

Способы отправки информации на веб-сервер в PHP

Ересь Артём Владимирович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье выделены особенности методов PHP, с помощью которых можно отправлять информацию на веб-сервер, а также продемонстрировано выполнение с помощью них поставленных задач.

Ключевые слова: PHP, информация, клиент, сервер, программирование

Methods of sending information for a Web server to PHP

Yeres Artem Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article features of the PHP methods by means of which it is possible to send information to a Web server are selected and also execution of objectives by means of them is shown.

Keywords: PHP, information, client, server, programming

В процессе разработки проектов одним из важных факторов является возможность ведения диалога с будущим пользователем. Система дает право задавать вопросы и в то же время сама наделена полномочия их предоставлять, так же имеются другие дополнительные разрешения для пользователя в зависимости от тематики определенного проекта.

Для осуществления этого придуманы специальные формы в PHP, использование которых происходит через методы взаимодействия объектов и их параметров.

Целью данной работы является раскрытие основных методов обмена информацией на PHP между системой и пользователем, демонстрация их использования.

Данная работа является не единственной по этой теме. Автор Д.Н. Колисниченко рассмотрел методы и подходы к проектированию и разработке программного обеспечения, дал начальное понимание программирования на PHP в этой сфере [1]. В работе И.Т. Степаненко, Е.В. Степаненко показаны важнейшие аспекты объектно-ориентированного программирования, выявлены предпосылки и причины перехода от традиционного программирования в сфере информационных технологий [2]. Р.Р. Абдулвелеева, Р.И. Абдулвелеев и И.Р. Абдулвелеев в своей статье

продемонстрировали применение программирования на PHP для сферы проектирования, показали возможности программных средств для ускорения и упрощения расчетов инженерии [3]. В интернет источнике представлены особенности форм в PHP, даны варианты их использования, методика и структура [4]. Подробно остановились на способах передачи информации между клиентом и сервером в последнем источнике [5].

Для отправки информации первоначальным действием с ней является кодирование, осуществляемое по принципу URL – адресов. То есть двойные элементы "имя / значение" соединяются символом "=", а остальные делятся через "&". Примером этого можно считать: name2 = value2 & name3 = value3 & name4 = value4.

Разберемся с двумя основными методами обмена информацией – GET и POST.

Особенности GET:

1. Длинная строка отражаемая на сервере и браузере.
2. Максимальное количество символов – 1024.
3. При манипуляциях с секретной и защищенной информацией метод лучше не использовать.
4. Запрет отправки изображений и документов текстовых форматов.
5. Чтобы данные переданный через этот метод стали доступны необходимо воспользоваться средой QUERY_STRING.

Для наглядности приведем простой пример, требующий от пользователя ввод таких данных как имя и возраст:

```
<?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>
```

Рис. 1. Метод Get

В итоге пользователь получит следующее окно:

Name: Age:

Рис. 2. Готовый пример

Для работы методология POST использует заголовки http формата, именно с помощью них идет отправка информации. Подобно первому методу происходит первоначальная обработка информации, а далее данные отправляются в QUERY_STRING.

Особенности POST:

1. Нет ограничений по количеству символов.
2. Отправка множества видов форматов.
3. Лучшая защищенность и безопасность при отправке.

Для примера выполним ту же задачу, что и в GET:

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

Рис. 2. Демонстрация кода примера

В итоге получим тот же ответ:

Name: Age:

Рис. 3. Окно диалога

Существует особая переменная под названием `$_REQUEST` имеющая в своем составе массивы `$_GET` и `$_POST`. Основной ее функцией можно обозначить получение итоговых данных из форм с двумя вышеизложенными методологиями.

Продемонстрируем следующий пример:

```
<?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>
```

Рис. 4. Использование переменной для примера

В `$_PHP_SELF` заложено наименование личного скрипта для которого оно используется. И результатом данной операции будет:



The image shows a simple web form. It consists of two text input fields. The first field is labeled "Name:" and the second is labeled "Age:". To the right of the "Age:" field is a button labeled "Submit". The form is enclosed in a thin border.

Рис. 5. Итог

Таким образом, в данной статье были выявлены особенности методов PHP, для обмена информацией между пользователем и сервером, а также продемонстрирована на примерах методика их использования. Каждый из методов имеет свои достоинства и выбор зависит от ситуации и интересов программиста.

Библиографический список

1. Колисниченко Д. Н. PHP И MYSQL // БХВ-Петербург. 2013. С. 560. URL: <https://elibrary.ru/item.asp?id=21556082> (Дата обращения: 23.08.2018)
2. Степаненко И.Т., Степаненко Е.В. Переход от традиционного к объектно-ориентированному программированию // Современные наукоемкие технологии. 2008. № 1. С. 26. URL: <https://elibrary.ru/item.asp?id=9926657> (Дата обращения: 23.08.2018)
3. Абдулвелеева Р.Р., Абдулвелеев Р.И., Абдулвелеев И.Р. Использование программирования на PHP в инженерных расчетах // Информатизация образования и науки. 2015. № 3 (27). С. 74-83. URL: <https://elibrary.ru/item.asp?id=23601389> (Дата обращения: 23.08.2018)
4. Работа с формами. Метод GET и POST в PHP URL: <http://blog.ox2.ru/php/php-metod-get-i-post/> (Дата обращения: 23.08.2018)
5. Методы передачи данных форм в PHP <http://labs.org.ru/php-8/> (Дата обращения: 23.08.2018)