Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol 2, No.5, 2012
www.iiste.org

FTP SERVLET

Sandeep Kumar *, Anuj Kumar Singh, Vandana Bansala
Department of CSE / IT, Tula's Institute
Dehradun, Uttarakhand, INDIA
* E-mail of the corresponding author: sandeepkumarsiet@gmail.com

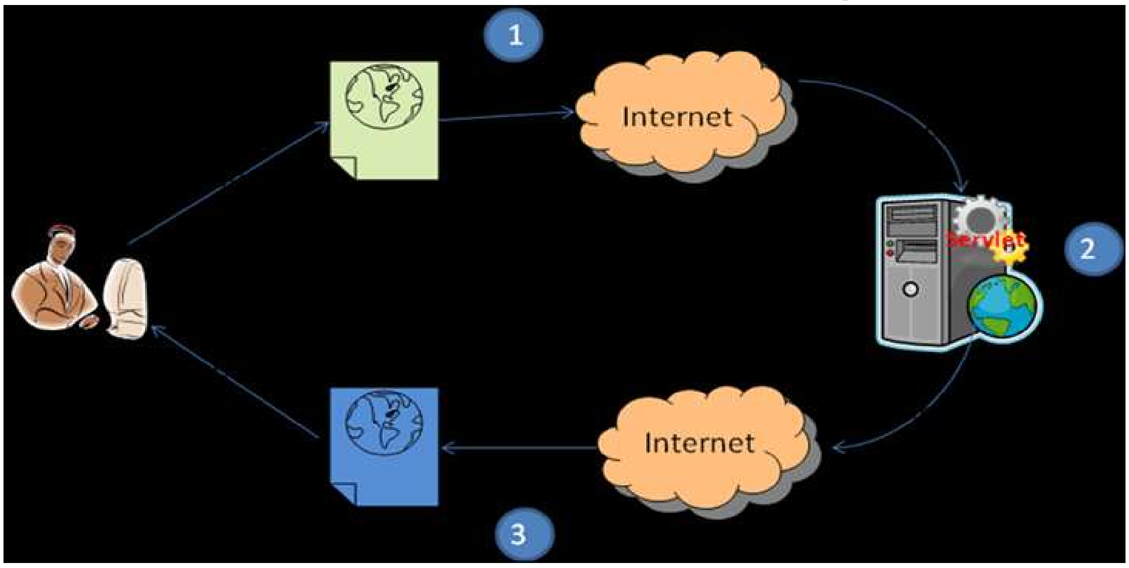
Abstract
This paper presents new idea about the servlet we know about http servlet and generic servlet case of http servlet we use http protocol for the server side programming this is a protocol dependent servlet if we use ftp and SMTP protocol instead of http protocol then servlet will be protocol independent. An FTP servlet is an intermediate application that resides between the FTP server and the FTP client. It works as a proxy interposed within client/server communications and helps to unload some of the computing power of the FTP server and distribute it to the FTP servlet. It also provides a firewall and proxy friendly file transfer environment by wrapping FTP traffic over HTTP. FTP traffic can be wrapped over Https using a SSL certificate to provide enhanced security.
Keywords: SMTP, wrapping FTP, enhanced security

1. Introduction

A servlet is a Java programming language class used to extend the capabilities of servers that host applications access via a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by Web servers. Thus, it can be thought of as a Java Applet that runs on a server instead of a browser. A Servlet is a java based server side web technology. As the name implies, it serves a client request and receives a response from the server. Technically speaking a Servlet is a Java class in Java EE that conforms to the Java Servlet API, a protocol by which a Java class may respond to requests. They are not tied to a specific client-server protocol, but are most often used with the HTTP protocol. Therefore, the word "Servlet" is often used in the meaning of "HTTP Servlet". Thus, a software developer may use a servlet to add dynamic content to a Web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to non-Java dynamic Web content technologies such as CGI and ASP.NET. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL rewriting.

2. How A Servlet Work?

Three methods are central to the life cycle of a servlet. These are ***init( )***, ***service( )***, and ***destroy( )***. They are implemented by every servlet and are invoked at specific times by the server. Let us consider a typical user scenario to understand when these methods are when we use FTT, SMTP in Place of HTTP protocol.



1. Assume that a user enters a Uniform Resource Locator (URL) to a web browser.
    - The browser then generates an FTP request for this URL.
    - This request is then sent to the appropriate server.
2. The FTP request is received by the web server.
    - The server maps this request to a particular servlet.
    - The servlet is dynamically retrieved and loaded into the address space of the server.
3. The server invokes the init() method of the servlet.
    - This method is invoked only when the servlet is first loaded into memory.
    - It is possible to pass initialization parameters to the servlet so it may configure itself.
4. The server invokes the service() method of the servlet.
    - This method is called to process the FTP request.
    - You will see that it is possible for the servlet to read data that has been provided in the FTP request.
    - It may also formulate an FTP response for the client.
5. The servlet remains in the server's address space and is available to process any other FTP requests received from clients.
    - The service() method is called for each FTP request.
6. The server may, at some point, decide to unload the servlet from its memory.
    - The algorithms by which this determination is made are specific to each server.
7. The server calls the destroy() method to relinquish any resources such as file handles that are allocated for the servlet; important data may be saved to a persistent store.
8. The memory allocated for the servlet and its objects can then be garbage collected.

2.1 Example

The following example servlet prints how many times its service() method was called.
Note that HttpServlet is a subclass of GenericServlet, an implementation of the Servlet interface.
The service () method of HttpServlet class dispatches requests to the methods doGet(), doPost(), doPut(), doDelete(), and so on; according to the HTTP request. In the example below method service() is overridden and does not distinguish which HTTP request method it serves.

3. Using Genericservlet For FTP

The protocol is handled by the servlet container, not the servlet itself. If you want to have FTP commands sent to a servlet, you need a special servlet container that handles the FTP protocol and sends the received FTP commands to a servlet. As far as I know there are no servlet containers available that can do this.
According to the the Javadoc the only method that you must override is service(). You could look at the patterns used in HttpServlet to get some ideas. For the FTP protocol I could see things like doPut(), doGet(), doList(), and so on. You service() method would parse appropriately and call those methods on a FTPServlet.
The real issue, as was pointed out, is that the container handles the actual protocol. This is a not so subtle side effect of the J2EE servlet specification. Sure, you could have a FTPServlet, a SMTPServlet, and whatever other kind of protocol you want. But at the lowest level something has to be listening for a connection on a particular port and route packets to and from that port appropriately. Tomcat follows the servlet spec but doesn't have a Connector (a Tomcat term) for FTP. It really wouldn't be too bad to write one. Looking at the docs for Http11Processor you'll see that it implements two interfaces. Basically you want to create your own connector for Tomcat.
All of this is great, but forget about portability. If you want this to work in WebLogic or WebSphere you'll have to figure out their API's tool.

4. Ftp Servlet

An FTP servlet is an intermediate application that resides between the FTP server and the FTP client. It works as a proxy interposed within client/server communications and helps to unload some of the computing power of the FTP server and distribute it to the FTP servlet. It also provides a firewall and proxy friendly file transfer environment by wrapping FTP traffic over HTTP. FTP traffic can be wrapped over Https using a SSL certificate to provide enhanced security.

5. Architecture

FTP clients can connect to the FTP servlet through the Internet. In most cases FTP is wrapped over an application layer protocol. Most commonly used are HTTP (for easy, unencrypted transfers) or HTTPs (for encrypted transfers).

The use of HTTPs requires an SSL certificate to be present at the site of the FTP servlet. A number of simultaneous connections can be made to the FTP servlet. The number of connections is restricted to the computing power of the server. The number of end-users supported through the number of connections is usually more. As all connected end-users aren't "active" until they make a request from the server. Consequently, the number of end-users simultaneously online on the FTP server can be greater than the number of active connections supported by the FTP server.

6. Exploring Other Protocols:

After the successful usage of HTTP Explorer we plan to apply the main idea to other protocols and applications as well. There are basically two requirements for a protocol or an application, it must

I. be text-based and
II. have a request-response characteristic.

These requirements have their roots in the characteristics of HTTP. Possible candidates for protocols include WebDAV and SOAP, but it is also possible to access applications such as compilers or simulators In this case the communication between the tool and the target application is either based on an API directly used by our tool or based on a network protocol. If there is no such protocol available a simple
Wrapper based on TCP/IP can provide this service. Our goal is to be define an API, such that all learning tools can use a common kernel. Also we want to develop a common testing environment.

7. Conclusion

If we use ftp servlet in place of http servlet then speed will be much better than http and unwanted thread do not effect or we can secure band width GenericServlet defines a generic, protocol-independent servlet. GenericServlet gives a blueprint and makes writing servlet easier. GenericServlet provides simple versions of the lifecycle methods init and destroy and of the methods in the ServletConfig interface. GenericServlet implements the log method, declared in the ServletContext interface. To write a generic servlet, it is sufficient to override the abstract service method

8. References

[1] Dustin R. Callaway. *Inside Servlets: Server-Side Programming for the JavaTM Platform*. Addison Wesley Longman, Inc., Redaing, Massachusetts, 1999.
[2] David Flanagan. *Java In A Nutshell, second edition*. O'Reilly and Associates, Inc., Sebastopol, CA, 1997.
[3] Mohammad R. Islam, Dr. Frederick C. Harris, Jr., advisor. Implementation of Interactive Course Web Site August 2000
[4] http://pdftutorial.net/pdf/1/introduction-to-java-servlet-technology.html
[5] http://www.coderanch.com/t/361522/Servlets/java/GenericServlet-FTP
[6] http://javapapers.com/servlet/difference-between-httpservlet-and-genericservlet