

Chapter 5 – Passing Information with PHP

Here 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.

```
name1=value1&name2=value2&name3=value3
```

Spaces are removed and replaced with the + character and any other nonalphanumeric characters are replaced with a hexadecimal values. After the information is encoded it is sent to the server.

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.

Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.

```
http://localhost/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.

Try out following example by putting the source code in test.php script.

```
<?php
if( $_GET["name"] || $_GET["age"] )
{
    echo "Welcome ". $_GET['name']. "<br />";
    echo "You are ". $_GET['age']. " years old.";
    exit();
}
?>
<html>
<body>
    <form action="<?php $_PHP_SELF ?>" method="GET">
        Name: <input type="text" name="name" />
        Age: <input type="text" name="age" />
        <input type="submit" />
    </form>
</body>
</html>
```

The 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.

Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

Try out following example by putting the source code in test.php script.

```
<?php
if( $_POST["name"] || $_POST["age"] )
{
    echo "Welcome ". $_POST['name']. "<br />";
    echo "You are ". $_POST['age']. " years old.";
    exit();
}
?>
<html>
<body>
    <form action="<?php $_PHP_SELF ?>" method="POST">

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

    <input type="submit" />
    </form>
</body>
</html>
```

The \$_REQUEST variable

The PHP \$_REQUEST variable contains the contents of both \$_GET, \$_POST, and \$_COOKIE. We will discuss \$_COOKIE variable when we will explain about cookies.

The PHP \$_REQUEST variable can be used to get the result from form data sent with both the GET and POST methods.

Try out following example by putting the source code in test.php script.

```
<?php
if( $_REQUEST["name"] || $_REQUEST["age"] )
{
    echo "Welcome ". $_REQUEST['name']. "<br />";
    echo "You are ". $_REQUEST['age']. " years old.";
    exit();
}
?>
<html> <body>
    <form action="<?php $_PHP_SELF ?>" method="POST">

    Name: <input type="text" name="name" />
    Age: <input type="text" name="age" />
    <input type="submit" />
    </form>
</body>
</html>
```

Difference between POST and GET methods

GET:

- Parameters remain in browser history because they are part of the URL
- Can be bookmarked.
- GET method should not be used when sending passwords or other sensitive information.
- 7607 character maximum size.
- Url example: page2.php?category=sport

POST:

- Parameters are not saved in browser history.
- Can not be bookmarked.
- POST method used when sending passwords or other sensitive information.
- 8 Mb max size for the POST method.
- Url example: page2.php