PHP Registration Form using GET, POST Methods with Example

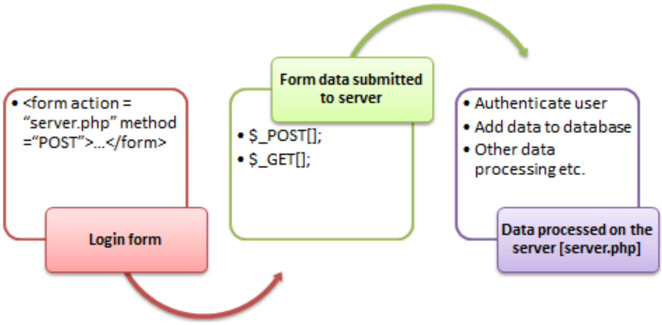
What is Form?

When you login into a website or into your mail box, you are interacting with a form.

Forms are used to get input from the user and submit it to the web server for processing.

The diagram below illustrates the form handling process.

(/images/2013/04/php_forms.png)



• <form action = "server.php" method ="POST">...</form>
Login form
Form data submitted to server
• $_POST[];
• $_GET[];
• Authenticate user
• Add data to database
• Other data processing etc.
Data processed on the server [server.php]

(/images/2013/04/php_forms.png)

A form is an HTML tag that contains graphical user interface items such as input box, check boxes radio buttons etc.

The form is defined using the <form>...</form> tags and GUI items are defined using form elements such as input.

In this tutorial, you will learn-

- When and why we are using forms?
- Create a form
- POST method
- GET method
- GET vs POST Methods
- Processing the registration form data
- More examples

When and why we are using forms?

- Forms come in handy when developing flexible and dynamic applications that accept user input.
- Forms can be used to edit already existing data from the database

Create a form

We will use HTML tags to create a form. Below is the minimal list of things you need to create a form.

- Opening and closing form tags <form>…</form>

PHP Tutorials

5) Arrays (/arrays.html)

6) Control structures (/control-structures-and-loops.html)

7) PHP Loop (/php-loop.html)

8) PHP strings (/php-strings.html)

9) Functions (/functions-in-php.html)

10) PHP forms handling (/php-forms-handling.html)

11) Cookies & sessions (/cookies-and-sessions.html)

12) PHP file processing (/php-file-processing.html)

13) Error handling & exception (/error-handling-and-exceptions.html)

14) Regular expression (/php-regular-expressions.html)

15) PHP mail (/php-mail.html)

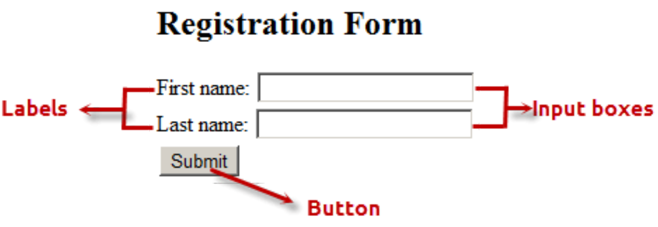
- Form submission type POST or GET
- Submission URL that will process the submitted data
- Input fields such as input boxes, text areas, buttons,checkboxes etc.

The code below creates a simple registration form

```
<html>
<head>
        <title>Registration Form</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>

    <h2>Registration Form</h2>

    <form action="registration_form.php" method="POST"> First name:

        <input type="text" name="firstname"> <br> Last name:

        <input type="text" name="lastname">

        <input type="hidden" name="form_submitted" value="1" />

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

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

Viewing the above code in a web browser displays the following form.

(/images/2013/04/html_form.png)



(/images/2013/04/html_form.png)

HERE,

- <form…>…</form> are the opening and closing form tags
- action="registration_form.php" method="POST"> specifies the destination URL and the submission type.
- First/Last name: are labels for the input boxes
- <input type="text"…> are input box tags
- <br> is the new line tag
- <input type="hidden" name="form_submitted" value="1"/> is a hidden value that is used to check whether the form has been submitted or not
- <input type="submit" value="Submit"> is the button that when clicked submits the form to the server for processing

Submitting the form data to the server

The action attribute of the form specifies the submission URL that processes the data. The method attribute specifies the submission type.

PHP POST method

- This is the built in PHP super global array variable that is used to get values submitted via HTTP POST method.
- The array variable can be accessed from any script in the program; it has a global scope.
- This method is ideal when you do not want to display the form post values in the URL.

- A good example of using post method is when submitting login details to the server.

It has the following syntax.

```
<?php
 $_POST['variable_name'];
?>
```

HERE,

- “$_POST[…]” is the PHP array
- “'variable_name'” is the URL variable name.

PHP GET method

- This is the built in PHP super global array variable that is used to get values submitted via HTTP GET method.
- The array variable can be accessed from any script in the program; it has a global scope.
- This method displays the form values in the URL.
- It’s ideal for search engine forms as it allows the users to book mark the results.

It has the following syntax.

```
<?php
$_GET['variable_name'];
?>
```

HERE,

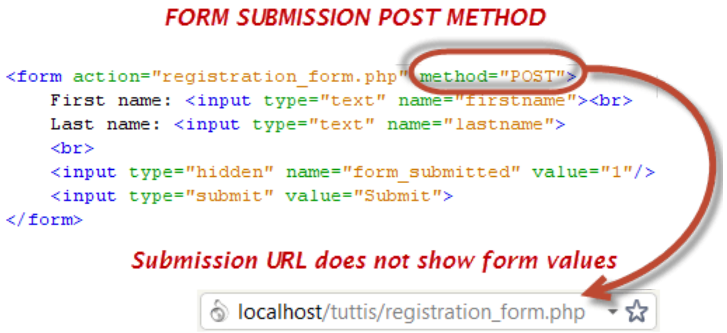
- “$_GET[…]” is the PHP array
- “'variable_name'” is the URL variable name.

GET vs POST Methods

POST	GET
Values not visible in the URL	Values visible in the URL
Has not limitation of the length of the values since they are submitted via the body of HTTP	Has limitation on the length of the values usually 255 characters. This is because the values are displayed in the URL. Note the upper limit of the characters is dependent on the browser.
Has lower performance compared to Php_GET method due to time spent encapsulation the Php_POST values in the HTTP body	Has high performance compared to POST method dues to the simple nature of appending the values in the URL.
Supports many different data types such as string, numeric, binary etc.	Supports only string data types because the values are displayed in the URL
Results cannot be book marked	Results can be book marked due to the visibility of the values in the URL

The below diagram shows the difference between get and post

(/images/2013/04/post_form_submission.png)



FORM SUBMISSION POST METHOD

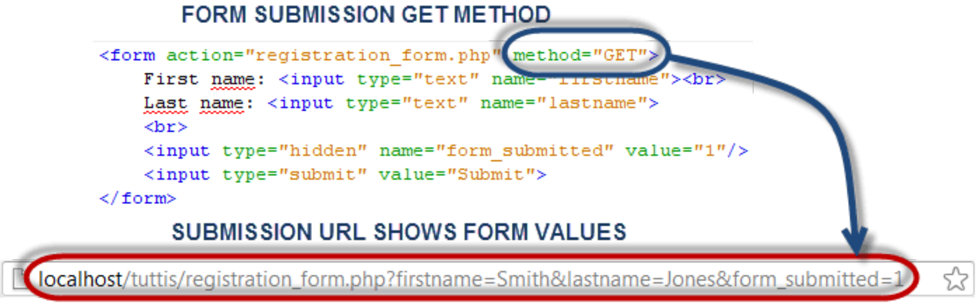
```
<form action="registration_form.php" method="POST">
    First name: <input type="text" name="firstname"><br>
    Last name: <input type="text" name="lastname">
    <br>
    <input type="hidden" name="form_submitted" value="1"/>
    <input type="submit" value="Submit">
</form>
```

Submission URL does not show form values

localhost/tuttis/registration_form.php

(/images/2013/04/post_form_submission.png)

(/images/2013/04/get_form_submission.png)



FORM SUBMISSION GET METHOD

```
<form action="registration_form.php" method="GET">
    First name: <input type="text" name="firstname"><br>
    Last name: <input type="text" name="lastname">
    <br>
    <input type="hidden" name="form_submitted" value="1"/>
    <input type="submit" value="Submit">
</form>
```

SUBMISSION URL SHOWS FORM VALUES

localhost/tuttis/registration_form.php?firstname=Smith&lastname=Jones&form_submitted=1

(/images/2013/04/get_form_submission.png)

Processing the registration form data

The registration form submits data to itself as specified in the action attribute of the form.

When a form has been submitted, the values are populated in the $_POST super global array.

We will use the PHP isset function to check if the form values have been filled in the $_POST array and process the data.

We will modify the registration form to include the PHP code that processes the data. Below is the modified code

```
<html>
<head>
        <title>Registration Form</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>
<body>

    <?php if (isset($_POST['form_submitted'])): ?> //this code is executed when the form is submitted

        <h2>Thank You <?php echo $_POST['firstname']; ?> </h2>

        <p>You have been registered as
            <?php echo $_POST['firstname'] . ' ' . $_POST['lastname']; ?>
        </p>

        <p>Go <a href="/registration_form.php">back</a> to the form</p>

        <?php else: ?>

            <h2>Registration Form</h2>

            <form action="registration_form.php" method="POST">

                 First name:
                <input type="text" name="firstname">

                <br> Last name:
                <input type="text" name="lastname">

                        <input type="hidden" name="form_submitted" value="1" />

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

            </form>

      <?php endif; ? >
</body>
</html>
```

HERE,

- <?php if (isset($_POST['form_submitted'])): ?> checks if the form_submitted hidden field has been filled in the $_POST[] array and display a thank you and first name message.
  If the form_fobmitted field hasn't been filled in the $_POST[] array, the form is displayed.

More examples

Simple search engine

We will design a simple search engine that uses the PHP_GET method as the form submission type.

For simplicity's sake, we will use a PHP If statement to determine the output.

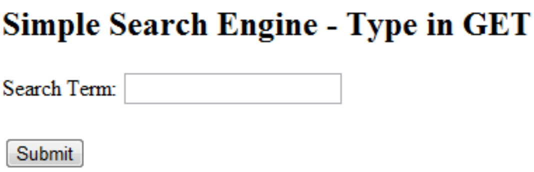
We will use the same HTML code for the registration form above and make minimal modifications to it.

```
<html>
<head>
        <title>Simple Search Engine</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>

    <?php if (isset($_GET['form_submitted'])): ?>

        <h2>Search Results For <?php echo $_GET['search_term']; ?> </h2>
        <?php if ($_GET['search_term'] == "GET"): ?>

            <p>The GET method displays its values in the URL</p>

            <?php else: ?>
                 <p>Sorry, no matches found for your search term</p>

            <?php endif; ?>

                <p>Go <a href="/search_engine.php">back</a> to the form</p>

                <?php else: ?>

                   <h2>Simple Search Engine - Type in GET </h2>

                   <form action="search_engine.php" method="GET">

                        Search Term:
                        <input type="text" name="search_term">
                        <br>

                        <input type="hidden" name="form_submitted" value="1" />

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

                  </form>
               <?php endif; ?>
</body>
</html>
```

View the above page in a web browser

The following form will be shown

(/images/2013/04/simple_search_engine.png)



Simple Search Engine - Type in GET

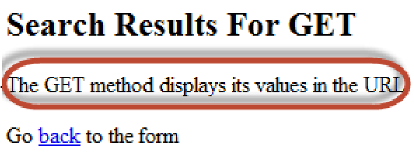
Search Term:

(/images/2013/04/simple_search_engine.png)

Type GET in upper case letter then click on submit button.

The following will be shown

(/images/2013/04/search_engine_results.png)



Search Results For GET

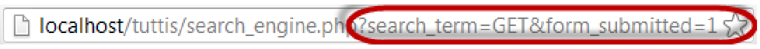
The GET method displays its values in the URL

Go back to the form

(/images/2013/04/search_engine_results.png)

The diagram below shows the URL for the above results

(/images/2013/04/search_engine_url.png)



localhost/tuttis/search_engine.php?search_term=GET&form_submitted=1

(/images/2013/04/search_engine_url.png)

Note the URL has displayed the value of search_term and form_submitted. Try to enter anything different from GET then click on submit button and see what results you will get.

Working with check boxes, radio buttons

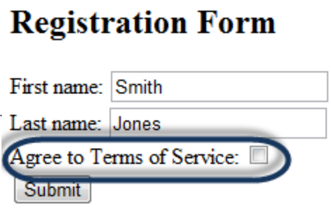
If the user does not select a check box or radio button, no value is submitted, if the user selects a check box or radio button, the value one (1) or true is submitted.

We will modify the registration form code and include a check button that allows the user to agree to the terms of service.

```
<html>
<head>
        <title>Registration Form</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>

    <?php if (isset($_POST['form_submitted'])): ?>

        <?php if (!isset($_POST['agree'])): ?>

            <p>You have not accepted our terms of service</p>

            <?php else: ?>

                <h2>Thank You <?php echo $_POST['firstname']; ?></h2>

                <p>You have been registered as
                    <?php echo $_POST['firstname'] . ' ' . $_POST['lastname']; ?>
                </p>

                <p> Go <a href="/registration_form2.php">back</a> to the form</p>

            <?php endif; ?>

            <?php else: ?>

                        <h2>Registration Form</h2>

                        <form action="registration_form2.php" method="POST">

                            First name:
                            <input type="text" name="firstname">

                            <br> Last name:
                            <input type="text" name="lastname">

                            <br> Agree to Terms of Service:
                            <input type="checkbox" name="agree">
                            <br>

                            <input type="hidden" name="form_submitted" value="1" />

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

                        </form>
        <?php endif; ?>
</body>
</html>
```

View the above form in a browser

(/images/2013/04/registration_form.png)



Registration Form

First name: Smith

Last name: Jones

Agree to Terms of Service: □

(/images/2013/04/registration_form.png)

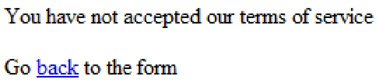
Fill in the first and last names

Note the Agree to Terms of Service checkbox has not been selected.

Click on submit button

You will get the following results

(/images/2013/04/radio_button_false.png)



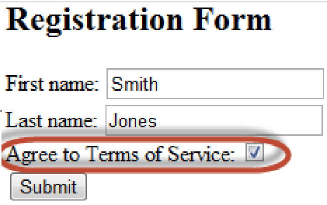
You have not accepted our terms of service

Go back to the form

(/images/2013/04/radio_button_false.png)

Click on back to the form link and then select the checkbox

(/images/2013/04/registration_form2.png)



Registration Form

First name: Smith

Last name: Jones

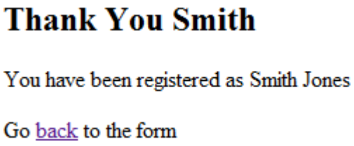
Agree to Terms of Service: ☑

(/images/2013/04/registration_form2.png)

Click on submit button

You will get the following results

(/images/2013/04/radio_button_true.png)



Thank You Smith

You have been registered as Smith Jones

Go back to the form

(/images/2013/04/radio_button_true.png)

Summary

- Forms are used to get data from the users
- Forms are created using HTML tags
- Forms can be submitted to the server for processing using either POST or GET method
- Form values submitted via the POST method are encapsulated in the HTTP body.
- Form values submitted via the GET method are appended and displayed in the URL.

❮ Prev (/functions-in-php.html) **Report a Bug** Next ❯ (/cookies-and-sessions.html)

Top Tutorials

(/selenium-tutorial.html)

Selenium (/selenium-tutorial.html)

(/software-testing.html)

Testing (/software-testing.html)

(/ethical-hacking-tutorials.html)

Hacking (/ethical-hacking-tutorials.html)

(https://www.facebook.com/guru99com/)

(https://twitter.com/guru99com)

(https://www.youtube.com/channel/UC19i1XD6k88KqHlET8atqFQ)

(https://forms.aweber.com/form/46/724807646.html)

About

About Us (/about-us.html)

Advertise with Us (/advertise-us.html)

Write For Us (/become-an-instructor.html)

Contact Us (/contact-us.html)

Career Suggestion

SAP Career Suggestion Tool (/best-sap-module.html)

Software Testing as a Career (/software-testing-career-complete-guide.html)

Certificates (/certificate-it-professional.html)

Interesting

Books to Read! (/books.html)
Blog (/blog/)
Quiz (/tests.html)
eBook (/ebook-pdf.html)

Execute online

Execute Java Online (/try-java-editor.html)
Execute Javascript (/execute-javascript-online.html)
Execute HTML (/execute-html-online.html)
Execute Python (/execute-python-online.html)

(/sap-training-hub.html)

SAP (/sap-training-hub.html)

(/java-tutorial.html)

Java (/java-tutorial.html)

(/python-tutorials.html)

Python (/python-tutorials.html)

(/jmeter-tutorials.html)

Jmeter (/jmeter-tutorials.html)

(/informatica-tutorials.html)

Informatica (/informatica-tutorials.html)

(/jira-tutorial-a-complete-guide-for-beginners.html)

JIRA (/jira-tutorial-a-complete-guide-for-beginners.html)

Privacy Policy (/privacy-policy.html)