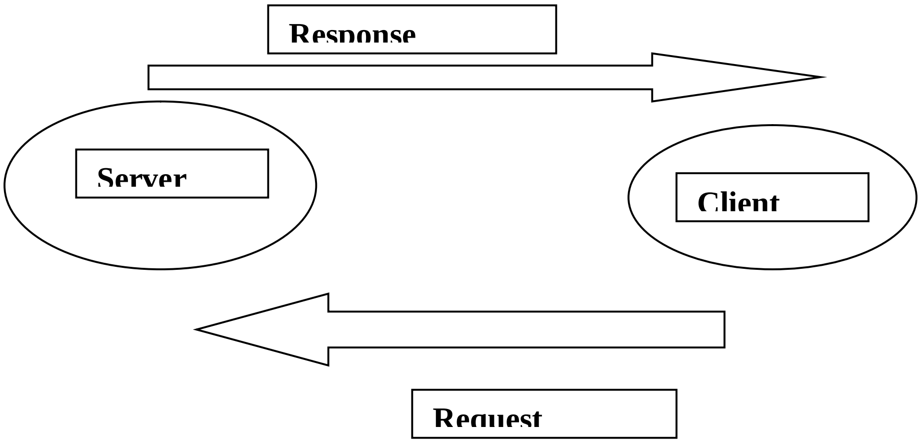
Methods For Sending Requests

1. Get and Post methods:



There are two methods by which we can make request to the server. These are:

1. Get Method
2. Post method

1. Get Method: In get method data is passed through a query string in the address bar. Through get method small amount of data upto 2MB only can be transferred. The data send through get method is insecure anyone can easily hack it because data is transferred through get method. Data transferred through get method is faster.
2. Post Method: In post method data is not passed through query string in the address bar. By default max data transferred through post method is 2MB but it can be increased by increasing max_post_size in the php.ini file. Max post data is not fixed in post method. Data send through post method is secure. This method is slower than get method.

Q:1 What are the differences between Get and post methods in form submitting. give the case where we can use get and we can use post methods?

A:1 When to use GET or POST

The HTML 2.0 specification says, in section Form Submission (and the HTML 4.0 specification repeats this with minor stylistic changes):

–>If the processing of a form is idempotent (i.e. it has no lasting observable effect on the state of the world), then the form method should be GET. Many database searches have no visible side-effects and make ideal applications of query forms.

–

–>If the service associated with the processing of a form has side effects (for example, modification of a database or subscription to a service), the method should be POST.

How the form data is transmitted?

quotation from the HTML 4.0 specification

–> If the method is “get” - -, the user agent takes the value of action, appends a ? to it, then appends the form data set, encoded using the application/x-www-form-urlencoded content type. The user agent then traverses the link to this URI. In this scenario, form data are restricted to ASCII codes.

–> If the method is “post” –, the user agent conducts an HTTP post transaction using the value of the action attribute and a message created according to the content type specified by the enctype attribute.

Quote from CGI FAQ

Firstly, the the HTTP protocol specifies differing usages for the two methods. GET requests should always be idempotent on the server. This means that whereas one GET request might (rarely) change some state on the Server, two or more identical requests will have no further effect.

This is a theoretical point which is also good advice in practice. If a user hits “reload” on his/her browser, an identical request will be sent to the server, potentially resulting in two identical database or guestbook entries, counter increments, etc. Browsers may reload a GET URL automatically, particularly if cacheing is disabled (as is usually the case with CGI output), but will typically prompt the user before re-submitting a POST request. This means you’re far less likely to get inadvertently-repeated entries from POST.

GET is (in theory) the preferred method for idempotent operations, such as querying a database, though it matters little if you’re using a form. There is a further practical constraint that many systems have built-in limits to the length of a GET request they can handle: when the total size of a request (URL+params) approaches or exceeds 1Kb, you are well-advised to use POST in any case.

I would prefer POST when I don’t want the status to be change when user resubmits. And GET when it does not matter.

Super Global Arrays

There are 8 Super Global Arrays are there in php. They are as follows:

1. $_GET: The data passed through get method is retrieved by $_GET super global array.
2. $_POST: The data passed through post method is retrieved by $_POST super global array.
3. $_REQUEST: If we don't know by which method (get or post) the data is sent then by the use of $_REQUEST super global array we can retrieve the data.
4. $_FILES: If a text file or any other type of file is send to the server then we can retrieve it by the use of $_FILES super global array.
5. $_SESSION: Any thing stored in the session can be retrieved by the use of $_ SESSION super global array also used to register session variables.
6. $_COOKIE: Any thing stored in the cookie can be retrieved by the use of $_ COOKIE super global array.
7. $_SERVER
8. $_GLOBALS