2024-02-02

P27 <HTTP를 비 연결지향형 프로토콜>

http 인터넷 서핑을 할 때 사용, 비연결(**비연결지향형**)

욫어 연결 통로를 만들고 내부적인 부분이 있다. 연결 요청 보내면 네이버 서버는 요청을 받아서

첫페이지 DB를 가지고 오고 응답을 가지고 접속을 끊어버린다

이미 다운받아버리는 것

브라우저 계속 띄워져 있는 것이지 서버 요청은 끊어진 상태이다

요청할 때만 서버 연결, 이후 데이터 불러오고 요청을 끊어버린다

(문의오면 처리하고 문 닫아버리는 것)

HTTP 핵심!

- 무상태 : 사용자가 누구인지 서버가 모르는 상태를 의미

특징 : 사용한 정보는 다음페이지에서 기억하지 못한다

웹브라우저 HTTP에 맞게 웹 서버에 요청(Request)을 보낸다, 헤더를 보내고 그다음에 본문을 보낸다. (사람 만날 때 인사하고 그다음에 이야기하는 것처럼)

요청을 보낼 때 여러가지 방식이 있다 웹브라우저에서 하는 방식 크게 두가지만 기억하면 된다.   
Get, Post 두가지만 기억하기!!!!

Get 웹 서버 가지고올 때 Post는 쓸 때,

Head는 쓸때 수정

Put 일부만 수정

Delete 삭제

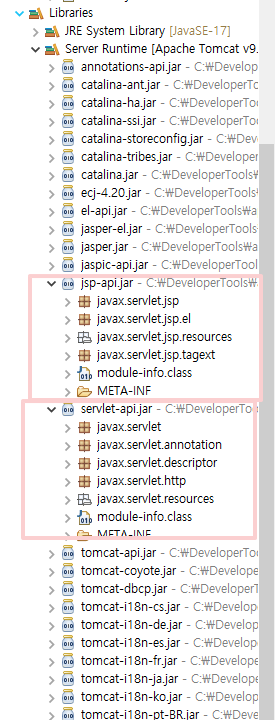
Post

헤더라는 부분이 먼저 가고 body는 나중에 간다 body는 크기가 정해져 있지 않는다 인코딩되어 갈 수 있는 상태 (post만 가능하다) 데이터 크기의 제한이 없고 사용자의 눈에 보이지 않기 때문에 보안성이 좋다

서블릿 두개가 있다

Javax.servlet

Javax.servlet.http

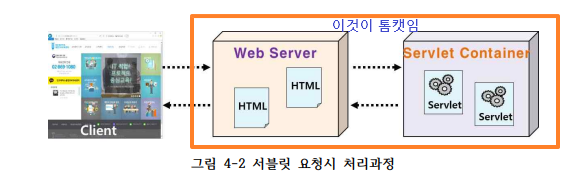


Libraries > JRE / Server / Web App Libraries

라이브러리 추가하면 Web App Libraries 라이브러리에 들어옴

HTTP 요청을 처리하기 위한 서블릿 클래스를 작성하려면 Httpservlet클래스를 상속받아 구현해야 한다 그러지 않으면 직접 처음부터 만들어야 한다

자식클래스에서 재정의 하지 않고 Get 방식의 요청을 처리하기 위해 doGet()메소드와 doPost()메소드를 자식 클래스에서 재정의 하면된다   
패키지의 2개의 객체를 파라미터로 넘겨받아 Http 요청과 응답에 대한 처리를 구현한다  
첫번째 인수(HTTpservletRequest) 두번째 인수(Httpservletresponse) 받도록 정의 되어 있다  
이 정해진 메소드로 구현하면됨

  
톰캣에 들어오면 웹서버가 먼저 받음 처리할 수 없으면 처리할 수 있게 servlet에 넘겨준다  
그 사용자에 대한 정보 넘겨주면 그 자료를 받아서 request와 response를 받아서 처리할 수있게 되어있다.

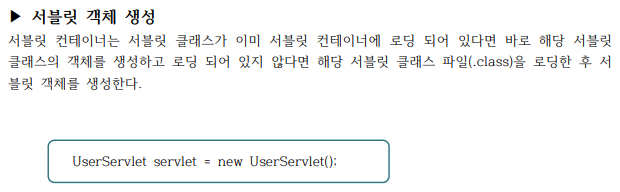
서블릿에 대한 요청이 들어오면 was 또는 컨테이너에게 처리를 의뢰한다

없어! 하면 없다고 넘겨주고 있으면 그 있는 값 넘겨준다  
web server은 받은 값을 Http에 넘겨주기만 하면 끝!

전달받은 요청 정보 분석한다음 응답함!

  
이렇게 객체 생성함, 사용자의 정보를 전달해줌

통로를 열어서 데이터 보내준 것 그래서 응답하게 어제 만들어준 것이다

  
서블릿도 톰캣이 자동으로 생성해줄 것

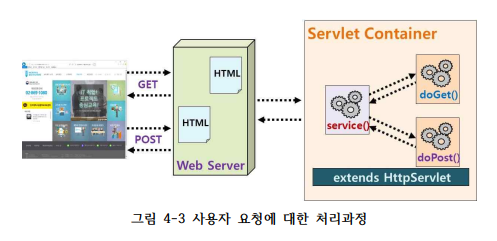
이런 역할들을 톰캣이 해주는 것이다.

서블렛 직접 호출하지 않고

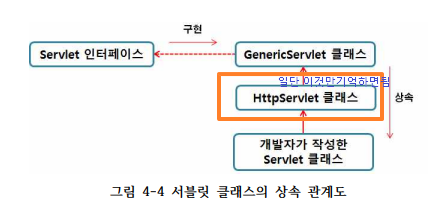
Service()을 통해서 호출한다 doGet이나 doPost를 통해 호출한다 그것을 오버라이드 (재정의)하는 것

Httpservlet 클래스를 상속받아 구현하게 된다

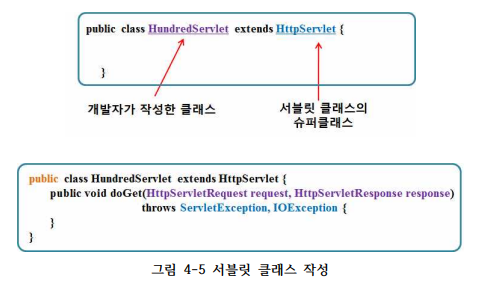
Httpservlet 클래스 service()메소드를 통해서 doget()메스도와 dopost()메소드를 간접적으로 호출

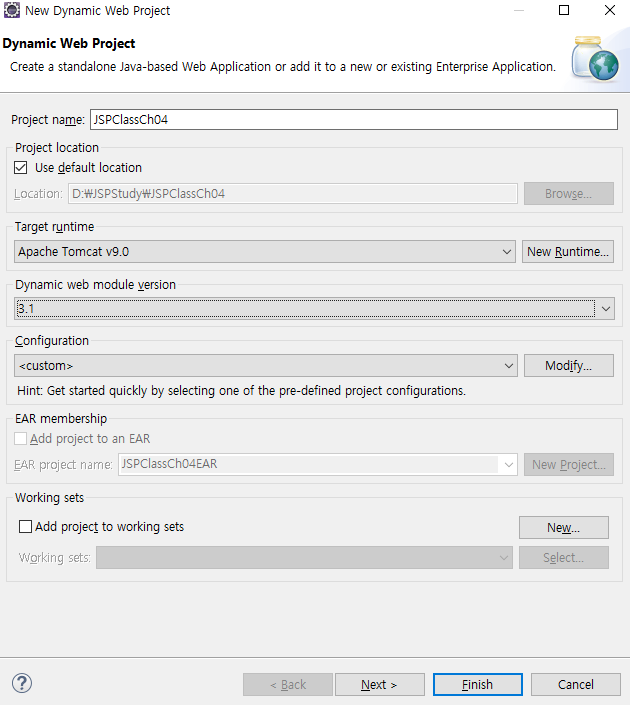
  
내부적으로 이렇게 동작을 함!!

4.3

  
Httpservlet을 상속받는다 !! 이것만 기억하면 됨 큰 핵심!

  
사용자 정보들이 여기 다 들어있다.

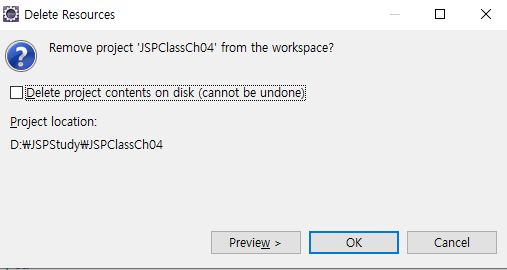
  
상속받은 우리의 클래스 그 안에서 상속받아서 get방식 사용하고 요청 응답 첫번째 두번째 인수로 선언하고 servletException IOException을 선언하면됨(이건 생략가능 다른 것은 생략X)



이제 생성해볼 것!!

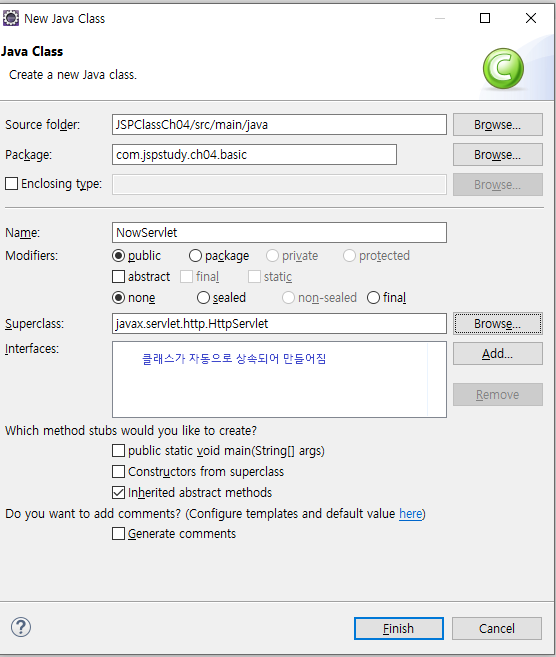
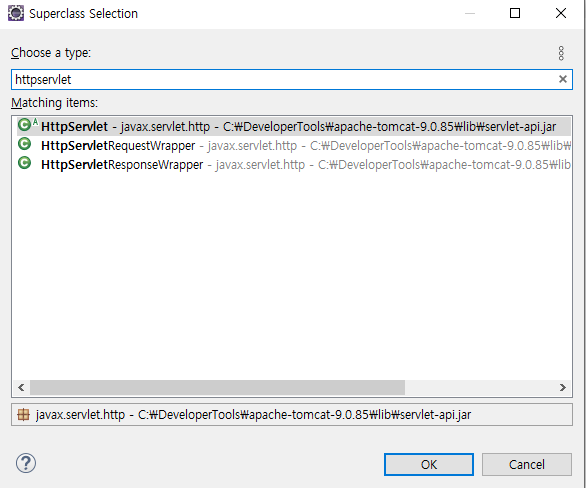
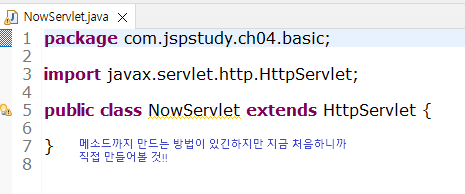
다이나믹 웹프로젝트 생성

패키지는 다 소문자로 사용

  
파일 삭제하려면 위와 같이 나온다 저 위에 체크박스 체크하지 않으면 파일이 폴더(디렉토리)안에 남아있어서 그 이름으로 프로젝트 생성할 수 없다 그래서 전체 삭제하려면 체크박스 체크하고 삭제하기!!!! (그리고 체크 안 했다면 디렉토리 가서 직접 삭제하면 됨)

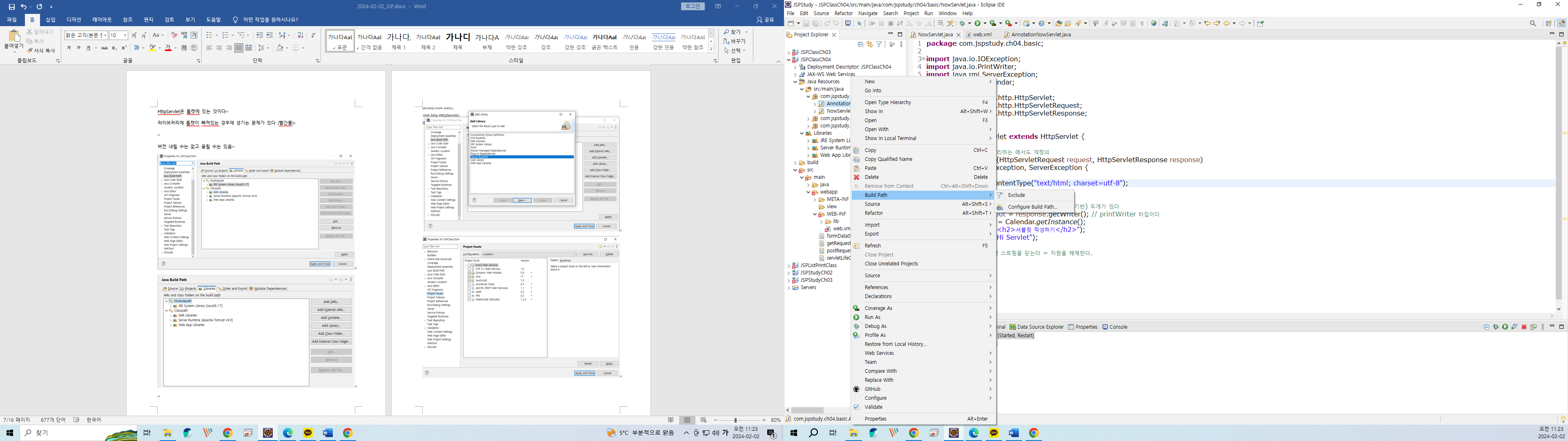
생성 패키지 생성 삭제하는 방법 설명함

Object에서 servlet 상속받아서 패키지 생성하는 방법

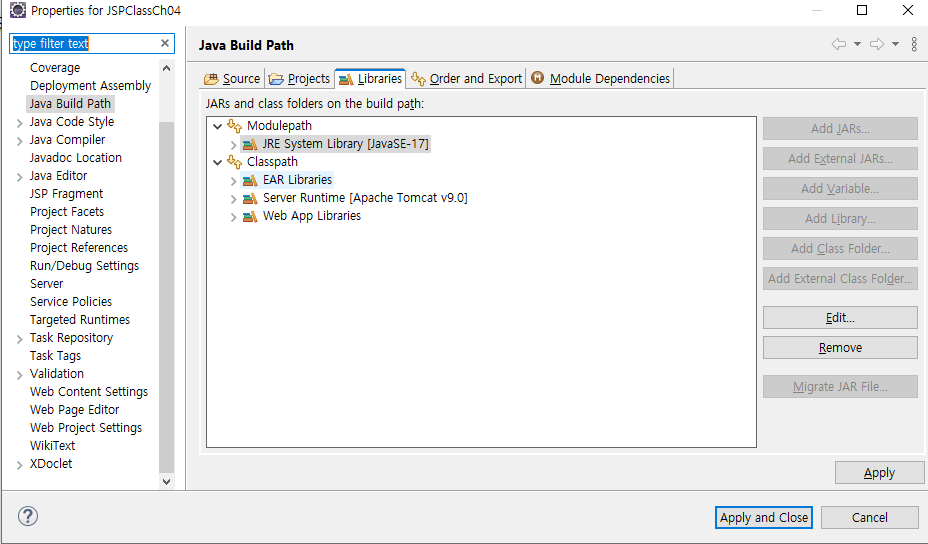
  


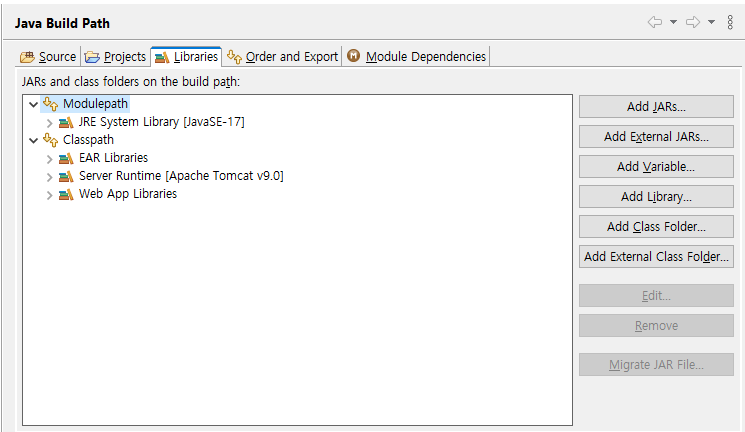
HttpServlet은 톰캣에 있는 것이다

라이브러리에 톰캣이 빠져있는 경우에 생기는 문제가 있다 (빨간줄)

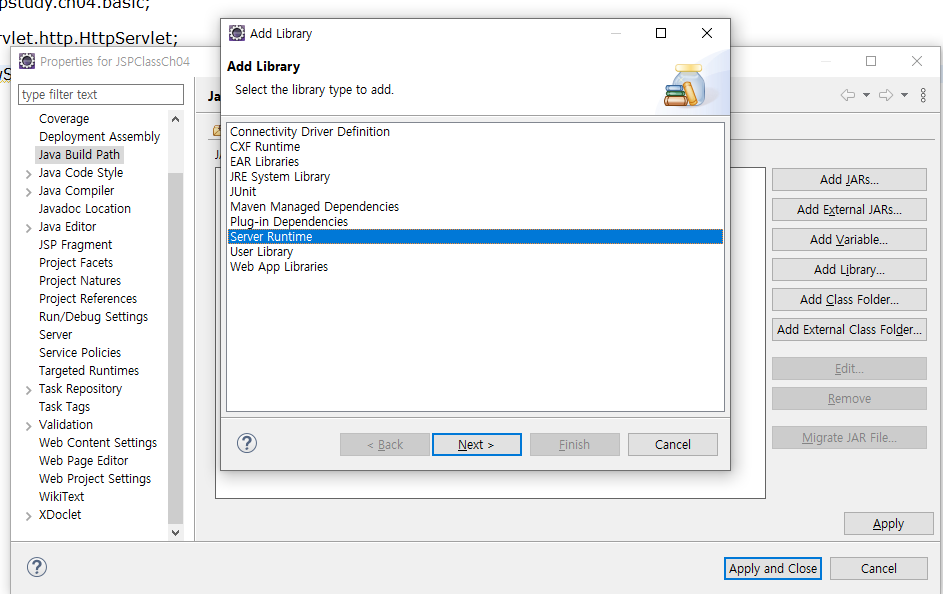
  
그때 project Explorer에서 오른쪽 마우스누르고 Build Path > configure Build Path 부분누르기

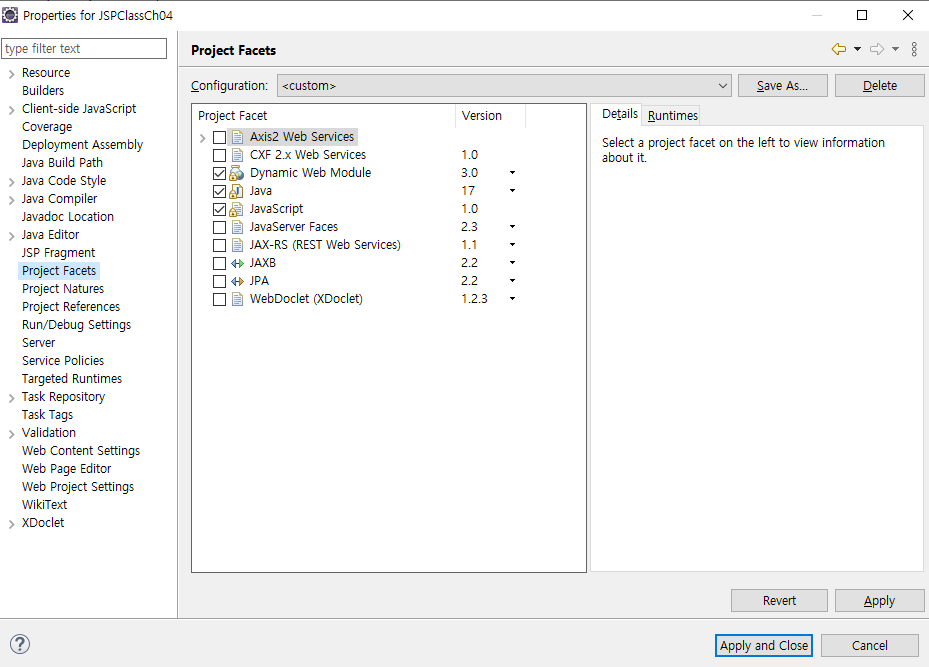
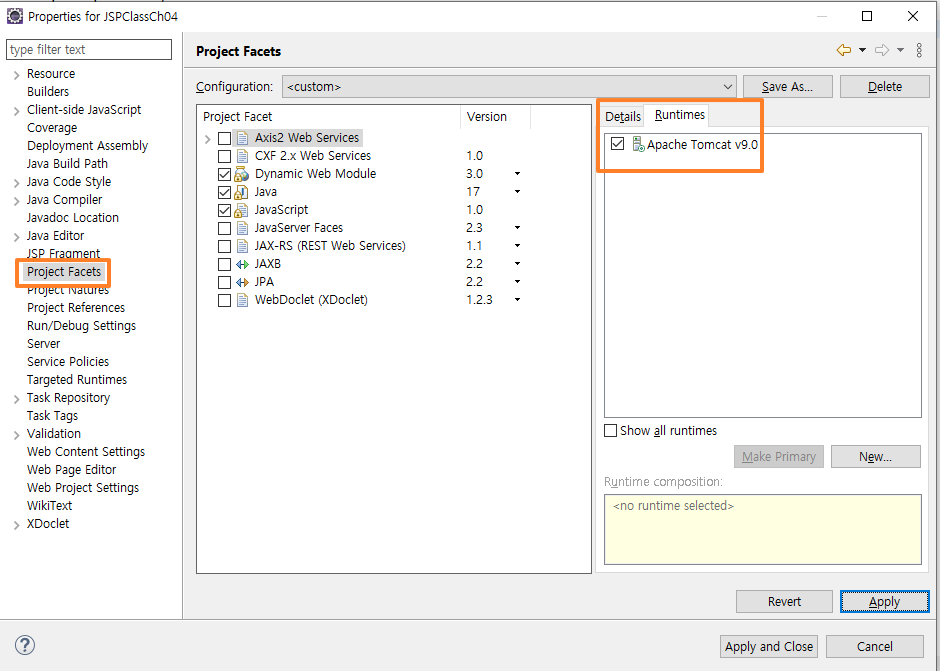
버전 내릴 수는 없고 올릴 수는 있음





Add Library누르고 톰캣 추가하기

  
server Runtime 추가하여 설치하기  
톰캣 다시 설치하는 것



위와 같이 톰캣 설치할 수도 있음

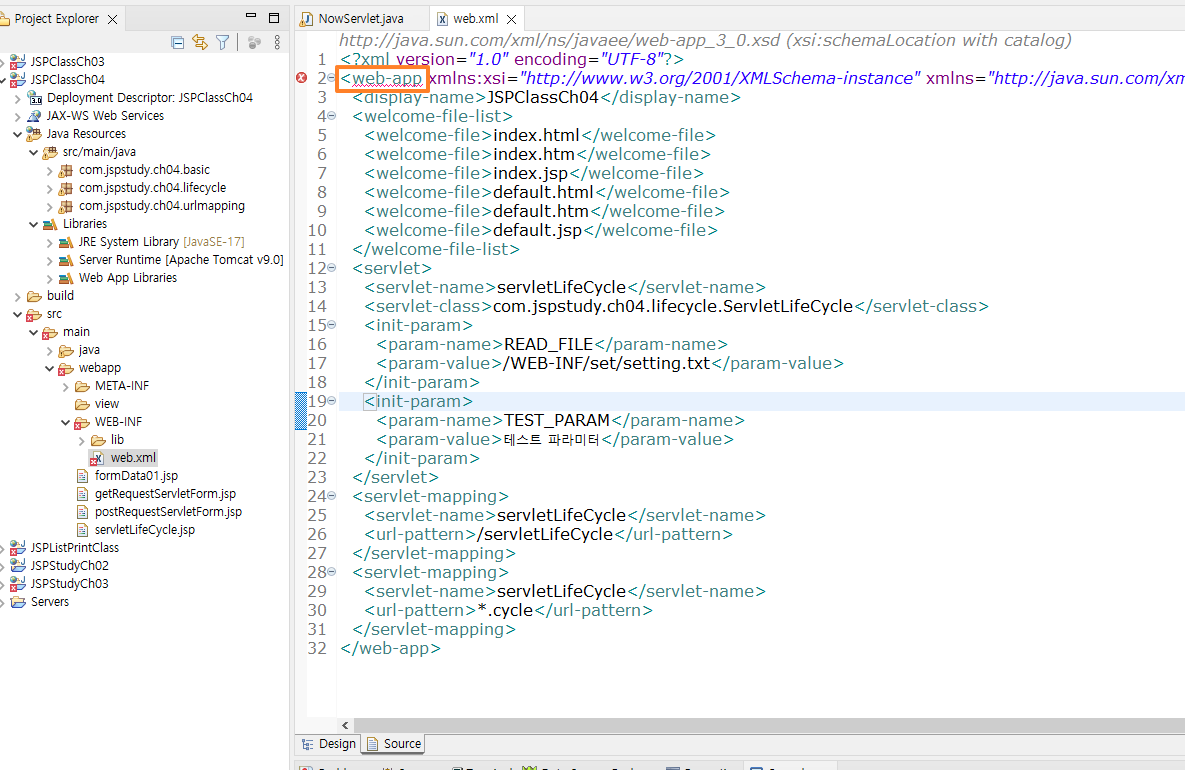
통로가 필요하다(요청과 응답이 지나갈)

요청에 대한객체 응답에대한객체

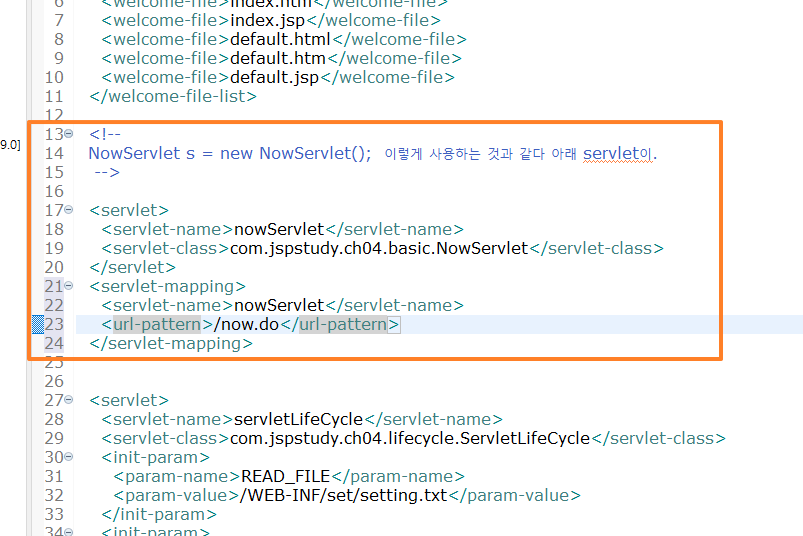
response응답을 이용해서 가지고 올 통로를 가지고 와야 한다 (String이 연결되어야 함)  
데이터 직접 써야할 때 response로 직접 받아오면 됨 (그것이 printwrite out 이다)

**public** **class** NowServlet **extends** HttpServlet

혼자 동작할 수 있는 아이가 아님, 톰캣한테 알려줘야 한다 등록이라는 것을 통해서  
webservice자체는 요청을 통해 실행된다. Get 방식일때는 얘를, post방식일 때는 얘를 실행하는 것

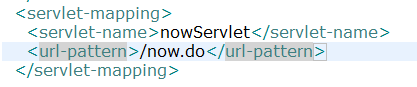


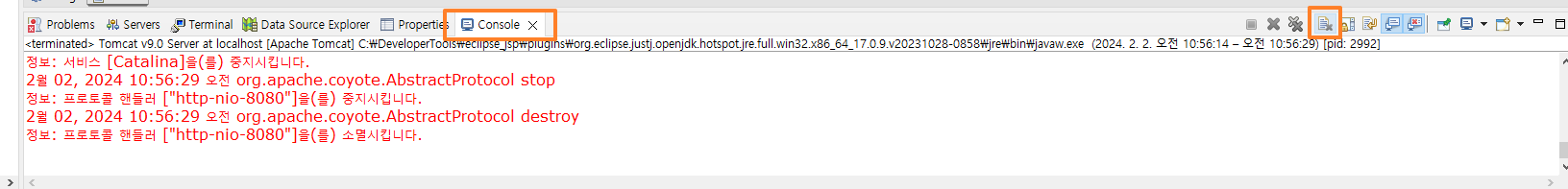
Xml도 html같이 노트태그 안에 다 안에 들어온다(소스 모드로 하기)

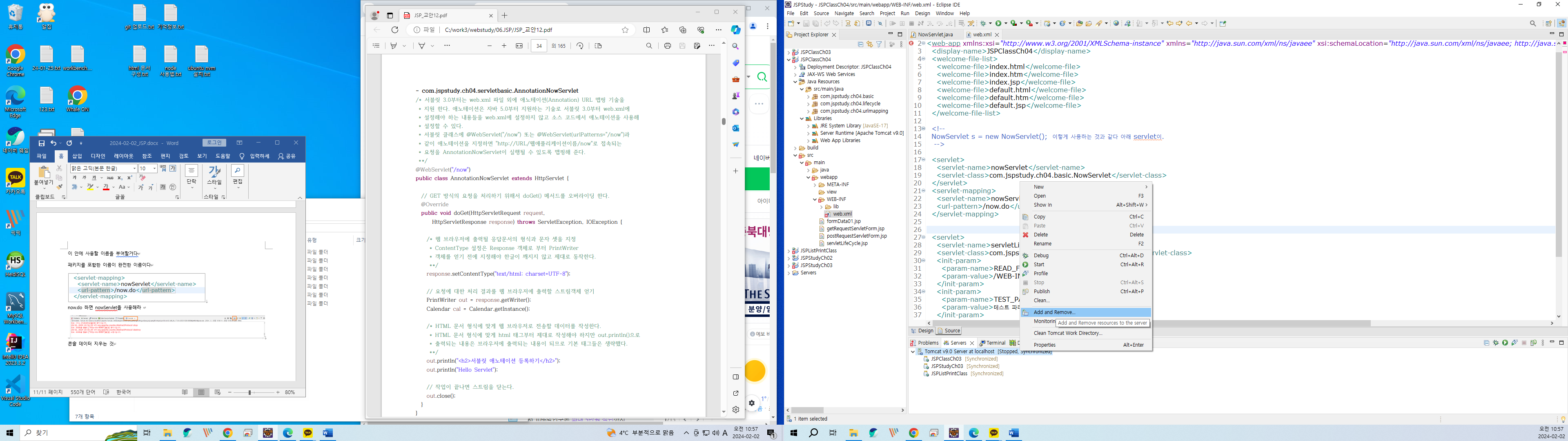


이 안에 사용할 이름을 부여할거다

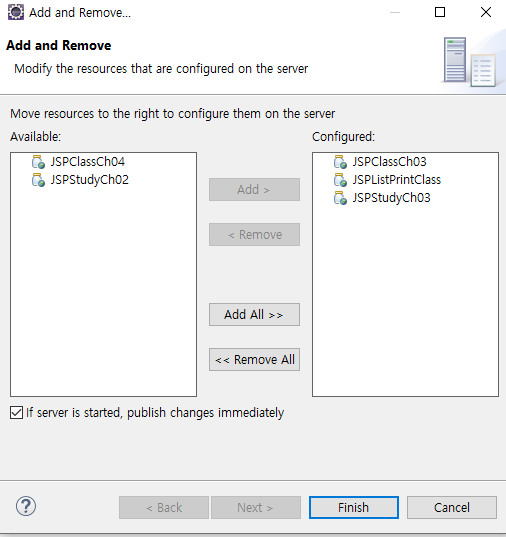
패키지를 포함한 이름이 완전한 이름이다

  
now.do 하면 nowServlet을 사용해라

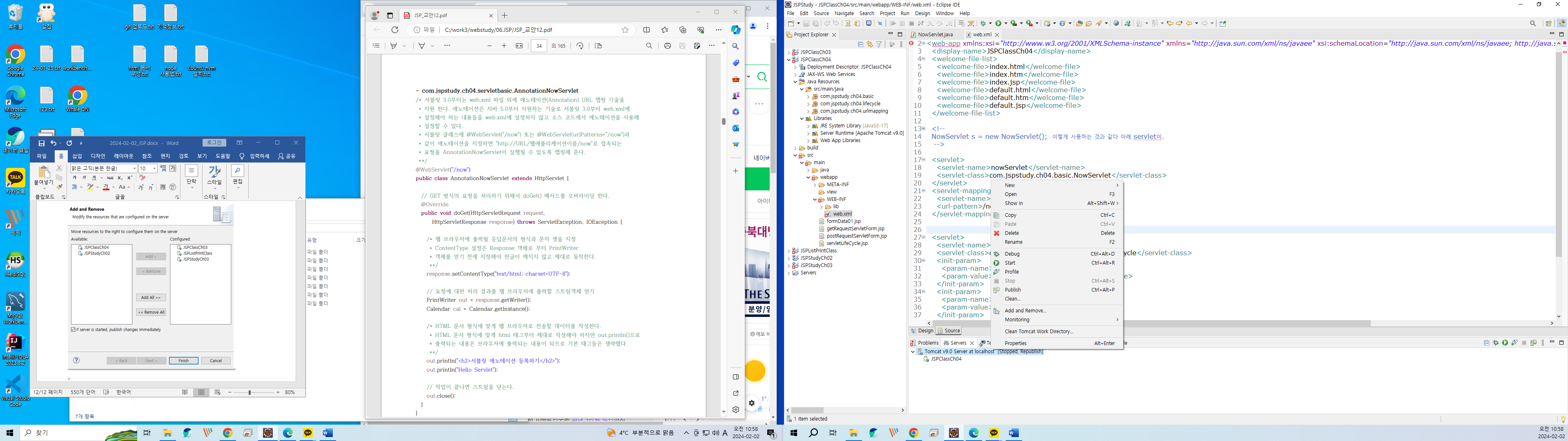
  
콘솔 데이터 지우는 것

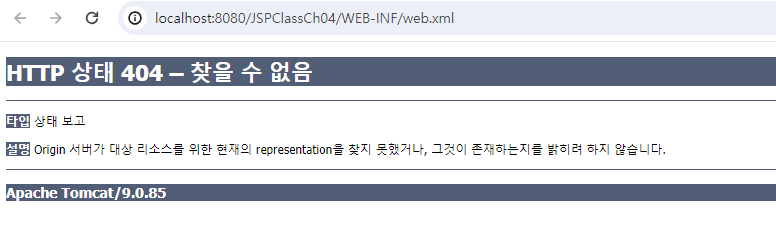


servers에서 길어지면 45초에 해결해야하는 것에 많은 정보들을 처리해야 하니까 위에 add and remove를 통해서 수정해준다.

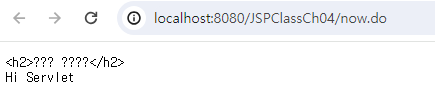


여기서 사용할 것만 하면 45초동안할 때 읽어야 하는 것을 줄일 수 있어서 무겁지 않게 실행할 수 있다

  
Clean Tomcat work Directory를 눌러서 깨끗하게 한번 정리해주고 다시 실행

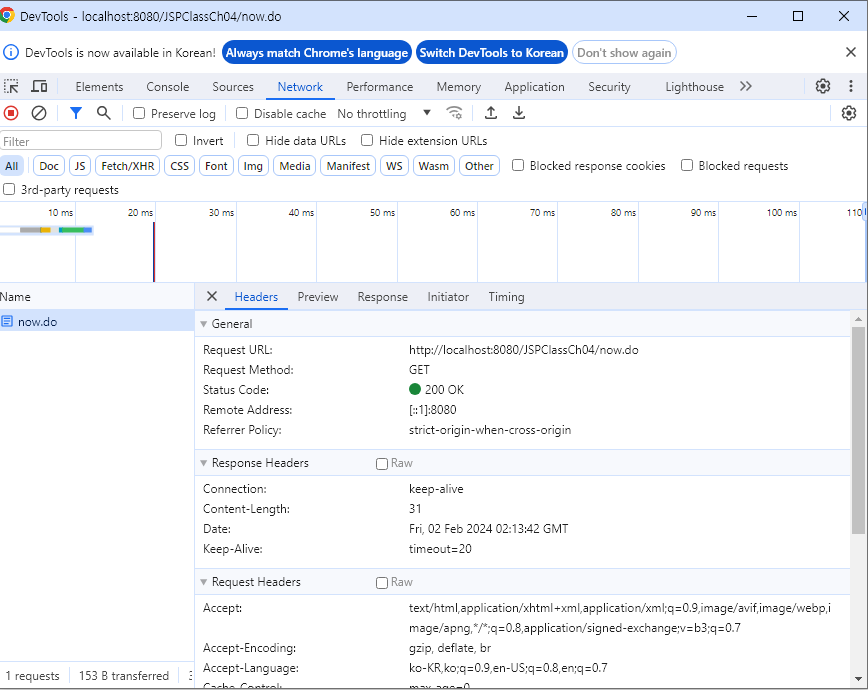
  
서버 오류가 나서 실행을 못했을 때  
주소가 실행될 때 이상하게 찍힌다

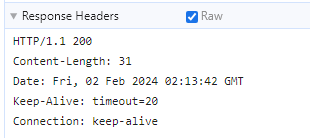
404는 없는데 들어가는 것 없다고 알려주는 것이다!!!  
위에 오류 난 것은 web.xml에서 실행해서 그런다!!!

  
NowServlet.java에서 실행해야 됨!!!  
위에는 한글 깨지는 것을 보여주려고 일부러 화면을 이렇게 띄운 것이다

오류메세지 해결하는 것 경쟁력이 된다

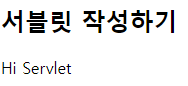
그 오류가 무엇 때문에 하는 건지 정확하게 알고 해결하는 것이 중요하다

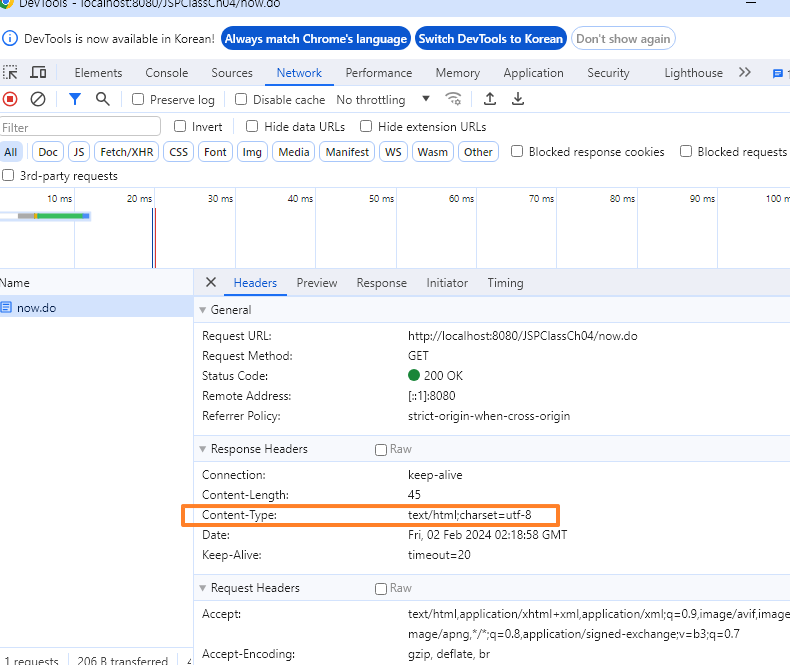
  
개발자모드 (network부분은 캡처해주는 것, 주소 엔터누르면 위와 같이 뜬다 (now.do 클릭하기)  
200은 정상적으로 처리된 것이다

  
처리한 날짜에 대해 알려준다 Raw를 체크하면 그에 대한 코드가 보여진다

이런데이터 들이 움직여진다

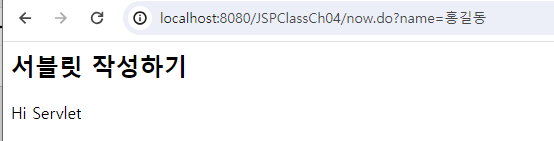
  
문자체계가 맞지 않으니까 오류가 발생하는 것 그래서 응답에 대해서 charset을 utf-8로 지정해주었다

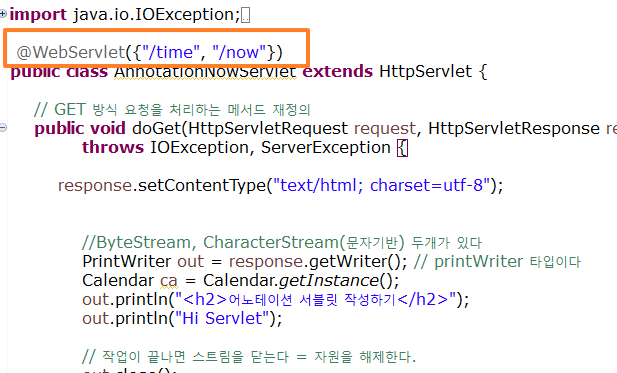
  
실행하면 위와 같이 잘 나온다

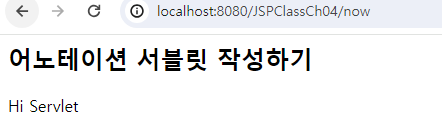
  
그리고 개발자모드 네트워크에서 문자가 UTF-8로 설정되어 있는 것을 확인할 수 있다

톰캣 부모

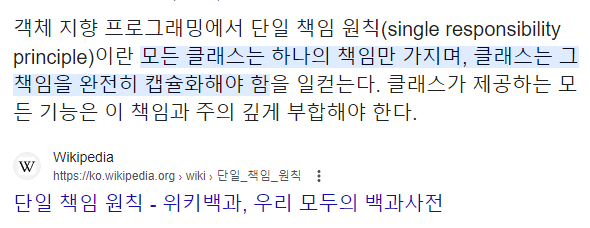
jspClassCh03 이런것이 자식

  
이것이 get 방식

  
annotationNowServlet 을 만들었는데  
거기에 webservlet (time과 now를 추가해주어서)

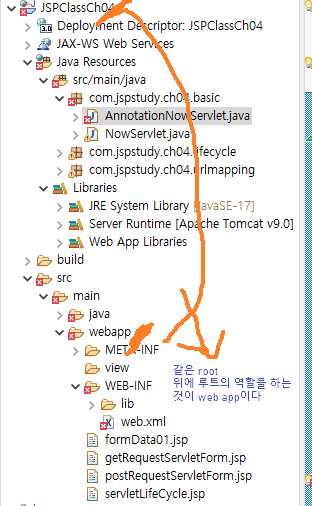
  
위에 /now와 /time할 때 화면에 출력되는 것을 확인할 수 있다.

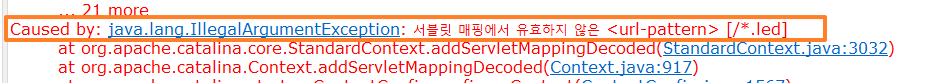
* 객체지향 5대원칙

첫번째 단일 책임 원칙이 있다.  


어떤 요청을 처리한 결과를 모델이라고 부른다

데이터 흐름을 처리하는 것을 해야 한다



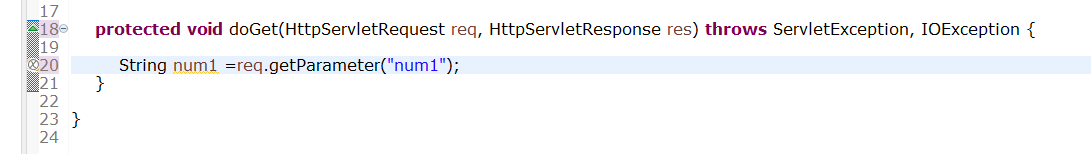
  
/\*.led 를 써주면 오류가 나온다

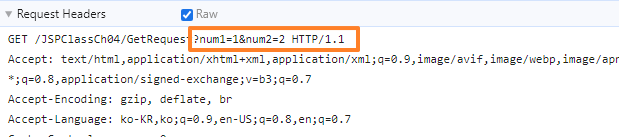
오후

서블릿으로 만드는 것!

--

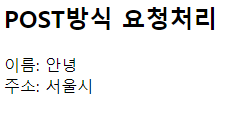
* GetRequestSevlet

  
node부분 res, req 하니까



Get 방식이다

* postRequestServletForm

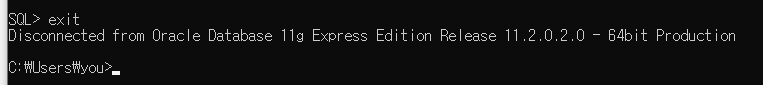


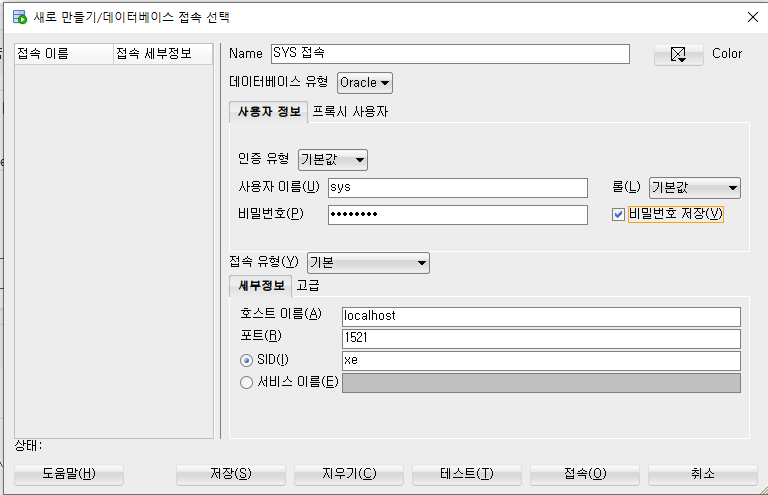
오라클

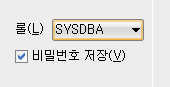
MYSQL을 많이 쓴다

* XE도구 설치

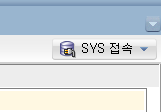
오라클 설치 <Oracle 11g XE 다운로드 및 설치 참고>

  
윈도우 프롬프트 오라클 빠져나온 것



  
관리자는 역할이 있다

JㅕJ

  
어떤상태로 접속했는지 보인다

Sys 스키마

테이블

뷰는 테이블을 통해서 조회된 가상테이블

시퀀스 : 오라클에서 알아야 함, mysql테이블 기본키 만들 때 Auto사용함 , 오라클은 오토가 없음 시퀀스 이용하여 순서를 지정해준다

계정관리는 몰라도 되고 지금은 11번임

  
여기서 어디가 제일 먼저 실행될까?

테이블 가지고 와야하니까 -> FROM

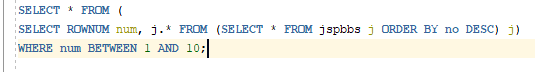
필터링 -> WHERE

검색 -> SELECT

ORDER BY (제일 마지막에 실행됨)

서브쿼리는 쿼리안에 들어있는 또 다른 쿼리를 말한다.  
예를 들면,

  
두번째에서 from, select, order 실행된다 -> 나중에 뒤죽박죽되어 버린다  
-> 또 서브쿼리를 집어넣는다



Order by를 원래 맨 나중에 해야 하는데 뒤죽박죽되니까 제일 처음 최신으로 실행시킨 것이다  
그러고 나서 num을 먼저 실행해야 하니까 select되게 하고  
그 다음에 where 쓰기!!

스프링가서 mysql 쓸것