



SOUTHERN LUZON STATE UNIVERSITY LUCBAN, QUEZON COLLEGE OF INDUSTRIAL TECHNOLOGY



SUBJECT: ITE24 – COGNATE/PROFESSIONAL COURSE 3
COURSE: BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY III
INSTRUCTOR: MR. JOMER R. SAMSON
S/Y: 2ND SEM 2024/2025

GET AND POST METHOD

HTTP GET and POST Methods in PHP

HTTP: The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a website may be the server. A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

There are 2 HTTP request methods:

1. GET: Requests data from a specified resource. **7.**
2. POST: Submits data to be processed to a specified resource. **8.**

GET Method: In the GET method, the data is sent as URL parameters that are usually strings of name and value pairs separated by ampersands (&). In general, a URL with GET data will look like this:

<http://www.example.com/action.php?name=Sam&weight55>

Here, the bold parts in the URL denote the GET parameters and the italic parts denote the value of those parameters. More than one parameter=value can be embedded in the URL by concatenating with ampersands (&). One can only send simple text data via GET method.

```
<?php
error_reporting(0);
if( $_GET["name"] || $_GET["weight"] )
{
    echo "Welcome ". $_GET['name']. "<br />";
    echo "You are ". $_GET['weight']. " kgs in weight.";
    exit();
}
?>
<html>
<body>
<form action="<?php $_PHP_SELF ?>" method="GET">
    Name: <input type="text" name="name" />
    Weight:<input type="text" name="weight" />
    <input type="submit" />
</form>
</body>
</html>
```

POST Method: In the POST method, the data is sent to the server as a package in a separate communication with the processing script. Data sent through the POST method will not be visible in the URL.

```
<?php
error_reporting(0);
if( $_POST["name"] || $_POST["weight"] )
    {
        if (preg_match("/^[^A-Za-z'-]"/,$_POST['name'] ))
        {
            die ("invalid name and name should be alpha");
        }
        echo "Welcome ". $_POST['name']. "<br />";
        echo "You are ". $_POST['weight']. " kgs in weight.";
        exit();
    }

?>
<html>
<body>
<form action = "<?php $_PHP_SELF ?>" method = "POST">
    Name: <input type = "text" name = "name" />
    Weight: <input type = "text" name = "weight" />
           <input type = "submit" />

</form>
</body>
</html>
```

PHP Form Processing

HTML forms are used to send the user information to the server and returns the result back to the browser. For example, if you want to get the details of visitors to your website, and send them good thoughts, you can collect the user information by means of form processing. Then, the information can be validated either at the client-side or on the server-side. The final result is sent to the client through the respective web browser. To create a HTML form, form tag should be used.

Attribute	Description
name or id	It specifies the name of the form and is used to identify individual forms.
Action	It specifies the location to which the form data has to be sent when the form is submitted.
method	It specifies the HTTP method that is to be used when the form is submitted. The possible values are get and post . If get method is used, the form data are visible to the users in the url. Default HTTP method is get .
encType	It specifies the encryption type for the form data when the form is submitted.
novalidate	It implies the server not to verify the form data when the form is submitted.

Controls used in forms: Form processing contains a set of controls through which the client and server can communicate and share information. The controls used in forms are:

- Textbox: Textbox allows the user to provide single-line input, which can be used for getting values such as names, search menu and etc.
- Textarea: Textarea allows the user to provide multi-line input, which can be used for getting values such as an address, message etc.
- DropDown: Dropdown or combobox allows the user to provide select a value from a list of values.
- Radio Buttons: Radio buttons allow the user to select only one option from the given set of options.
- CheckBox: Checkbox allows the user to select multiple options from the set of given options.
- Buttons: Buttons are the clickable controls that can be used to submit the form.

Creating a simple HTML Form: All the form controls given above are designed by using the input tag based on the type attribute of the tag. In the below script, when the form is submitted, no event handling mechanism is done. Event handling refers to the process done while the form is submitted. These event handling mechanisms can be done by using JavaScript or PHP. However, JavaScript provides only client-side validation. Hence, we can use PHP for form processing.

```
<!DOCTYPE html>
<html>

<head>
    <title>Simple Form Processing</title>
</head>

<body>
    <form id="form1" method="post">
        FirstName:
        <input type="text" name="firstname" required/>
        <br>
        <br>
        LastName
        <input type="text" name="lastname" required/>
        <br>
        <br>
        Address
        <input type="text" name="address" required/>
        <br>
        <br>
        Email Address:
        <input type="email" name="emailaddress" required/>
        <br>
        <br>
        Password:
        <input type="password" name="password" required/>
        <br>
        <br>
        <input type="submit" value="Submit"/>
    </form>
</body>
</html>
```

Form Validation: Form validation is done to ensure that the user has provided the relevant information. Basic validation can be done using HTML elements. For example, in the above script, the email address text box is having a type value as “email”, which prevents the user from entering the incorrect value for an email. Every form field in the above script is followed by a required attribute, which will intimate the user not to leave any field empty before submitting the form. PHP methods and arrays used in form processing are:

- `$_GET[]`: It is used to retrieve the information from the form control through the parameters sent in the URL. It takes the attribute given in the url as the parameter.
- `$_POST[]`: It is used to retrieve the information from the form control through the HTTP POST method. It takes name attribute of corresponding form control as the parameter.
- `$_REQUEST[]`: It is used to retrieve an information while using a database.