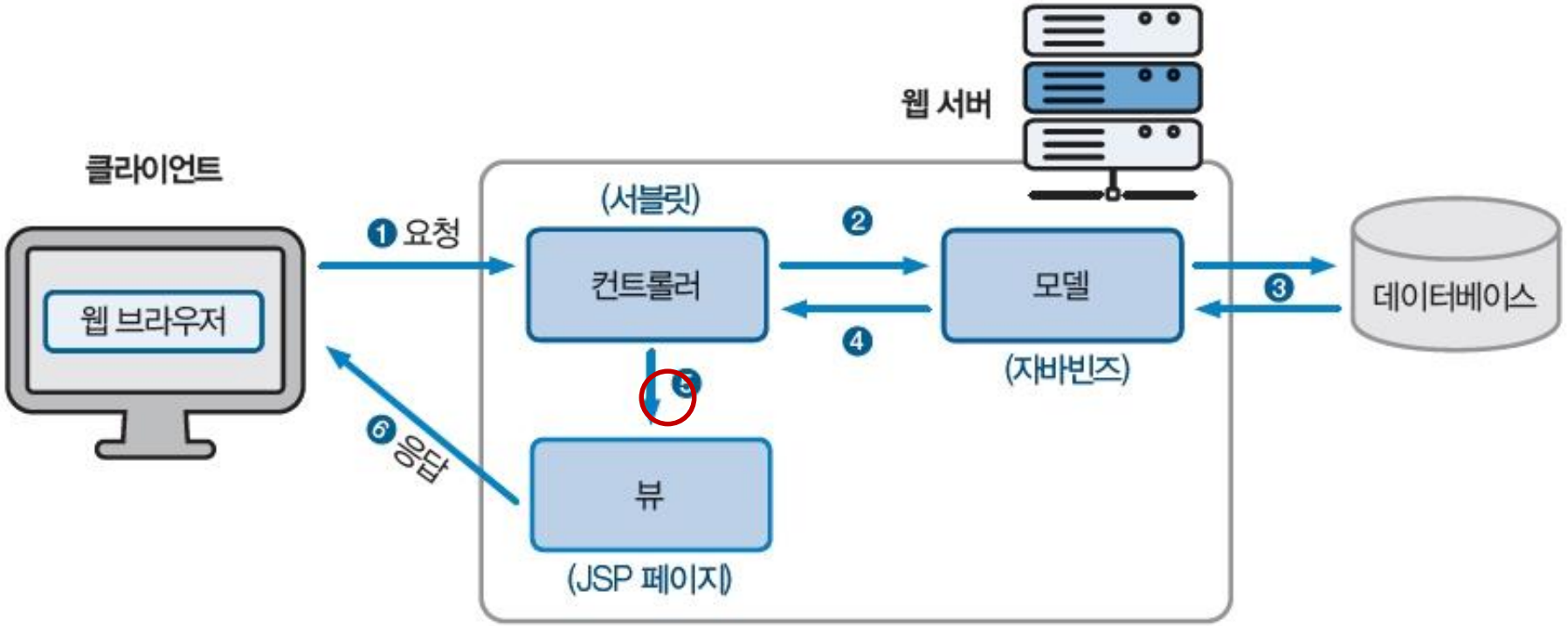
  - 요청 처리: FrontController
  - 응답 처리: JSP(View)

요청 포워딩 방법

- RequestDispatch 클래스를 이용한 forward 방법
- HttpServletResponse 클래스를 이용한 redirect 방법

RequestDispatcher 클래스를 이용한 forward 방법

```
RequestDispatcher dis = req.getRequestDispatcher(target);
dis.forward(req, res);
```



1, 2번의 요청이 같은 HttpServletRequest를 사용
파라미터 값과 setAttribute로 저장한 정보를 공유 → Request scope

forward 방법 실습

- 프로젝트명: ex05

- 패키지: org.scoula.ex05
- 클래스: RequestServlet
- URL 맵핑: /request

RequestServlet.java

```
@WebServlet("/request")
public class RequestServlet extends HttpServlet {

  @Override
  protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

    //속성 설정
    req.setAttribute("username", "홍길동");
    req.setAttribute("useraddress", "서울");

    //forward
    RequestDispatcher dis = req.getRequestDispatcher("/res.jsp");
    dis.forward(req, res);

  }
}
```

res.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>Insert title here</title>
</head>
<body>
username 값: <%=request.getAttribute("username") %><br>
useraddress 값: <%=request.getAttribute("useraddress") %><br>
</body>
</html>
```

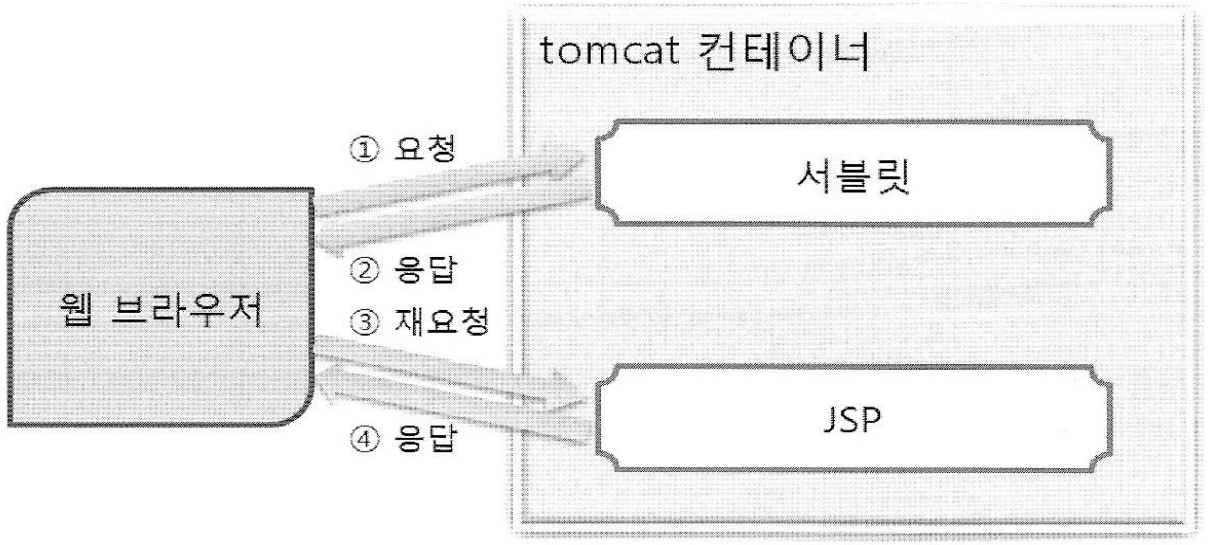
http://localhost:8080/request

username 값: 홍길동
useraddress 값: 서울

HttpServletResponse 클래스를 이용한 redirect 방법

res.sendRedirect(target);

- target: 이동할 페이지

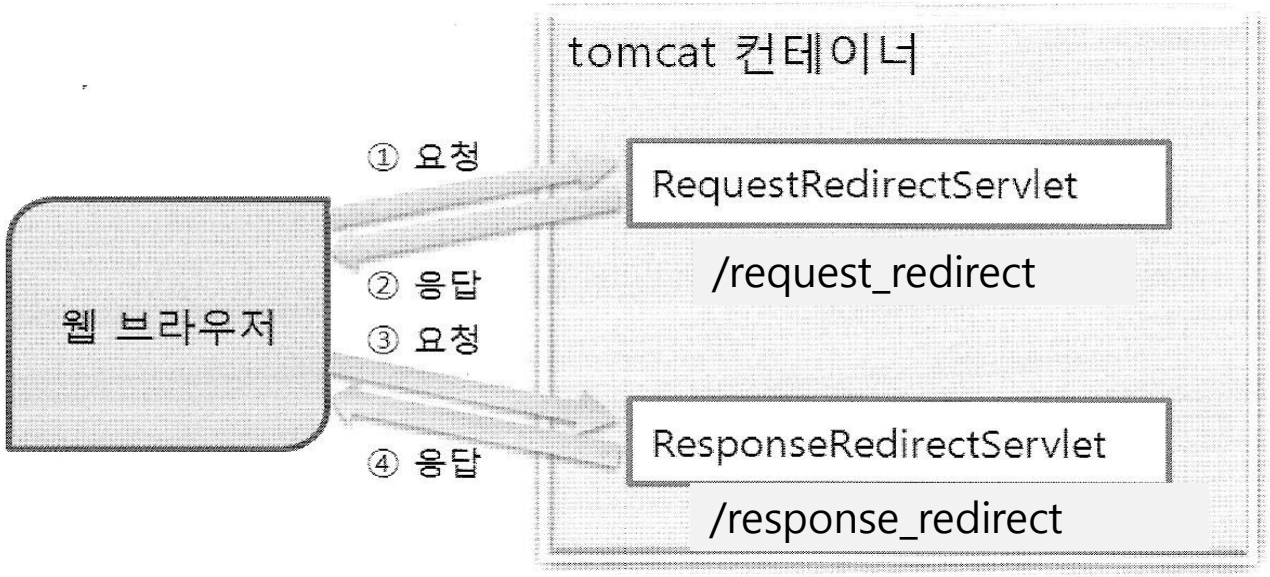


- 첫번째 요청에 대해 응답하고 브라우저가 바로 redirect
- 동일 HttpServletRequest가 아닌 새로운 request가 사용됨
  → Request scope가 다름

redirect 방법 실습

- 패키지: org.scoula.ex05
- 클래스: RequestRedirectServlet
- URL 맵핑: /request_redirect

- 클래스: ResponseRedirectServlet
- URL 맵핑: /response_redirect



RequestRedirectServlet.java

```
@WebServlet("/request_redirect")
public class RequestRedirectServlet extends HttpServlet {
  protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

    //속성 설정
    req.setAttribute("username", "홍길동");
    req.setAttribute("useraddress", "서울");

    //redirect
    res.sendRedirect("response_redirect");

  }
}
```

ResponseRedirectServlet.java

```
@WebServlet("/response_redirect")
public class ResponseRedirectServlet extends HttpServlet {

  protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

    String username = (String) req.getAttribute("username");
    String useraddress = (String) req.getAttribute("useraddress");

    //속성 설정
    req.setAttribute("username", username);
    req.setAttribute("useraddress", useraddress);

    //forward
    RequestDispatcher dis = req.getRequestDispatcher("/redirect_res.jsp");
    dis.forward(req, res);
  }

}
```

redirect_res.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="utf-8">
    <title>Insert title here</title>
</head>
<body>

username 값: <%=request.getAttribute("username") %><br>
useraddress 값: <%=request.getAttribute("useraddress") %><br>

</body>
</html>
```

http://localhost:8080/request_redirect → http://localhost:8080/response_redirect

username 값: null
useraddress 값: null