

221025 수업 -2

🕒 작성일시	@2022년 10월 25일 오전 11:50
≡ 키워드	Servlet
📄 유형	
📎 자료	
☑ 복습	<input type="checkbox"/>

링크는 다 GET 방식

default.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Servlet 요청하기</title>
</head>
<body>
<h3>요청 처리하기</h3>
<h3>getContextPath</h3>
<%= request.getContextPath()%>
<hr>
<a href="<%= request.getContextPath()%>/simple">simple요청하기(GET)</a>
</body>
</html>
```

web.xml에

```
<servlet>
  <servlet-name>simplecontroller</servlet-name>
  <servlet-class>com.SimpleController</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>simplecontroller</servlet-name>
  <url-pattern>/simple</url-pattern>
</servlet-mapping>
```

- http://localhost:8090/WebServlet_1/simple 서버에 요청이 되면
- com.SimpleController 클래스를 simplecontroller라는 이름으로 저장
 - /simple 주소를 simplecontroller이름의 클래스로 매칭

SimpleController.java 파일의

```
public class SimpleController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public SimpleController() {
        super();
    }

    public ServletConfig getServletConfig() {
        return null;
    }
}
```

```

    }

    //요청이 GET 방식인 경우 자동 실행되는 함수
    //요청에 어떠한 것도 지정하지 않으면 모든 요청을 GET 판단
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("클라이언트 요청 : GET");
    }

    //요청이 posst 방식인 경우 자동 실행되는 함수
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("클라이언트 요청 : POST");
        doGet(request, response);
    }
}
}

```

서블릿 조건

- 일반 java 파일이 extends HttpServlet 을 상속하는 경우 서블릿이라 칭함
 - 웹 전용이라 부름
- 웹 요청(request), 응답 (response) 객체를 사용할 수 있다.
- servlet은 url에 바로 요청이 안됨
 - 특정 요청이 필요하므로 요청 주소와 일치하는 객체를 찾으려 web.xml에서 매핑함

매핑 작업

1. web.xml : 구방식
2. @WebServlet (현재 ...) 매핑작업

doGet() 함수와 doPost()

- request, response 객체를 받도록 설계됨
 - doGet(HttpServletRequest request, HttpServletResponse response)
 - 함수 안에서 request와 response 사용해 작업가능

doGet()

- JSP에서 하는 방식 그대로 사용
1. 한글 처리
 2. 데이터 받기 (요청 의도 파악)
 3. 로직(업무수행) >> service 수행
 4. 요청 완료에 따른 결과를 저장
 - 결과를 저장 : application, session, request

5. 저장한 결과를 JSP 페이지에게 전달해서 UI를 구성하도록

- forward 방식으로 JSP까지 전달
- RequestDispatcher -> View 페이지 주소 지정

```
RequestDispatcher dis = request.getRequestDispatcher("/simpleview.jsp");
```

- 특정 페이지에 원하는 정보를 전달

```
dis.forward(request, response);
```

request 객체 주소를 파라미터로 같이 전송

request 객체 주소를 파라미터로 같이 전송

보안 폴더

- webapp의 파일들은 사용자가 주소만 입력하면 접근 가능
- WEB-INF 안의 파일은 접근 불가

처리순서

1. 한글처리

- a. request.setCharacterEncoding("UTF-8");

2. 요청받기

- a. String cmd = request.getParameter("cmd");

3. 요청 판단

- a. 로직

4. 결과 저장

- a. List<board> boardlist = dao.selectBoardList();
- b. request.setAttribute("list", boardlist);

5. View 지정

- a. RequestDispatcher dis = request.getRequestDispatcher(viewpage);

6. View에게 값 전달

- a. dis.forward(request, response);
 - request 객체를 view에게 통으로 전달