ICON OF JAVA

ADV. JAVA

Servlets

2. Client server arch

Mr. Nagoor Babu M.Tech

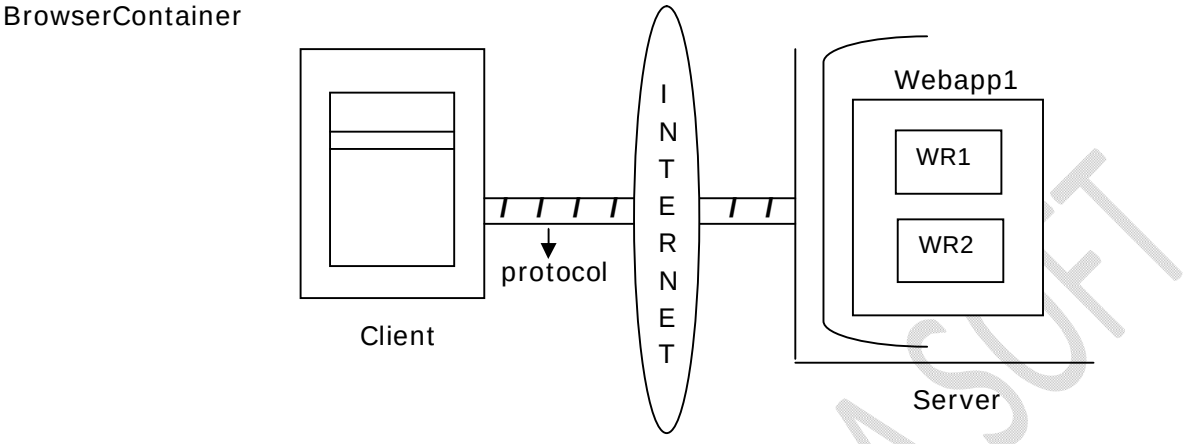
[ICON of JAVA]
(Sun certified & Realtime Expert)
Ex. HCL Employee
Trained Lakhs of Students
for last 14 years across INDIA

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Client-Server Architecture



From the above representation, there are three major components in client-server architecture.

1. Client
2. Protocol
3. Server

1. Client:

The main purpose of the Client in client-server architecture is to send request to the server and to set the responses from server.

To send request and to set response from server we need to use a tool at client machine called as **Browser.** In client-server application browser is acting as client.

To access a particular resource available at server from client browser we need to specify a particular string at browser address bar called **URI.**

There are two types of URI's

1. URL 2. URN

Q: What are the differences between URI, URL and URN?

Ans: URIis a string specification provided at client address bar, it can be used to refer a particular resource available at server machine.

URL is a string specification provided at client address bar, it can be used to refer a particular resource available at server machine through its locator.

URN is a string specification, it can be used to refer a particular resource available at server machine through its logical name.

Note 1: In case of servlets, locator is an URL pattern defined in web.xml file.

Note 2: In case of servlets, logical name is a name specified along with <servlet-name> tag in web.xml file.

Note 3: Almostall the servers are able to accept URL kind of request, but almost all the servers are not accept URN kind of request**.**

If we want to provide URL at client address bar then we have to use the following syntax.

Protocol_Name://Server_IP_Address:Server_Port_No/Application_Context/Resource_Name[Query_String]

Ex: http://121.120.92.98.8080/loginapp/logon?uname=abc&upwd=abc

Here Query_String, i.e. uname=abc&upwd=abc is optional.

Q: What is the difference between IP Address and Port Number?

Ans: IP Address is an unique identification for each and every machine over Network, which will be provided by Network Manager at the time of Network Configuration.

Port Number is an unique identification to each and every process being executed at the same machine and which could be provided by the local operating system.

Q: What is Query String and what is the purpose of Query String in web applications?

Ans: Query String is a collection of name-value pair appended to the URL, which can be used to pass input parameters to the respective server side resource in order to perform the required server side action.

2. Protocol:

The main job of the Protocol in client-server architecture is to carry the request data from client to server and to carry the response data from server to client.

Protocol is a set of rules and regulations, which can be used to carry the data from one machine to another machine over the Network.

Ex: TCP/IP, FTP, HTTP, SMTP, ARP, RARP.....

In general in web applications, we will use http protocol to send request from client to server and to set response from server to client.

Q: What is the requirement of http protocol in web applications?

Ans: In web applications, to transfer the data between client and server we require a protocol, it should be

1. A Connectionless Protocol
2. A Stateless Protocol
3. A compatible Protocol to carry hypertext data.

Where Connectionless Protocol is protocol, it should not require a physical connection, but require a logical connection to carry the data.

Where Stateless Protocol is a protocol, which should not remember previous request data at the time of processing the later request.

In general in client server application, request data will be transferred from client to server in the form of hypertext data and the response data will be transferred from server to client in the form of hypertext data so that we require a Compatible Protocol to carry hypertext data between client and server.

Among all the protocols http protocol is able to satisfy all the above requirements so that we will use http protocol in web applications.

Q: How http protocol is able to manage stateless nature?

Ans: In client server applications, when we send a request from client to server protocol will pick up the request and perform the following actions.

1. Protocol will establish a virtual socket connection between client and server as per the server IP address and protocol which we provided in URL.
2. Protocol will prepare the request format with header part and body part, where header part will manage all the request headers(metadata about client) and body part will manage request parameters(the data which was provided by the user at client browser).
3. After preparing request format protocol will carry request format to server through the virtual socket connection.

Upon receiving request from protocol server will identify the request resource, execute generate dynamic response and dispatch that dynamic response to client. When server dispatch the dynamic response to the client protocol will pick up the response and perform the following actions.

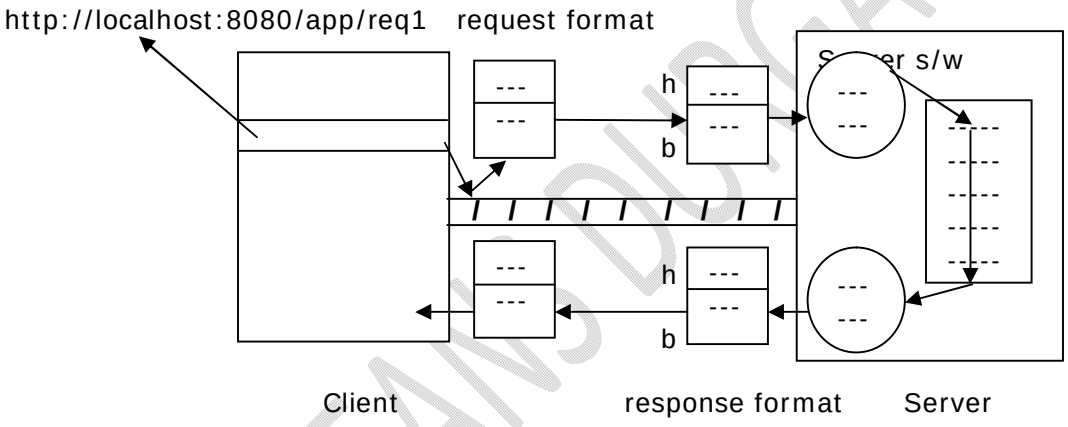
1. Protocol will prepare response format with header part and body part, where header part will manage response headers(metadata about the dynamic response) and body part will manage the actual dynamic response.
2. After setting response format protocol will carry response format to client.

3. When the dynamic response reached to client protocol will terminate the virtual socket connection, with this protocol will eliminate the present request data from its memory.

In the above context, the present request data will be managed by the protocol up to the connections existence, will protocol connection has terminated then protocol will not manage request data.

Due to the above reason http protocol is unable to manage clients previous request data at the time of processing later request. Therefore, http protocol is a stateless protocol.

Note: If we use http protocol, a stateless protocol in our web applications then we are unable to manage clients previous request data, but as per the application requirements we need to manage clients previous request data at the time of processing later request. In this context, to achieve the application requirement we have to use a set of explicit mechanism at server side called as **Session Tracking Mechanisms.**



In web applications, with http protocol we are able to specify different types of requests at client browser. The above flexibility is possible for the http protocol due to the availability of 7 number of http methods called as BIG 7 http methods.

Http protocol has provided the following http methods along with http1.0 version.

1. GET
2. POST
3. HEAD

Http protocol has provided the following http methods as per http1.1 version.

1. OPTIONS
2. PUT
3. TRACE

4. DELETE

Http1.1 version has provided a reserved http method i.e. CONNECT.

Q: What are the differences between GET and POST methods?

or

What are the differences between doGet(_,_) and doPost(_,_) methods?

Ans: 1. GET request type is default re quest type in web applications, but POST request type is not default request type.

2. GET request type should not have body part in the request format, but POST request type should have body part in the request format.

3. If we specify request parameters along with GET request then that request parameters will be transferred to server through request format header part due to the lack of body part.

In general request format header part will have memory limitation so that it is able to carry maximum 256 no. of characters. Therefore, GET request is able to carry less data from client to server

Due to the availability of body part in POST request all the request parameters will be transferred to server through request format body part, here there is no memory limitation in request format body part so that the POST request is able to carry large data from client to server.

4. If we specify request parameters along with GET request then GET request will display all the request parameters at client address bar as query string. Therefore, GET request is able to provide less security for the client data.

If we provide request parameters along with POST request then GET request will not display the request parameters at client address bar. Therefore, POST request is able to provide very good security for the client data.

5. In general in web applications, GET request can be used to get the data from server i.e. download the data from server.

In general in web applications, POST request can be used to post the data to server i.e. upload the data on to the server.

Note: Bookmarks are supported by GET request and which are not supported by POST request.

Q: What is the difference between GET request and HEAD request?

Ans: If we send Get request for a particular resource available at server machine then server will send only the requested resource as a response to client.

If we send HEAD request for a particular resource available at server then server will send requested resource as well as the metadata about the requested resource as response.

Note: Internally HEAD request uses GET request to get the requested resource from server.

OPTIONS Request:

The main purpose of the OPTIONS request type is to get the http methods which are supported by the present server.

Note: In general http protocol has provided by 7 http methods conceptually, supporting all the methods or some of the methods or none is completely depending on the server implementation provided by the server providers.

With this convention we are unable to credit how many number of http methods are supported by the present application server, where to credit the http methods which are supported by the present server we have to use OPTIONS request type.

Q: What is the difference between POST request and PUT request?

Ans: Both POST request and PUT request can be used to upload the data on the server machine. To upload the data on server machine if we use POST request then it is not mandatory to specify particular address location along with POST request.

To upload the data on server machine if we use PUT request then it is mandatory to specify server side location along with PUT request.

TRACE Request:

The main purpose of the TRACE request is to get the working status of a particular resource available at the server machine. TRACE request type is able to execute its functionality like echo server.

DELETE Request:

The main purpose of this request type is to delete a particular resource available at server machine.

Note: Almost all the servers may not support PUT, DELETE and request types as per their security constraints. In general almost all the servers are able to support GET and POST request types.

Status Codes:

In web applications, the main purpose of the status codes is to give the status of the request processing to the client.

Http1.1 version has provided all the status codes in the form of number representations. As per the web application requirement http1.1 version has provided the following status codes.

1xx --→ 100 to 199 --→ Inforamtional status codes

2xx --→ 200 to 299 --→ Success related status codes

3xx --→ 300 to 399 --→ Redirectional status codes

4xx --→ 400 to 499 --→ Client side error status codes

5xx --→ 500 to 599 --→ Server side error status codes

In general in web applications, when we send a request from client to server, server will identify the requested resource, execute it and generate dynamic response to client.

In the above context, when server dispatches the response protocol will pick up the response and prepare response format, where the dynamic response will be stored in the response format body part.

At the time of processing response format server will provide the respective status code value in the response format header part i.e. with status line field.

When protocol carry the response format to the client then client will pick up the status code value from status line field, prepare itself to get the response from response format body part.

3. Server:

The main purpose of the server in client server applications is to pick up the request from client, identify the requested resource, generate the dynamic response and dispatch dynamic response to client.

Note: Servlet is a program available at server machine, it is not capable to pick up the request and dispatch response to client, if server execute servlet program then some dynamic response will be generated.

Examples of Servers: Apache Tomcat, BEA Weblogic, IBM Websphere, Macromedia JRun, SUN Sunone, J2EE, GlassFish and so on.

There are 2 types of servers to execute enterprise applications.

1. Web Servers
2. Application Servers

Q: What are the differences between Web Servers and Application Servers?

Ans: 1.Web Server is a server, which will provide very good environment to execute web applications only. But Application servers will provide very good environment to execute any type of J2EE applications like web applications, distributed applications and so on.

2. In general web servers will not provide all the middle ware services. But application servers will provide all the middle ware services like JND, Jdbc and so on as in-built support.

Application server = Web server + Middleware services

Note: Initially the main intention of web servers is to execute static resources in order to generate static response and the main intention of application servers is to execute dynamic resources in order to generate dynamic response.

If we want to specify a particular machine has server machine then we have to install a particular server software, when we install server software on server machine automatically that server software will be available in the form of the following 2 modules.

1. Main Server
2. Container

Q: What is the difference between Main server and Container?

Ans: When we send a request from client to server then main server will pick up the request from protocol and check whether the request data is in well-formed formator not**,** if it is not in well-formed format then main server will stop request their itself and generate the respective response to client.

If the request data is in well-formed format then that request will by pass to container, wherecontainer will identify the requested resource, execute it, generate dynamic response and dispatch dynamic response to main server, where main server will bypass response to client through the protocol.

In general containers could be classified into the following 2 ways.

1. As per the technology which we used to design server side component. There are some containers

1. Servlet Container --→ To execute servlets

2. Jsp Container --→ To execute Jsp's

3. EJB Container --→ To execute EJB's

Note: All the above specified containers can be used to execute the respective components because the above containers have implemented the respective technology API.

2. As per the containers physical existency there are 3 types of containers.

1. Standalone Container:

It is an integration of main server and container as a single program.

2. Inprocess Container:

It is a container existed inside the main server.

3. Out of process Container:

It is a container existed outside of the main server.