

GET and POST Function

H S Rana

CIT,UPES

February 26, 2014



There are two ways the browser client can send information to the web server.

- The GET Method
- The POST Method

Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

<http://www.test.com/index.htm?name1=value1&name2=value2>



The GET Method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character.

<http://www.test.com/index.htm?name1=value1&name2=value2>

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY_STRING environment variable.
- The PHP provides \$_GET associative array to access all the sent information using GET method.



GET Method Example

This the example of get function

```
<?php
    if( $_GET["name"] || $_GET["age"] )
    {
        echo "Welcome ". $_GET['name']. "<br />";
        echo "You are ". $_GET['age']. " years old.";
        exit();
    }
?>

<html>
<body>
    <form action="get.php" method="GET">
    Name: <input type="text" name="name" />
    Age: <input type="text" name="age" />
    <input type="submit" />
    </form>
</body>
```



POST Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides \$_POST associative array to access all the sent information using POST method.



Example

This is the example of post function

```
<?php
    if( $_POST["name"] || $_POST["age"] )
    {
        echo "Welcome ". $_POST['name']. "<br />";
        echo "You are ". $_POST['age']. " years old.";
        exit();
    }
?>

<html>
<body>
    <form action="post.php" method="POST">

    Name: <input type="text" name="name" />
    Age: <input type="text" name="age" />

    <input type="submit" />
```

