Servlet Life Cycle

라이프 사이클: 서블릿은 생명 주기를 통해서 작동하는데 서블릿 컨테이너라는 곳에 존재한다.
init(): 요청이 왔을 때 최초 구동이 되며 로드 되는 단계
Servlet In Service: 각종 요청이 들어오면 작동되며 응답하게 된다. get, post, head 등의 요청
destroy(): 코드 변경, 자원 부족 등의 상황일 때 destory 호출.

Servlet Container(=Web container)의 특징
- Servlet 을 관리하고 실행하는 component 이다.
- 자체적으로 JVM 과 JRE 를 포함한다.
- 웹서버의 URL 요청을 받아 매핑되는 servlet 을 실행한다.
- Servlet 의 전반적인 Life Cycle 을 관리한다.
- 그 예로 Apache Tomcat, BEA WebLogic 등이 있다.
- JSP 도 내부적으로 servlet 으로 변환되어 실행된다.(컨테이너에서 변환)

Container 와 Servlet 동작 방식

1. Client 의 URL 요청
2. 웹 서버가 Request, Response 객체를 생성
3. Servlet 인스턴스와 Thread 생성
4. 스레드가 service() 메소드 호출 (여기서 service() 메소드는 각종 요청에 응답하게 하는 메소드)
5. doGet() 또는 doPost() 에서 각 response, request 객체를 인자로 호출
6. 메소드 호출 뒤, 웹 페이지 생성하면 웹 컨테이너가 응답(Response) 형태로 바꾸어 웹 서버에 전송

Client <-> Web Server <-> Container 로 동작
클라이언트에서 요청을 하면 웹 서버가 받아서 컨테이너로 보낸다. 컨테이너 안에는 각종 Url 들이 있는데 servlet 과 각각 매칭해서 메모리에 올리고 스레드

가동 후 get, post 방식에 맞춰서 doGet(), doPost() 메소드 호출 후 응답한 것을 다시 거꾸로 보내서 클라이언트는 어떤 요청에 대해서 결과를 띄운다.

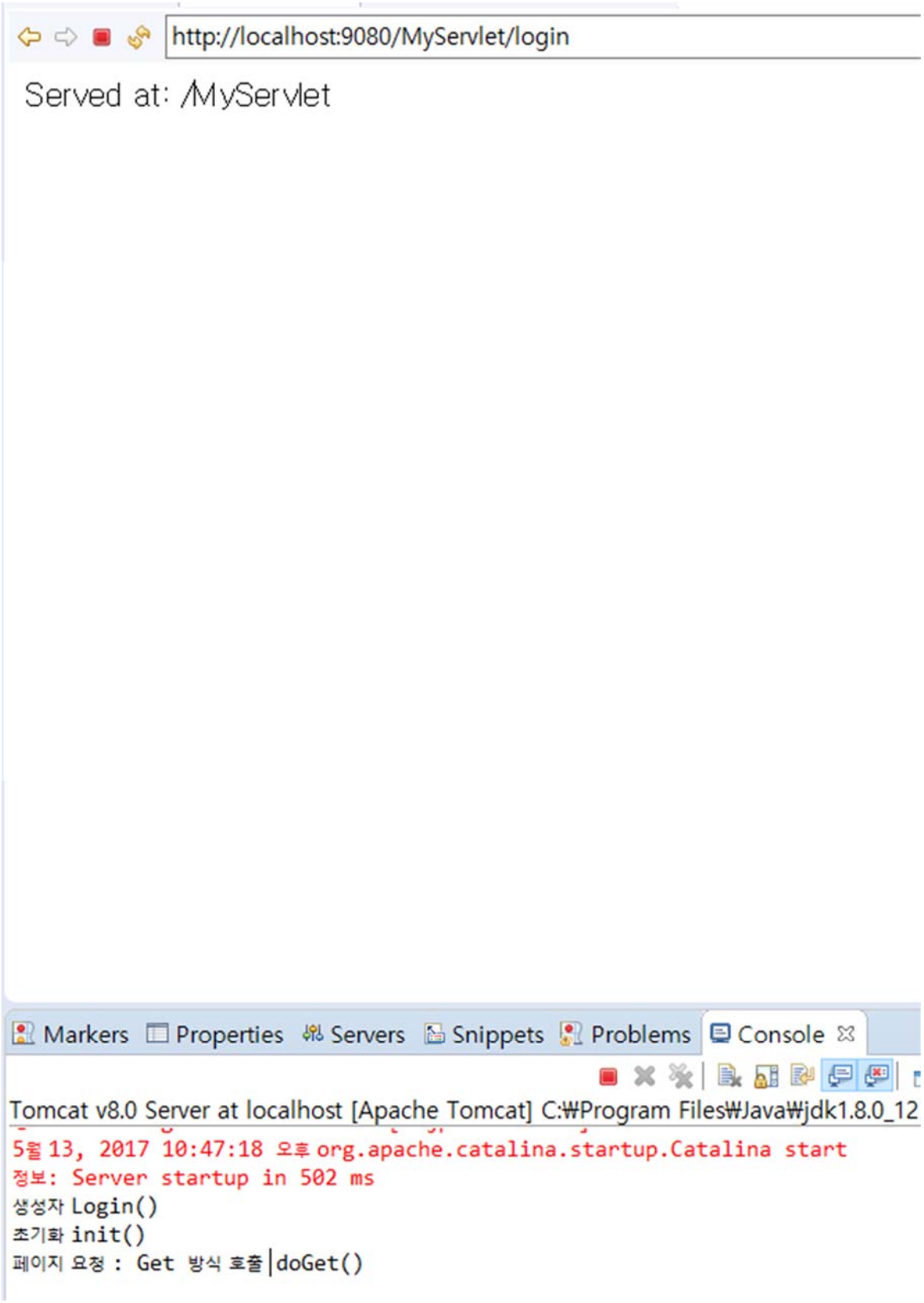
주요 메소드 비교

메소드	개요
init() - 시작	서블릿이 메모리에 로드 될 때 1 수정으로 인해 리로드 되면 다시
destory() - 끝	서블릿이 메모리에서 해제되면 호 코드가 수정되면 다시 호출
doGet()	Get 방식으로 서블릿을 요청하면
doPost()	Post 방식으로 서블릿을 요청하면
service()	모든 요청은 service() 를 통해서 d

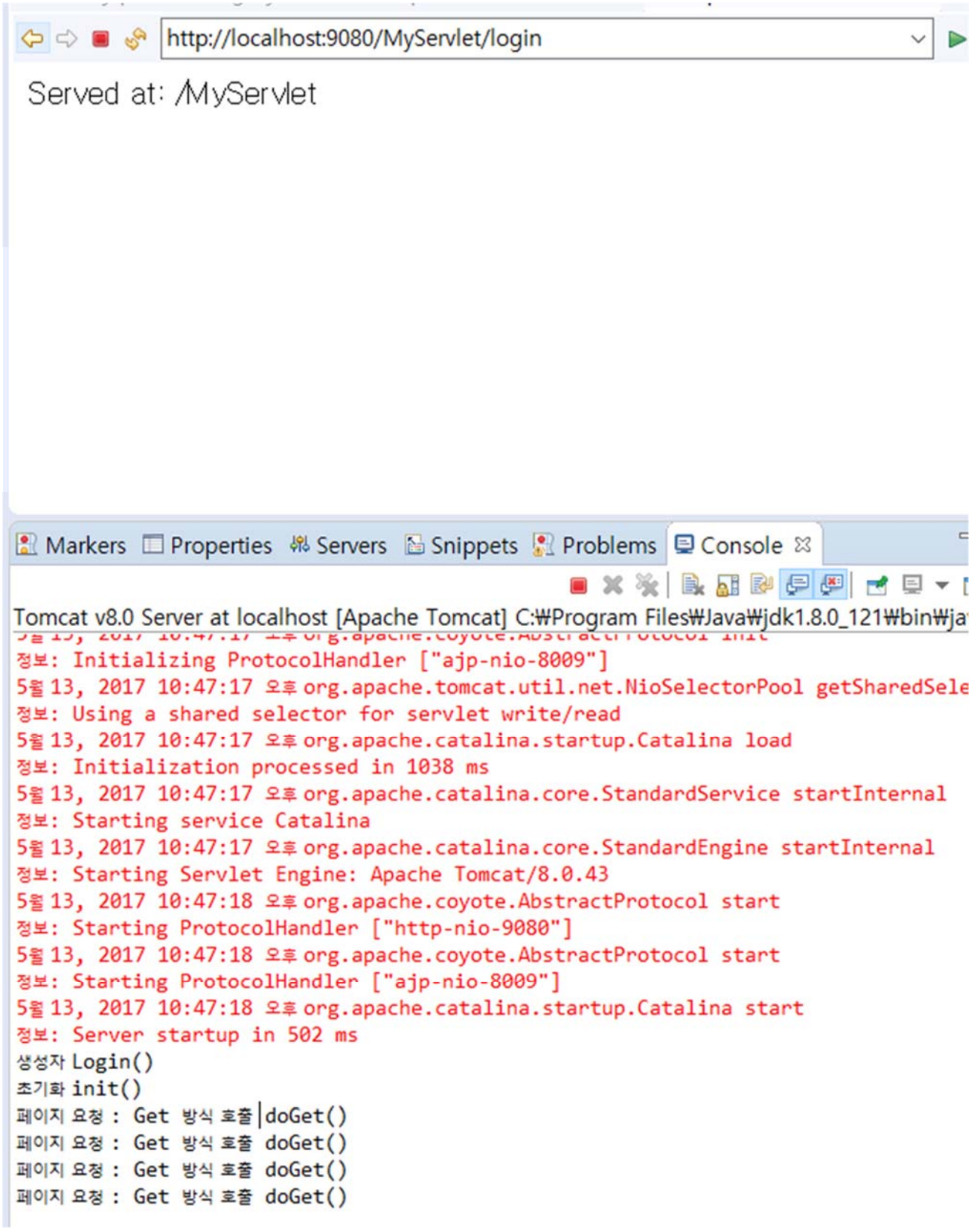
이제 예제 코드를 통해서 Life Cycle 을 확인해보자!

```java
@WebServlet(name = "login", description = "로그인하는 서블릿", urlPatterns = { "/login" })
public class Login extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public Login() {
        super();
        System.out.println("생성자 Login()");
    }

    public void init(ServletConfig config) throws ServletException {
        System.out.println("초기화 init()");
    }

    public void destroy() {
        System.out.println("소멸자 destroy()");
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("페이지 요청 : Get 방식 호출 doGet()");
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("페이지 요청 : Post 방식 호출 doPost()");
        doGet(request, response);
    }

}
```

[사진 1: 라이프 사이클 예제 소스 코드]

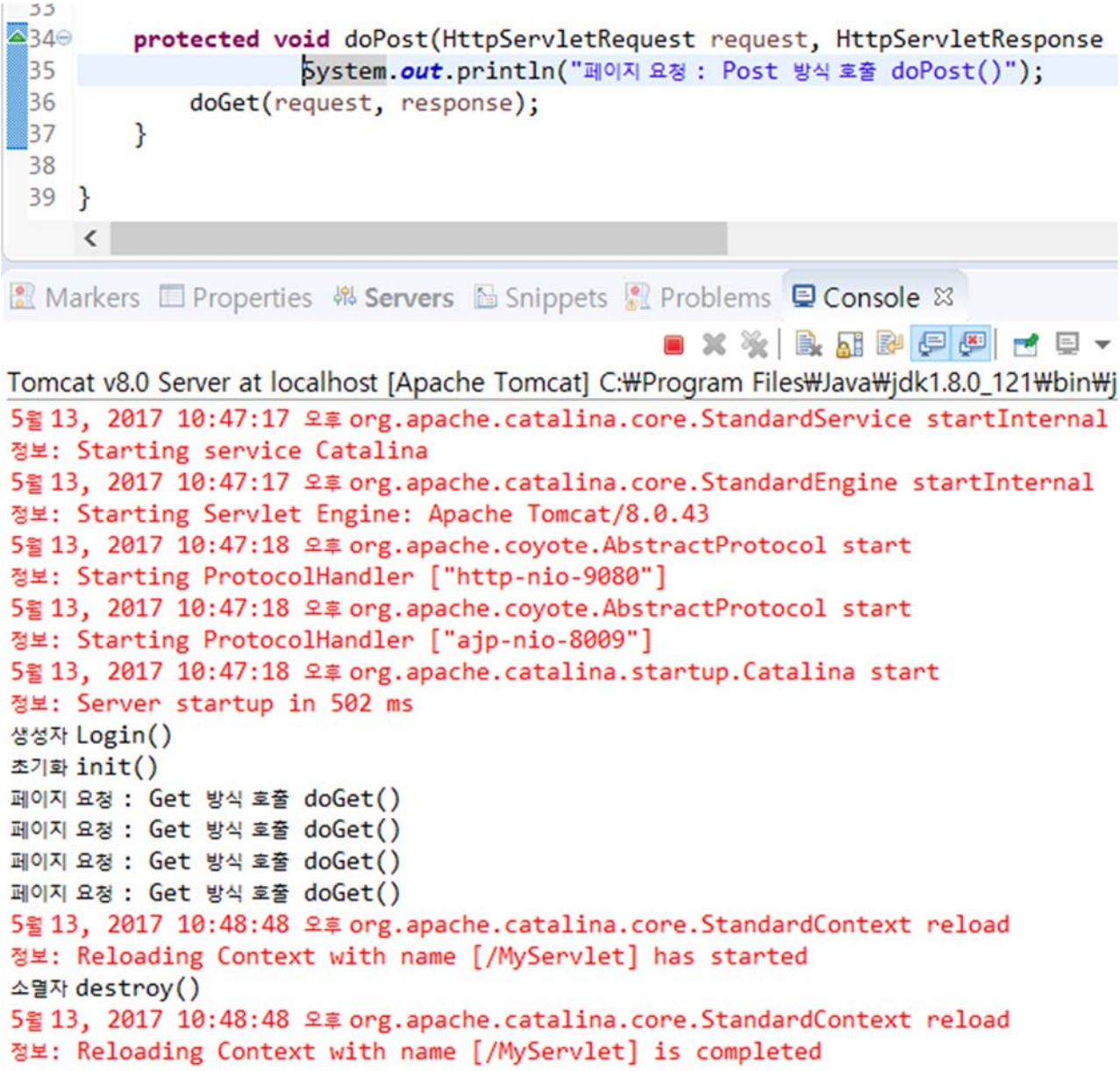


[사진 2: URL 접속 했을 시 콘솔 창]

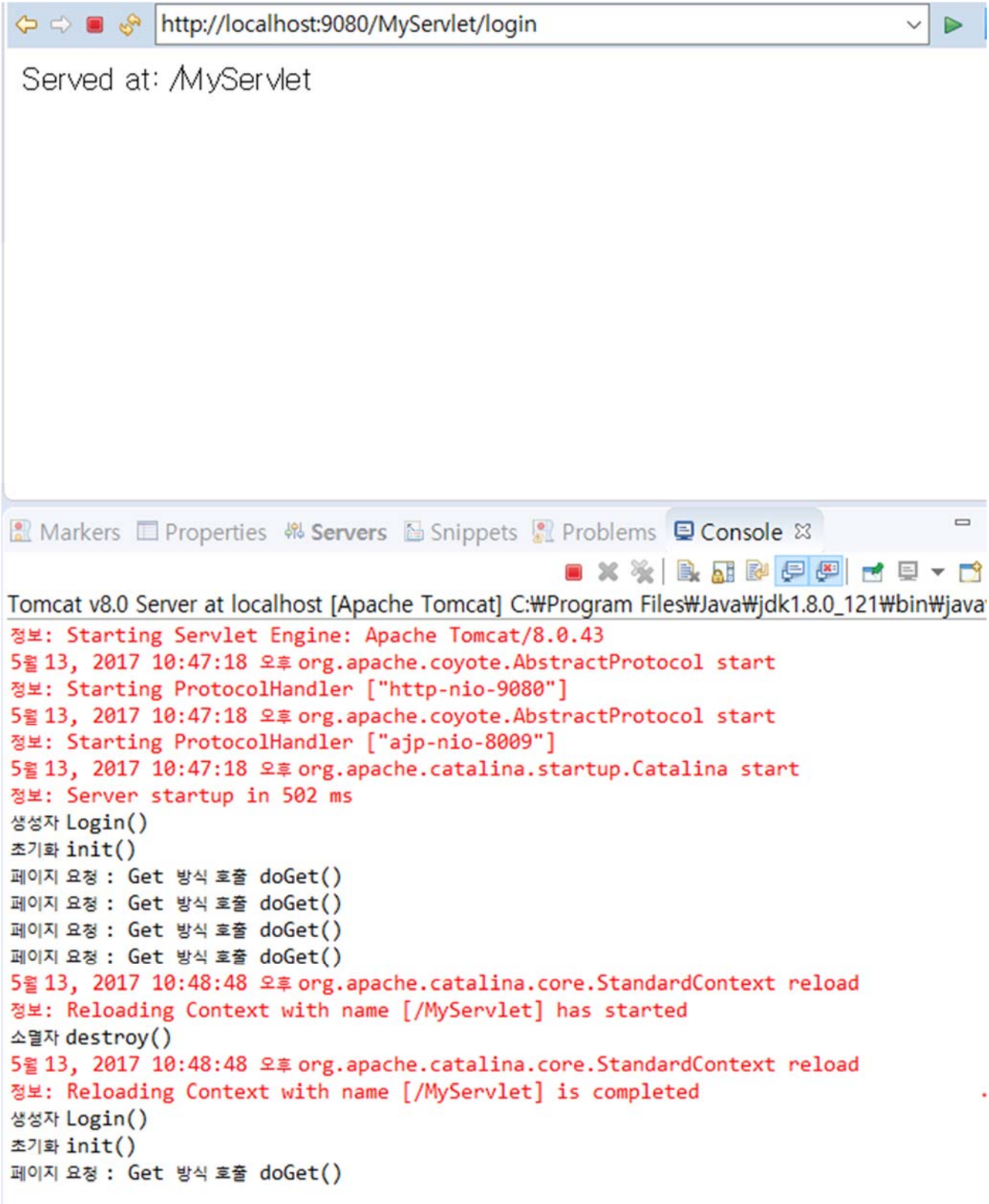


]

[사진 3: 새로 고침 했을 시 콘솔 창]



[사진 4: 서블릿 코드 변경해 리로드 될 시]



[사진 5: 새로 고침 했을 시]

다음과 같이 콘솔 창이 출력된다. 이것이 바로 서블릿 라이프 사이클이다.