JSP 학습 필기용

* 1. **웹 프로그램이란 ?**

브라우저에서 다른 컴퓨터로 정보를 요청 (Request)하면 서버는 응답(Response)를 한다.

즉, 인터넷 서비스를 이용해서 서로 다른 구성요소들(PC 등)이 통신할 수 있는 프로그램이 웹 프로그램이다.

* 1. **프로토콜(Protocol)과 IP**

프로토콜 : 통신을 하기위한 규약 ex. HTTP, FTP, SMTP, POP 등

**HTTP** : 브라우저에서, 웹 사이트에 접속을 하면 하이퍼링크를 통해 그림, 글 등의 정보를 제공하는 규약

FTP(File Transfer Protocol) : 파일 자체를 전송하는 규약

IP 주소 : 숫자로 표현된 인터넷의 특정한 주소

DNS : IP주소를 매핑한 이름 주소( naver.com , google.com 등)

Port : 웹 서버의 여러 프로그램 중에 어떤 프로그램으로 들어갈 지 정하는 것. (기본 포트 80)

Index.html : 특정 웹 사이트 접속 시 가장 기본적으로 실행되는 경로

* 1. **웹 프로그래밍의 동작 원리**

유저가 웹 서버로 요청을 보낸다.

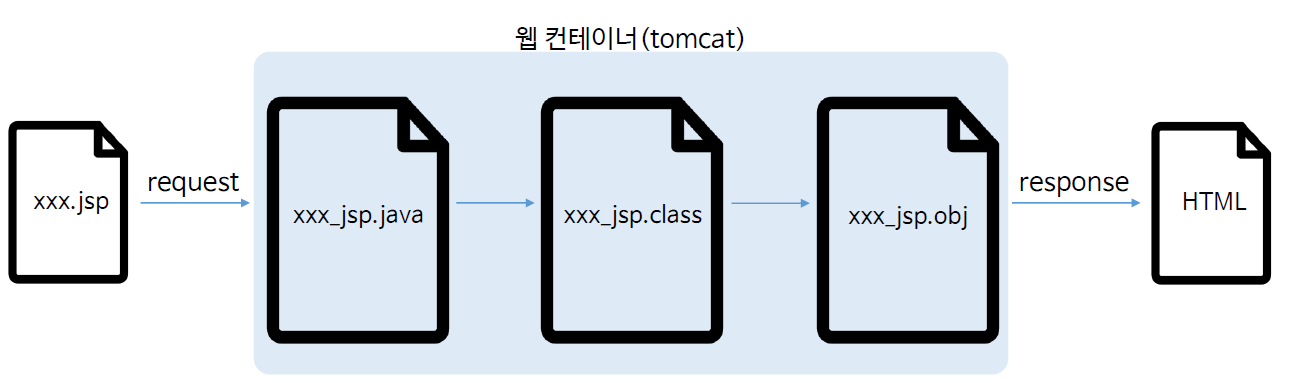
서버는 요청을 판단해서

1. 데이터 베이스에서 query를 날려서 result를 얻은 후 유저에게 response를 보낸다.
2. 정적인 데이터인 html로 응답한다.
3. 동적인 데이터인 웹 컨테이너로 데이터를 가공해, 새로운 데이터를 뽑아낸 후 Html로 응답한다..

**3.1 웹 컨테이너 구조**

WAS : Web Application Server 속에 Web Container가 있다.

개발자가 만든 jsp 파일을 Web Container 서버가 jsp.java로 변환, 컴파일해서 .class파일로 변환, .obj파일로 변환해서 그 결과가 html로 응답된다.



**3.2 JSP 파일 작성**

Jsp 파일은 Html 파일과 형식이 같고 호환이 된다.

**4.1**

Servlet : jsp와 달리 순수 java 파일.

Servlet 파일 내에서 html 태그를 이용해서 html 출력이 가능하다.

**5.1 servlet mapping**

Jsp : view 부분을 주로 개발

Servlet : model 부분 개발

Servlet 맵핑이란 : 브라우저의 요청에 서버에서 적당한 servlet을 골라야 한다.

Full path : Servlet의 정확한 경로, 보안에 취약하고 URL이 복잡하다.

Mapping path : context path는 그대로 쓰고 package와 servlet 이름을 간단한 닉네임으로 변경하는 것

**5.2 맵핑하는 방법 (1) : web.xml 파일을 이용한 맵핑**

Web.xml에서 아래 코드를 써준다.

Servlet을 먼저 선언하고 정확한 위치를 package를 포함해서 적어준다.

그 후 mapping tag에서 맵핑할 서블릿 이름과 맵핑할 url 주소를 적어준다.

<servlet>

<servlet-name>servletEx</servlet-name>

<servlet-class>com.servlet.ServletEx</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>servletEx</servlet-name>

<url-pattern>/SE</url-pattern>

</servlet-mapping>

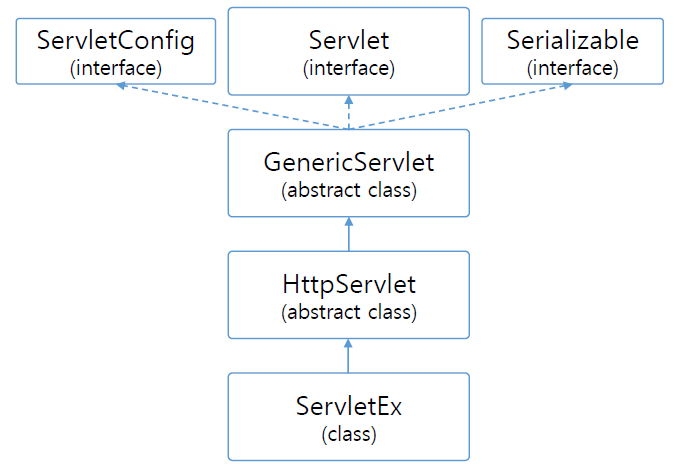
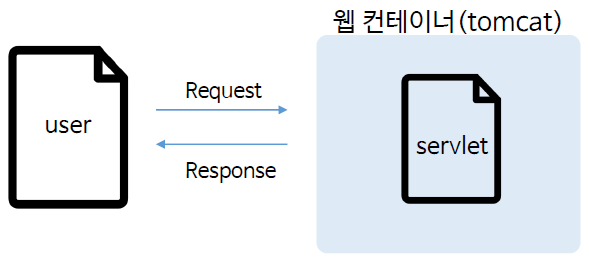
**5.2 맵핑하는 방법 (2) : 어노테이션 사용**

@WebServlet("/SE")

코드를 servlet 파일의 class 위에 적어준다.

Web.xml 사용하는 방법보다 어노테이션이 더 최신이고 잘 사용하지만 둘 다 알아두자.

**6.1 HttpServlet**



사용자가 요청하는 객체를 Request, 서버가 응답하는 객체를 Response 객체라고 한다.

HttpServlet : 추상 클래스로, 개발하는 servlet 코드에서 상속받아서 구현

HttpServlet은 GenericServlet을 상속받고 또 여러 인터페이스가 상속되지만, 개발자는 HttpServlet만 상속해서 기능을 구현하면 된다.

**6.2 HttpServletRequest**

사용자가 서버측에 요청을 보낼 때 사용하는 객체

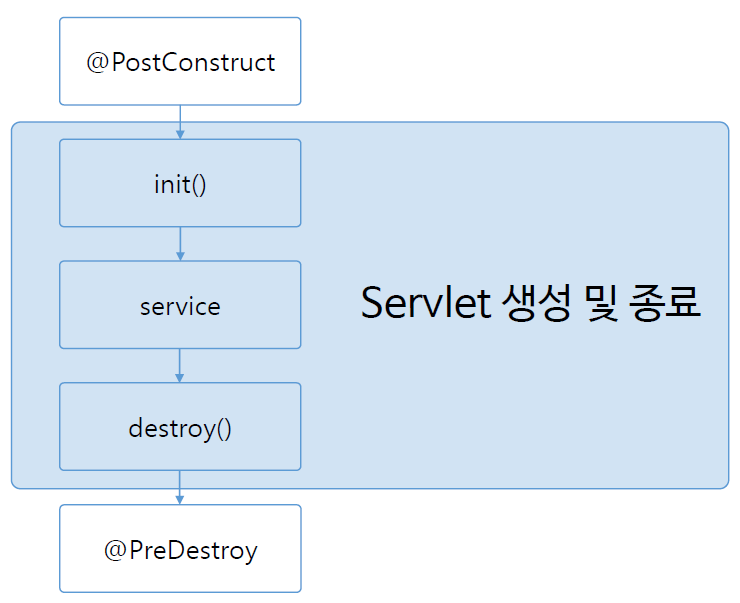
Request.addCookie, getSession, getAttribute 등의 기능이 있다.

**6.3 HttpServletResponse**

서버가 사용자측에 응답을 보낼 때 사용하는 객체

Response.addCookie, getStatus, sendRedirect 등이 있다.

**7.1 Servlet Life Cycle**



@PostConstruct : Servlet의 준비 단계

Init() : servlet의 생성 단계

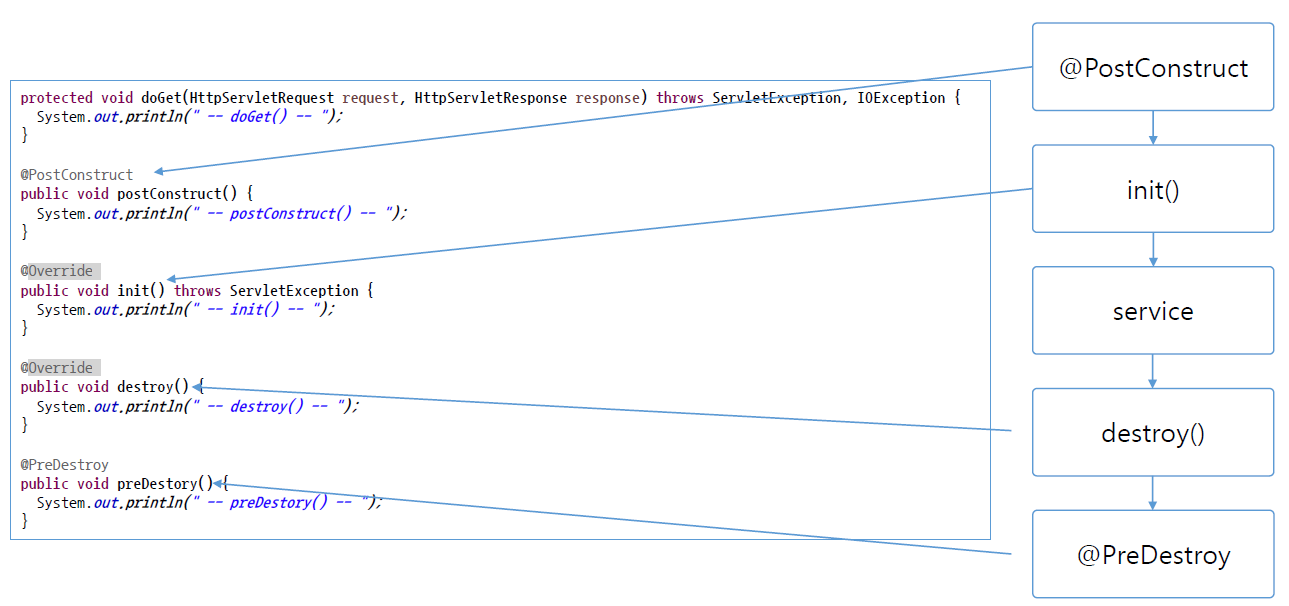
Service : servlet의 기능을 하는 것, 코드에서 doGet 부분이다.

Destroy() : servlet이 소멸되는 단계

@PreDestroy : Servlet의 종료 후 정리하는 단계

Container인 톰캣 서버가 알아서 적절한 시기에 각 단계를 실행시켜준다.

**7.2 생명주기 관련 메서드**



**8.1 form 태그**

form 태그 : 브라우저에서 서버로 데이터를 전송하는 태그. Request 객체를 통해 Data를 서버로 전송하게 된다.

**8.2 doGet 방식**

form태그에서 method=”get”일 때, 서버에서 doGet() 으로 데이터를 받는다.

default가 get방식이다.

form 태그에 입력한 데이터가 웹 브라우저 URL에 노출되어 보안에 약하다.

URL 길이가 한계가 있어서 많은 데이터 전송에 무리가 있다.

**8.3 doPost 방식**

form 태그에서 method=”post” 방식일 때, 서버에서 doPost()로 데이터를 받는다.

URL에 맵핑 정보만 보이기 때문에 보안에 더 강하다.