

Servlet

Servlet ?

Spring을 배우며 많이 등장하는 Servlet

사실 정확한 의미는 모른채 ... 배워왔다...

그럼, 도대체 뭘까 ? 🙄

Servlet ?

A servlet is a Java programming language class that is used to **extend the capabilities of servers that host applications accessed by means of a request-response programming model.**

<https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html>

Java Servlets are server-side Java program modules that **process and answer client requests** and implement the servlet interface. It helps in enhancing Web server functionality with minimal overhead, maintenance and support.

They can receive and respond to requests made by the client. **Request and response objects of the servlet offer a convenient way to handle HTTP requests and send text data back to the client.**

<https://www.techopedia.com/definition/12874/java-servlet>

Servlets are the Java programs that runs on the Java-enabled web server or application server. They are used **to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server.**

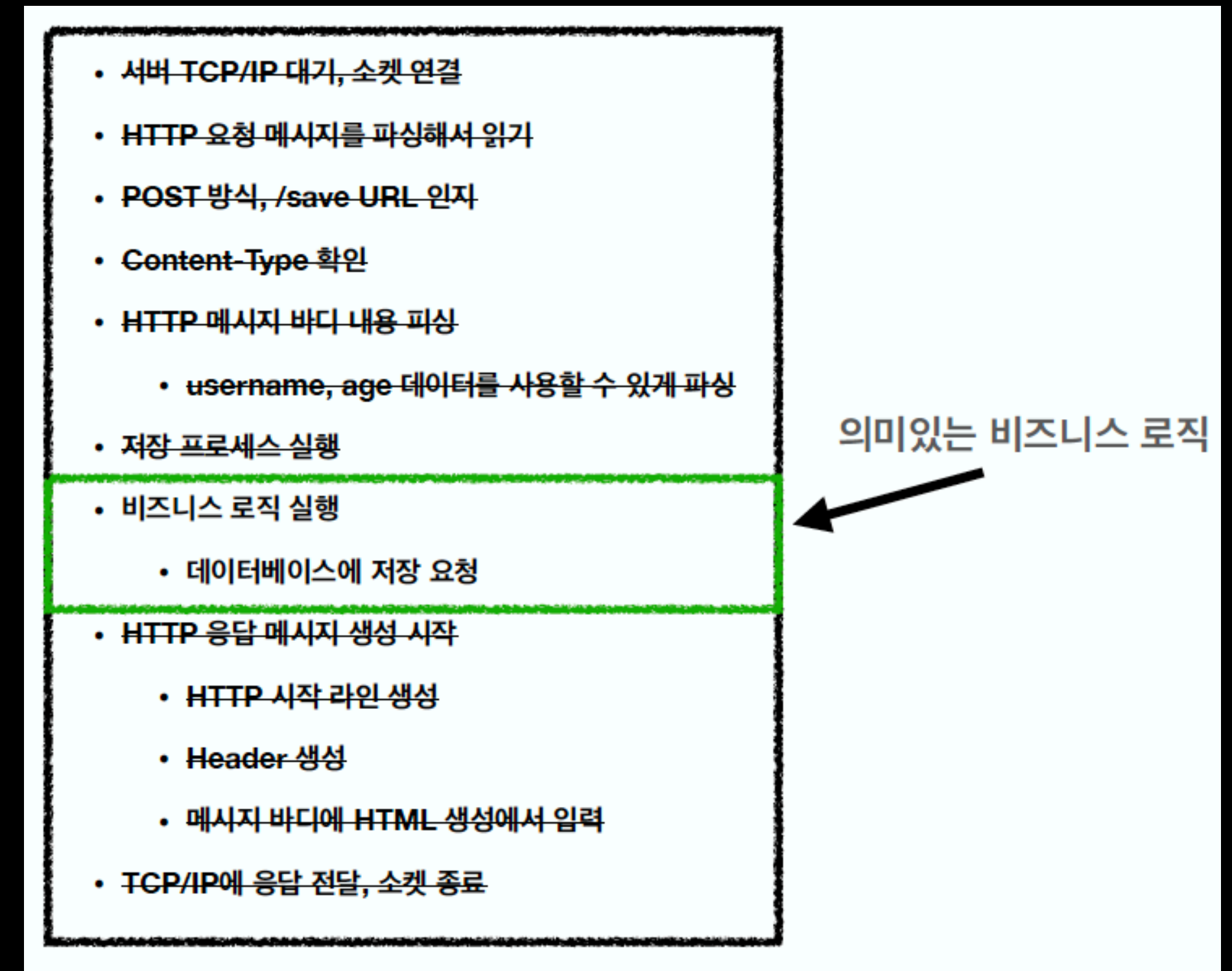
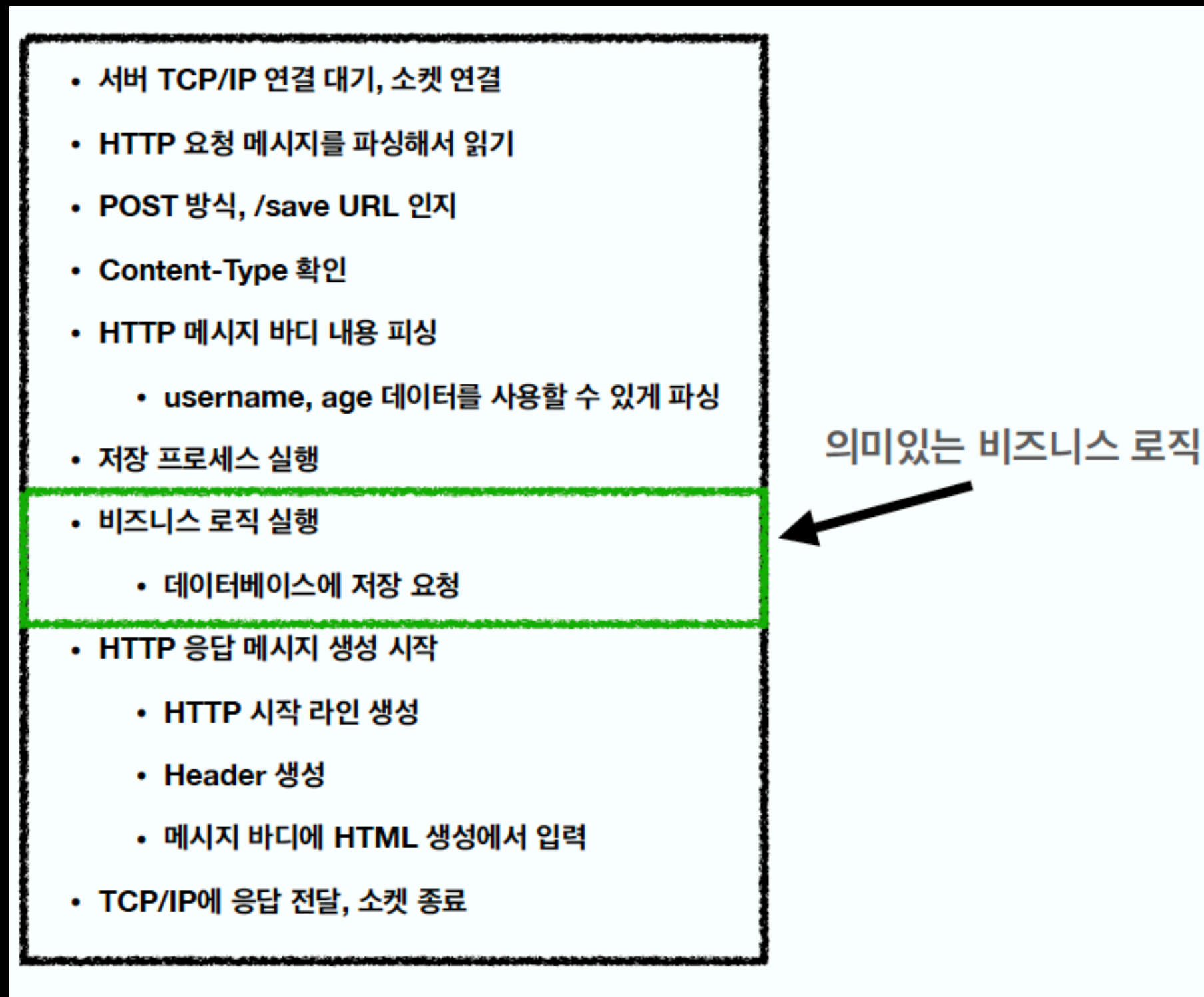
<https://www.geeksforgeeks.org/introduction-java-servlets/>

Servlet ?

즉, Servlet이란 Java를 사용해 웹을 개발하기 위한 기술로써,
Client가 한 요청에 따라 그에 대한 결과를 반환하는 Java Program

Servlet은 개발자가 비즈니스 로직 개발에 집중할 수 있도록 도와준다.

Servlet ?



Servlet: 내가 다 해줄게 !

Servlet ?

개발자가 직접 서블릿 정의시, `javax.servlet.http.HttpServlet`을 상속받아 구현

Interface
Servlet

This interface defines methods to **initialize a servlet, to service requests, and to remove a servlet from the server**. Defines methods that all servlets must implement.

Abstract Class
GenericServlet

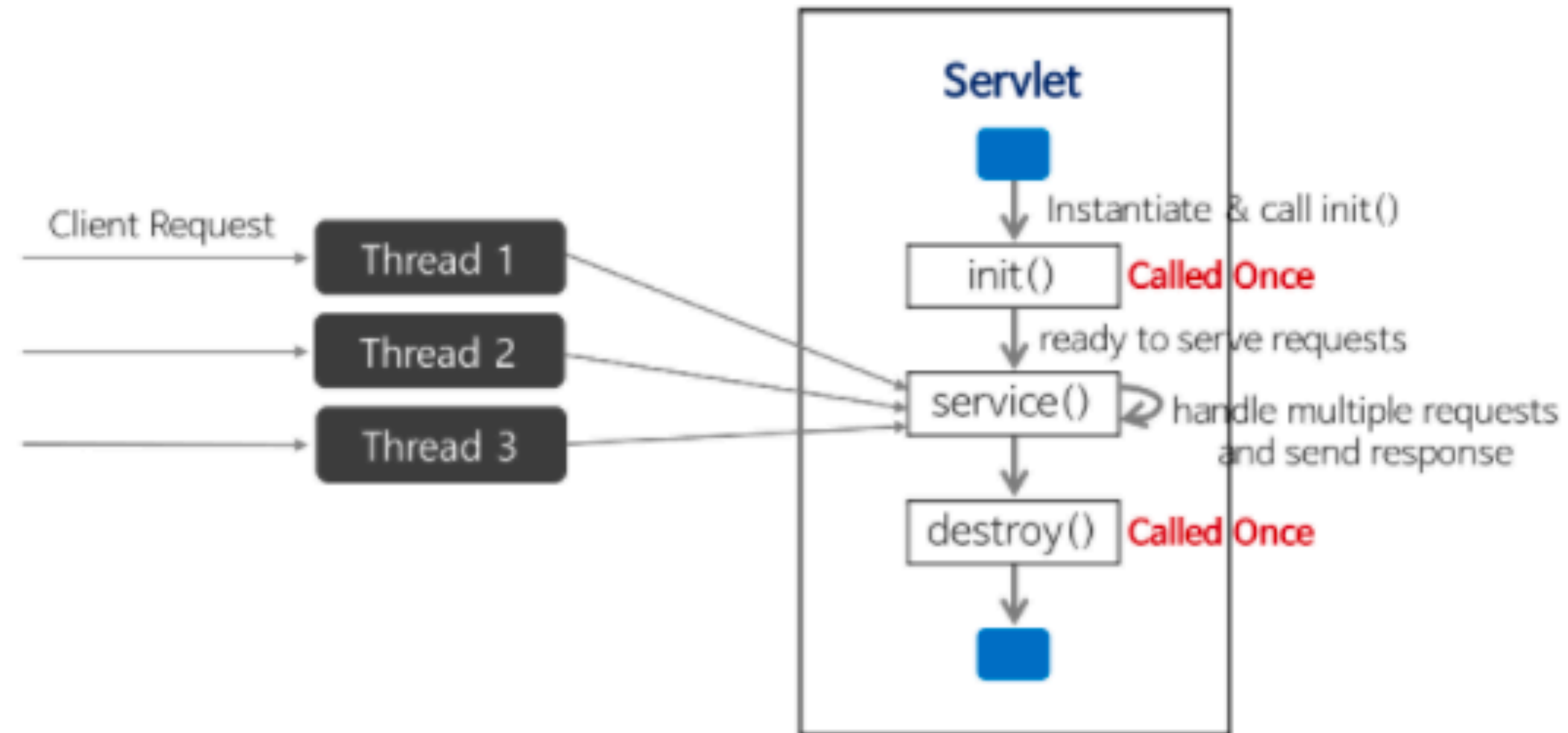
Defines a generic, protocol-independent servlet. GenericServlet makes writing servlets easier. It provides simple versions of the lifecycle methods `init` and `destroy` and of the methods in the `ServletConfig` interface.

Abstract Class
HttpServlet

Provides an abstract class to be subclassed to **create an HTTP servlet suitable for a Web site**.

Servlet ?

Servlet Life Cycle



Servlet ?

```
protected void service(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException
{
    String method = req.getMethod();
    if (method.equals(METHOD_GET)) {
        ...
        doGet(req, resp);
        ...
    } else if (method.equals(METHOD_HEAD)) {
        long lastModified = getLastModified(req);
        maybeSetLastModified(resp, lastModified);
        doHead(req, resp);
    } else if (method.equals(METHOD_POST)) {
        doPost(req, resp);
    } else if (method.equals(METHOD_PUT)) {
        doPut(req, resp);
    } else if (method.equals(METHOD_DELETE)) {
        doDelete(req, resp);
    } else if (method.equals(METHOD_OPTIONS)) {
        doOptions(req, resp);
    } else if (method.equals(METHOD_TRACE)) {
        doTrace(req, resp);
    }

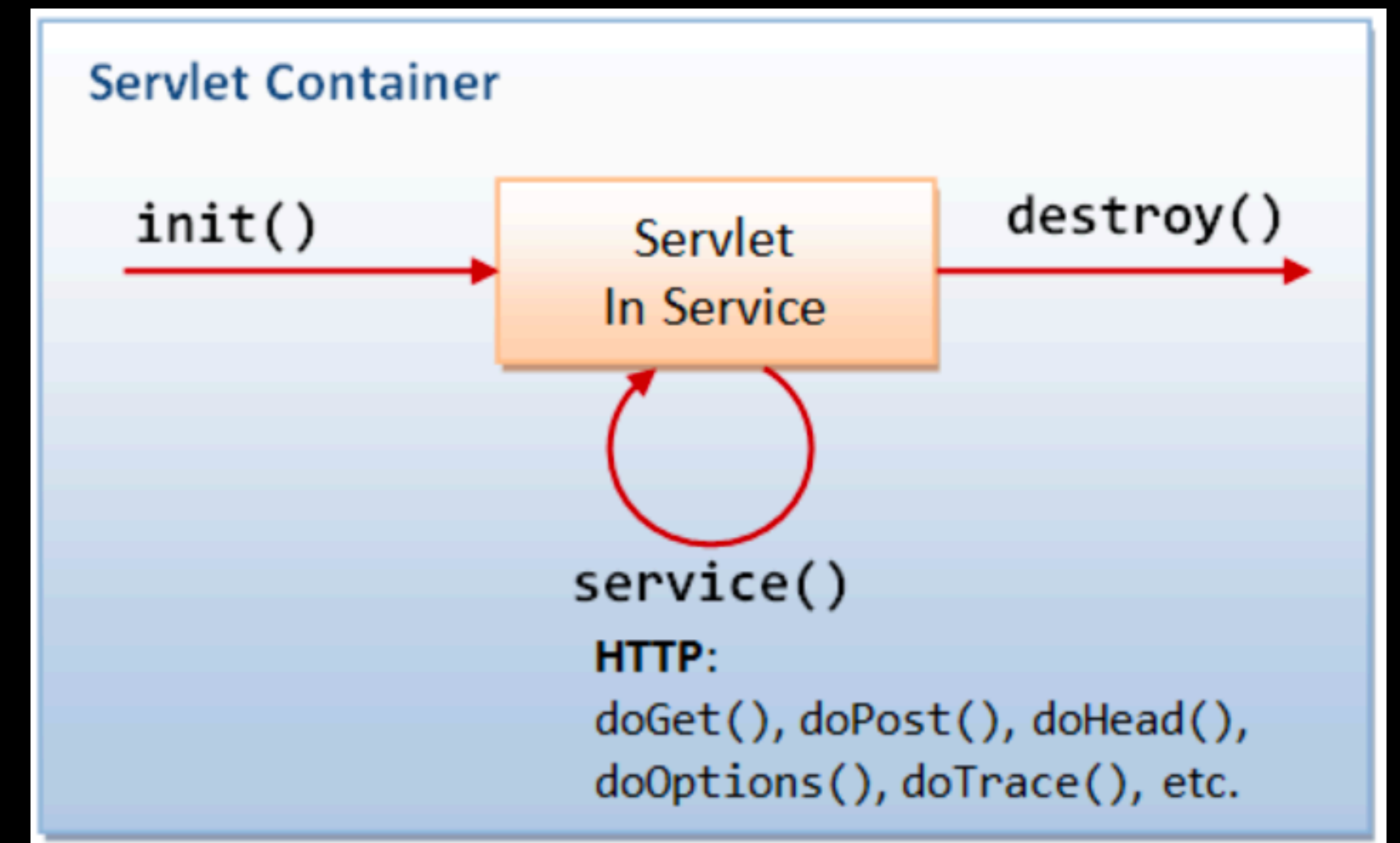
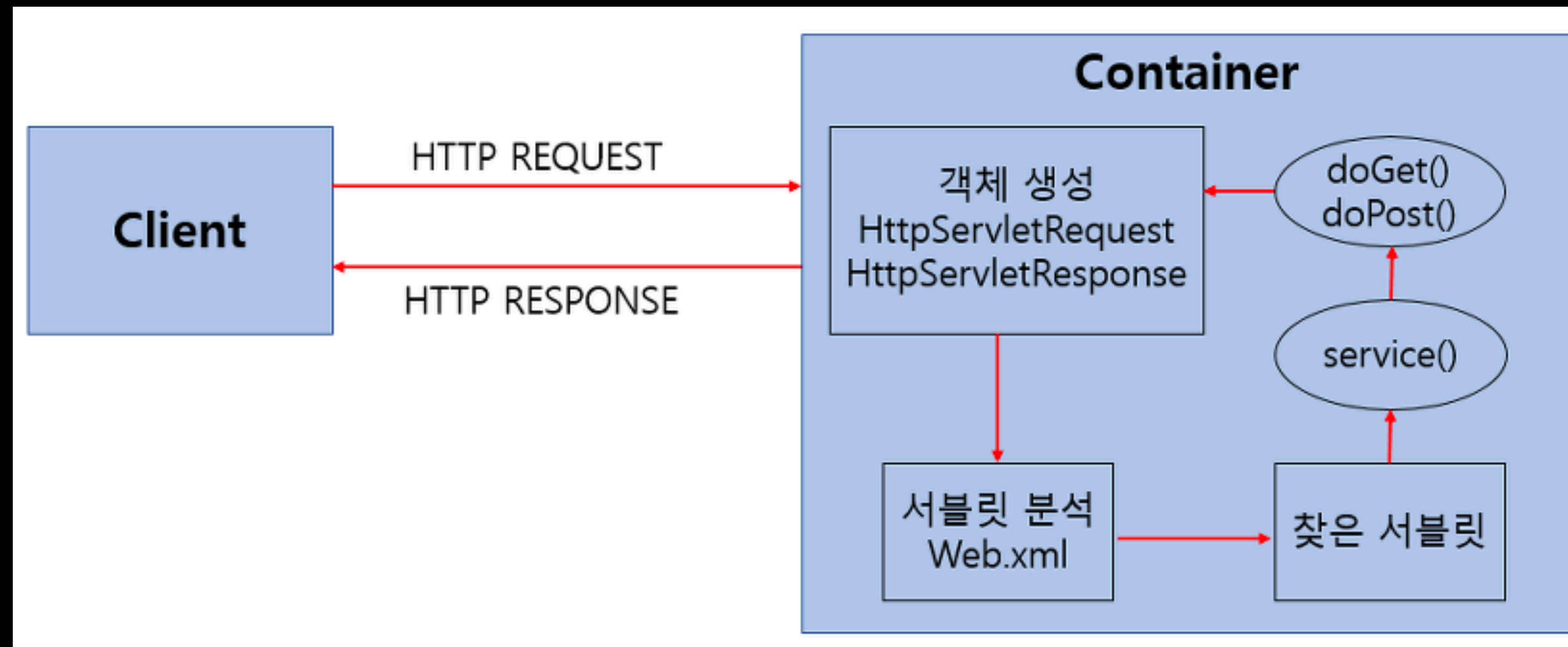
    ...
}
```


Servlet container

Servlet은 혼자 스스로 작동하는 단위가 아니다
이런 Servlet들을 관리하는 Servlet container가 필요 !

- Client request를 받고, response를 할 수 있도록 웹 서버와 통신
- Servlet 생명주기(Life cycle) 관리, Servlet 자체는 기본적으로 Singleton으로 관리
- Multithread support → Thread pool

Servlet container



Http의 경우, HttpServletRequest/HttpServletResponse

Servlet container

