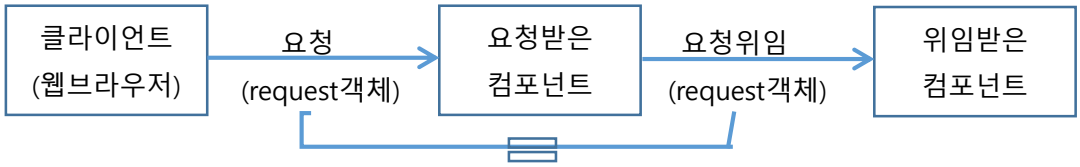
포워딩(Forwarding)

1. RequestDispatcher 클래스 ; 내가 받은 요청을 내가 처리하지 않고 다른 쪽으로 위임해 버릴 때 그것을 포워드 한다고 한다. 그것을 JSP 서블릿에서는 두가지 클래스를 많이 이용한다.
   - ✓ 서블릿 또는 JSP에서 요청을 받은 후 다른 콤포넌트로 요청을 위임 할 수 있다. 그리고 이러한 위임 방법에는 2개의 클래스를 이용 합니다.
   - ✓ 하나는 RequestDispatcher 클래스 이고, 또 하나는 HttpServletResponse 클래스
   - ✓ RequestDispatcher클래스의 경우 요청 받은 **요청객체(request)를 위임하는 컴포넌트에 동일하게 전달**할 수 있다.



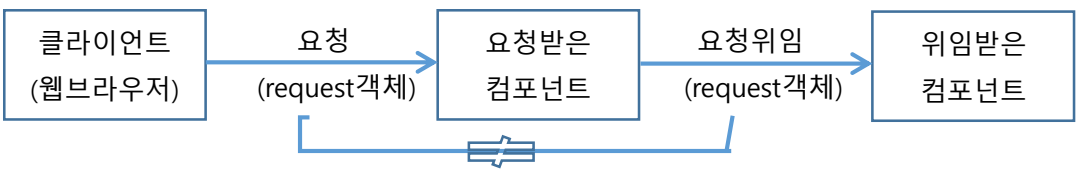
Servlet

```
private void actionDo(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    String id = request.getParameter("id");
    String pw = request.getParameter("id");
    // 로그인 체크하는 로직 수행 후 view단에서 쓸 데이터는 request에 set한다
    request.setAttribute("id", id);
    request.setAttribute("pw", pw);
    request.setAttribute("name", "홍길동");
    RequestDispatcher dispatcher = request.getRequestDispatcher("1_dispatcher/ex1View.jsp");
    dispatcher.forward(request, response);
}
```

Model1까지 jsp안에서 request객체를 set을 했었다.

패턴화를 익혀가는 과정에서 스크립트릿 부분과 자바영역 즉 , view단과 자바 영역을 분리해서 패턴화를 배운다. 그러므로 view단에서 쓸 데이터는 서블릿에서 request에 set한다

2. HttpServletResponse 클래스
   - ✓ RequestDispatcher 클래스와 동일하게 요청을 위임하는 클래스.
   - ✓ RequestDispatcher 클래스와 차이점은 요청 받은 요청객체를 위임 받은 컴포넌트에 전달 하는 것이 아닌, 새로운 요청객체를 생성



```java
private void actionDo(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    // 넘어온 파라미터 id, pw를 이용하여 객체를 만든 후 View단으로 넘긴다
    String id = request.getParameter("id");
    String pw = request.getParameter("pw");
    Member member = new Member(id, pw);
    request.setAttribute("member", member);
    // View단으로 포워드
    RequestDispatcher dispatcher
                    = request.getRequestDispatcher("1_dispatcher/ex2_View.jsp");
    dispatcher.forward(request, response);
    response.sendRedirect("1_dispatcher/ex2_View.jsp?id=aaa&pw=111");//request객체가 새롭게
```

```java
response.sendRedirect("1_dispatcher/ex2_View.jsp?id=aaa&pw=111")
```

위 로직은 프로세스 단에서 id와 pw를 받을 수 있다

Forward를 하고 싶으면 위에서 session을 생성을 해줘야 한다. 하지만 forward는 자바에서 못 쓰는데, 문법에 따라 이용 할 수 있다.

자바에서 forward를 하기 위해서 객체가 필요하다 (dispatcher)

//1. 객체 생성

```java
RequestDispatcher dispatcher =
request.getRequestDispatcher("1_dispatcher/ex2_View.jsp");
```

new해서 새롭게 만들지 않고, 기존에 만들어 놓은 request 객체를 그대로 안고가는 문법이다.

//2. Forward 실행

```java
dispatcher.forward(request, response);
```

Request 객체는 파라미터에 값을 넣어 전달하기도 하고 attribute에 set하여 전달하기도 한다

Q1)

Forward를 하면 새로운 request가 아닌 기존 요청이 유지 된다. Forward를 안하면 요청이 새로 들어가 기존 request가 지워진다. 그럼 언제 쓰는 걸까>

A1)

Forward는 redirect와 반대되는 개념

기존 요청을 그대로 받아들여서 처리하는게 효율적일 때 사용

고객전화 생각해보기. Forward는 고객사에서 연결이 되지만 redirect는 고객이 다시 전화를 끊고 다른 곳으로 전화하는 개념