

기존 웹은 runtime으로 구성된 W3C와 다음과 같은 URL로 개발자들
 설정 가능

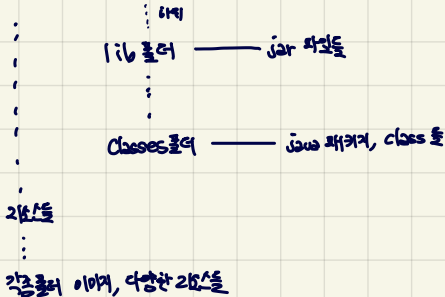
예)

http://localhost:8080/[프로젝트명]/[URL Mapping 값]
 프로토콜 IP 서버 포트

자바 웹어플리케이션을 만들어 플러그인을 제공해야 한다

// Servlet 3.0 부터는 애플리케이션에 의존할 수 있다

자바 웹어플리케이션 — WEB-INF 폴더 — web.xml 파일 (웹어플리케이션에 대한 정보를 가진)



- Servlet은 WAS에서 동작하는 JVM의 클래스이다
- Servlet은 HttpServlet 클래스를 상속 받아야 한다
- Servlet이나 JSP로부터 리소스 접근을 필요하면 클래스에 자원을 개입시켜 이 두가지를 조합해서 사용해야 한다

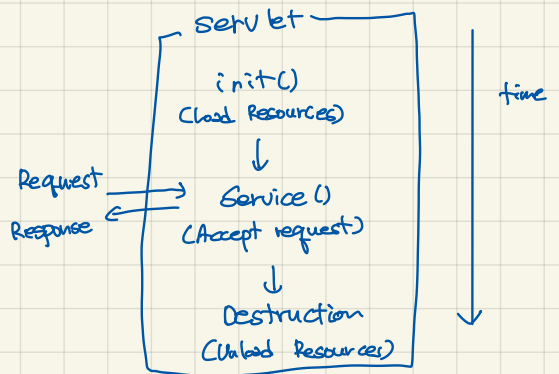
+ system.out은 콘솔에 출력하지 응답결과로 나타내지 않는다

Servlet 작성방법 { 3.0 ↑ web.xml 사용 X ⇒ 자바 어노테이션(annotation)로 사용
 3.0 ↓ web.xml 파일이 등록

* Lifecycle 메서드 (init, service, destroy) 메서드 override

Servlet을 실행시킴으로써

- 해당 URL로 클라이언트가 서버한테 요청
- 서버는 URL로 받아서 Lifecycle Servlet의 URL 매핑정보 찾아내고
- 해당 클래스의 메모리 존재여부를 확인
- 존재하지 않는다면 객체를 생성, 메모리에 올려 준다 // 생성자 호출
- init 호출, service 호출
- 브라우저 새로고침(새로운 요청) ⇒ service 메서드만 호출됨
 ⇒ 요청이 여러 번 들어오면 매번 생성하는 게 아니라 요청 객체의 메모리 존재여부를 판단해서 있다면, service 메서드만 호출한다.
- Servlet을 수정해서 더 이상 메모리 상의 Servlet 객체를 사용할 수 없게 되었을 때 destroy 메서드를 호출
 ⇒ 이후 새로고침시 생성자 부터 다시 호출된다.



WAS는 Servlet 요청을 받으면 해당 Servlet이 메모리에 있는지 확인

if (메모리에 없음) {
 • 해당 Servlet 클래스를 메모리에 올림
 • init() 메서드를 호출
 }
 • service() 메서드를 실행
 WAS가 종료되거나, 웹어플리케이션이 종료될 경우 destroy() 메서드가 실행됨

* service 메서드 (request, response) // 이이 관련이 되어 있음, 상속 override

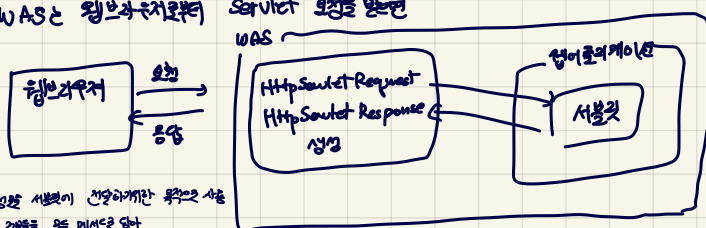
HttpServlet의 service 메서드는 애플리케이션 메서드 패턴으로 구현

- 클라이언트의 요청이 GET 요청인 doGet 메서드를,
- 클라이언트의 요청이 POST 요청인 doPost 메서드를 호출한다

+ request는 요청정보를 저장해서 갖고있는 객체

+ 객체 저장 프로그램에서 객체: 관련된 정보를 모아서 가지고 있음

WAS는 웹브라우저로부터 Servlet 요청을 받으면



// HttpServletRequest
 : http 프로토콜의 request 정보를 Servlet이 전달하기 위한 목적으로 사용
 ⇒ 요청할 때 주로 활용되는 모든 메서드를 담고

// HttpServletResponse
 : WAS는 어떤 클라이언트가 요청을 보냈는지 알 수 없고, 해당 클라이언트에 응답을 보내기 위한 HttpServletResponse 객체를 생성해서 응답으로 전달