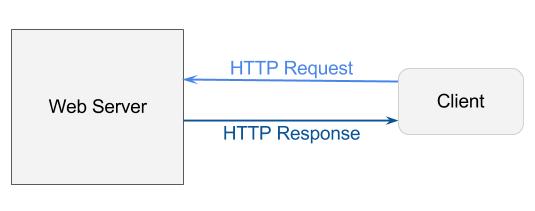
**1. What is a Web Server?**

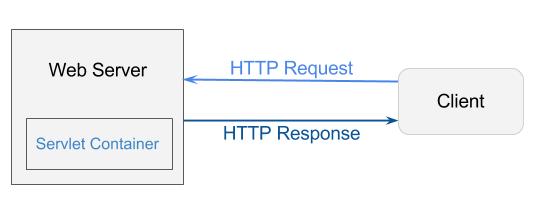
To know what is a Servlet container, we need to know what is a Web Server first

[](http://www.programcreek.com/wp-content/uploads/2013/04/web-server.jpg)

A *web server* uses HTTP protocol to transfer data. In a simple situation, a user type in a URL (e.g. [www.programcreek.com/static.html)](http://www.programcreek.com/static.html)) in browser (a client), and get a web page to read. So what the server does is sending a web page to the client. The transformation is in HTTP protocol which specifies the format of request and response message.

**2. What is a Servlet Container?**

As we see here, the user/client can only request static webpage from the server. This is not good enough, if the user wants to read the web page based on his input. The basic idea of Servlet container is using Java to dynamically generate the web page on the server side. So servlet container is essentially a part of a web server that interacts with the servlets.

[](http://www.programcreek.com/wp-content/uploads/2013/04/web-server-servlet-container.jpg)

*Servlet container* is the container for Servlets.

**3. What is a Servlet?**

[Servlet](http://docs.oracle.com/javaee/1.4/api/javax/servlet/Servlet.html) is an interface defined in javax.servlet package. It declares three essential methods for the life cycle of a servlet – init(), service(), and destroy(). They are implemented by every servlet(defined in SDK or self-defined) and are invoked at specific times by the server.

1. The init() method is invoked during initialization stage of the servlet life cycle. It is passed an object implementing the javax.servlet.ServletConfig interface, which allows the servlet to access initialization parameters from the web application.
2. The service() method is invoked upon each request after its initialization. Each request is serviced in its own separate thread. The web container calls the service() method of the servlet for every request. The service() method determines the kind of request being made and dispatches it to an appropriate method to handle the request.
3. The destroy() method is invoked when the servlet object should be destroyed. It releases the resources being held.

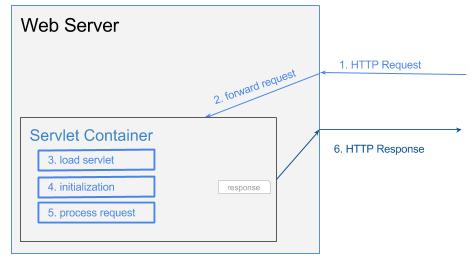
From the life cycle of a servlet object, we can see that servlet classes are loaded to container by class loader dynamically. Each request is in its own thread, and a servlet object can serve multiple threads at the same time(thread not safe). When it is no longer being used, it should be garbage collected by JVM.

Like any Java program, the servlet runs within a JVM. To handle the complexity of HTTP requests, the servlet container comes in. The servlet container is responsible for servlets’ creation, execution and destruction.

**4. How Servlet container and web server process a request?**

1. Web server receives HTTP request
2. Web server forwards the request to servlet container
3. The servlet is dynamically retrieved and loaded into the address space of the container, if it is not in the container.
4. The container invokes the init() method of the servlet for initialization(invoked once when the servlet is loaded first time)
5. The container invokes the service() method of the servlet to process the HTTP request, i.e., read data in the request and formulate a response. The servlet remains in the container’s address space and can process other HTTP requests.
6. Web server return the dynamically generated results to the correct location

The six steps are marked on the following diagram:

[](http://www.programcreek.com/wp-content/uploads/2013/04/servlet-container-life-cycle.jpg)

**5. The role of JVM**

Using servlets allows the JVM to handle each request within a separate Java thread, and this is one of the key advantage of Servlet container. Each servlet is a Java class with special elements responding to HTTP requests. The main function of Servlet contain is to forward requests to correct servlet for processing, and return the dynamically generated results to the correct location after the JVM has processed them. In most cases servlet container runs in a single JVM, but there are solutions when container need multiple JVMs