

Servlet(서블릿)

클라이언트(브라우저)의 요청을 처리하고,
그 결과를 반환하는 Servlet 클래스의 구현 규칙을 지킨
자바 웹 프로그래밍 기술

자바를 사용해 웹을 만들기 위해 필요한 기술이다.

특징

- 클라이언트 요청에 대해 동적으로 작동하는 웹 어플리케이션 컴포넌트
- html을 사용하여 요청에 응답
- Java Thread를 이용해 동작
- MVC 패턴에서 Controller로 이용
- HTTP 프로토콜 서비스를 지원하는 javax.servlet.http.HttpServlet클래스를 상속 받는다.
- UDP보다 처리 속도가 느리다.
- HTML 변경 시 Servlet을 재컴파일해야 하는 단점이 있다.

일반적인 웹 서버는 정적인 콘텐츠를 제공한다.

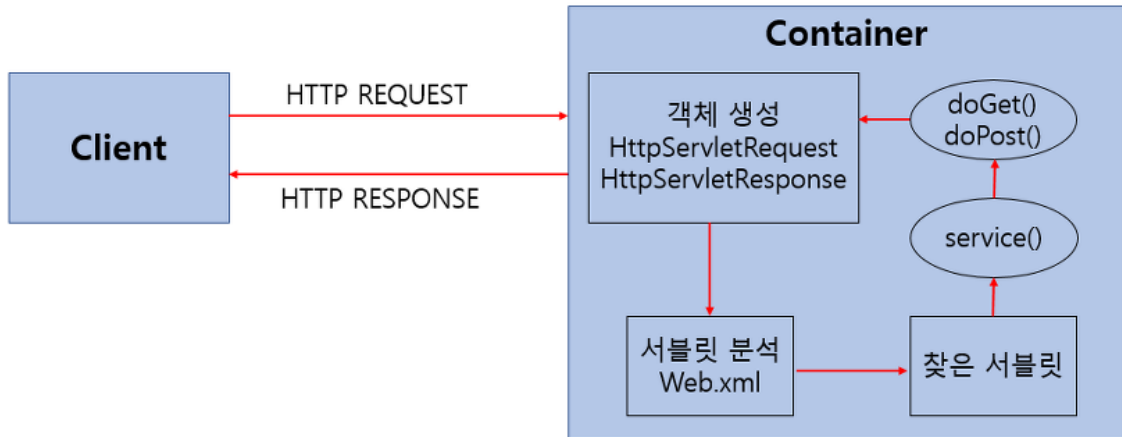
동적인 콘텐츠를 제공하기 위해서 웹 서버는 다른 곳의 도움을 받아야 한다.

동적인 콘텐츠로는 임의의 이미지만을 보여주는 페이지와 같이 사용자가 요청한 시점의 콘텐츠를 생성해 전달해주는 것을 의미한다.

여기서 웹 서버가 동적인 콘텐츠를 제공할 수 있도록 도와주는 것이 "서블릿"이며 동적인 콘텐츠를 생성하는 어플리케이션이 CGI이다.

CGI (Common Gateway Interface)

동작 방식



1. 클라이언트가 URL입력 → HTTP 요청 (Request)이 Servlet Container로 전송
2. 객체 생성
요청받은 Servlet Container는 HttpServletRequest, HttpServletResponse 객체 생성
3. 서블릿 분석
web.xml을 통해 클라이언트가 요청한 URL이 어느 서블릿에 대한 요청인지 찾는다.
4. 찾은 서블릿
찾은 서블릿에서 Service메서드를 호출 → 클라이언트의 GET,POST여부에 따라 doGet(), doPost()를 호출
5. 동적 페이지 생성
doGet(), doPost()메서드는 동적 페이지를 생성 → HttpServletResponse객체에 응답을 보냄
6. 객체 소멸
응답이 끝나면 HttpServletRequest, HttpServletResponse 두 객체를 소멸 시킴

서블릿 컨테이너 (Servlet Container)

| 서블릿을 관리해주는 컨테이너

서블릿을 만들기만 해서 스스로 작동하지 않는다.
이를 관리하는 컨테이너가 필요하며,
이것이 "서블릿 컨테이너"이다.

"서블릿이 메뉴얼이라면, 서블릿 컨테이너는 그 메뉴얼을 보고 수행하는 것."

서블릿 컨테이너는 클라이언트의 요청을 받고, 응답할 수 있게 웹서버와 소켓 통신을 하며, 대표적으로 "톰캣"이 있다.

- 톰캣은 실제로 웹 서버와 통신하며,
- JSP와 서블릿이 작동하는 환경을 제공해준다.

서블릿 컨테이너 역할

- 웹 서버와 통신 지원

▼ 서블릿과 웹 서버가 손쉽게 통신할 수 있게 해준다.

일반적으로 우리는 소켓을 만들고 listen, accept 등을 해야하지만 서블릿 컨테이너는 이러한 기능을 API로 제공하여 복잡한 과정을 생략할 수 있게 해준다.

그래서 개발자가 서블릿에 구현해야 할 비즈니스 로직에 대해서만 초점을 두게끔 도와준다.

- 서블릿 생명주기 관리

▼ 서블릿 컨테이너는 서블릿의 생성과 소멸을 관리한다.

서블릿 클래스(정의서)를 로딩하여 인스턴스화(객체 생성)하고,

초기화 메서드를 호출하고, 요청이 들어오면 적절한 서블릿 메서드를 호출한다.

또한 서블릿이 생명을 다 한 순간에는 적절하게 Garbage Collection을 진행하여 편의를 제공한다.

- Multi Thread 지원 및 관리

▼ Thread 생명주기 관리

서블릿 컨테이너는 요청이 올 때마다 새로운 자바 Thread를 하나 생성하는데, HTTP 서비스 메서드를 실행하고 나면, Thread는 자동으로 소멸한다.

원래는 Thread를 관리해야 하지만 서버가 Multi Thread를 생성 및 운영해주니 Thread의 안정성에 대해서 걱정하지 않아도 된다.

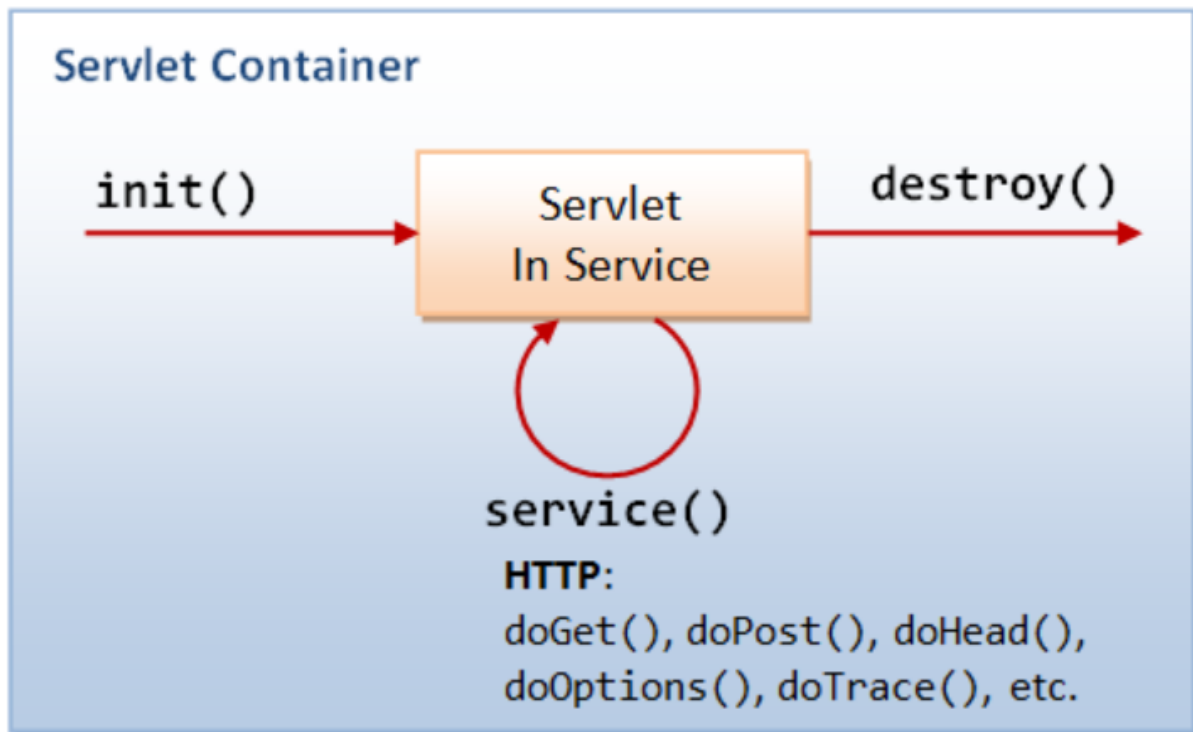
- 선언적인 보안 관리

▼ XML에서 보안 관리

서블릿 컨테이너를 사용하면 개발자는 보안에 관련된 내용을 서블릿 또는 자바 클래스에 구현해놓지 않아도 된다.

일반적으로 보안관리 XML배포 서술자에 기록하므로, 보안에 대해 수정할 일이 생겨도 자바 소스 코드를 수정하여 다시 컴파일 하지 않아도 보안관리가 가능하다.

서블릿 생명주기



1. 클라이언트의 요청 → 컨테이너는 해당 서블릿이 메모리에 있는지 확인
 - a. 없을 경우 init()메서드를 호출하여 적재한다.
 - b. 있다면 해당 서블릿 호출하여 적재
2. init()이 호출된 후 클라이언트 요청에 따른 Service()메서드를 통해 요청에 대한 응답이 doGet()과 doPost()로 분기된다. 이때, 서블릿 컨테이너가 클라이언트의 요청이 오면 가장 먼저 처리하는 과정으로 생성된 HttpServletRequest, HttpServletResponse에 의해 request와 response 객체가 제공된다.
3. 컨테이너가 서블릿에 종료 요청을 하면 destroy()메서드가 호출되는데 마찬가지로 한번만 실행되며, 종료시에 처리해야하는 작업들은 destroy()메서드를 오버라이딩하여 구현하면 된다.