

PHP GET and POST

Methods of Sending Information to Server

A web browser communicates with the server typically using one of the two HTTP (Hypertext Transfer Protocol) methods — GET and POST. Both methods pass the information differently and have different advantages and disadvantages, as described below.

The GET Method

In 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=john&age=24
```

The bold parts in the URL are the GET parameters and the italic parts are 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.

Advantages and Disadvantages of Using the GET Method

- Since the data sent by the GET method are displayed in the URL, it is possible to bookmark the page with specific query string values.
- The GET method is not suitable for passing sensitive information such as the username and password, because these are fully visible in the URL query string as well as potentially stored in the client browser's memory as a visited page.
- Because the GET method assigns data to a server environment variable, the length of the URL is limited. So, there is a limitation for the total data to be sent.

PHP provides the superglobal variable `$_GET` to access all the information sent either through the URL or submitted through an HTML form using the `method="get"`.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Example of PHP GET method</title>
</head>
<body>
  <?php
if(isset($_GET["name"])){
    echo "<p>Hi, " . $_GET["name"] . "</p>";
}
?>
<form method="get" action="<?php echo $_SERVER["PHP_SELF"];?>">
  <label for="inputName">Name:</label>
  <input type="text" name="name" id="inputName">
  <input type="submit" value="Submit">
</form>
</body>
```

The POST Method

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

Advantages and Disadvantages of Using the POST Method

- It is more secure than GET because user-entered information is never visible in the URL query string or in the server logs.
- There is a much larger limit on the amount of data that can be passed and one can send text data as well as binary data (uploading a file) using POST.
- Since the data sent by the POST method is not visible in the URL, so it is not possible to bookmark the page with specific query.

Like `$_GET`, PHP provide another superglobal variable `$_POST` to access all the information sent via post method or submitted through an HTML form using the `method="post"`.

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Example of PHP POST method</title>

</head>

<body>

<?php if(isset($_POST["name"]))

{ echo "<p>Hi, " . $_POST["name"] . "</p>"; } ?>

<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">

<label for="inputName">Name:</label>

<input type="text" name="name" id="inputName">

<input type="submit" value="Submit">

</form>

</body>
```

The `$_REQUEST` Variable

PHP provides another superglobal variable `$_REQUEST` that contains the values of both the `$_GET` and `$_POST` variables as well as the values of the `$_COOKIE` superglobal variable.

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Example of PHP $_REQUEST variable</title>

</head>
```

```
<body>

<?php if(isset($_REQUEST["name"]))

{

echo "<p>Hi, " . $_REQUEST["name"] . "</p>";

}

?>

<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">

<label for="inputName">Name:</label>

<input type="text" name="name" id="inputName">

<input type="submit" value="Submit">

</form>

</body>
```

Note: The superglobal variables `$_GET`, `$_POST` and `$_REQUEST` are built-in variables that are always available in all scopes throughout a script.