DOI: 10.17117/na.2016.09.01.448

Поступила (Received): 22.09.2016

<http://ucom.ru/doc/na.2016.09.01.448.pdf>

**Назарцев М.С., Приставка В.А.,
Шельпов А.В., Куприенков В.Д.
Технология разработки многостраничных web-
приложений с межстраничной передачей параметров**

**Nazarcev M.S., Pristavka V.A., Shelpov A.V., Kuprienkov V.D.
Technology of development multipage web-application with cross-
pass parameters**

Одно- или многостраничные динамические сайты, написанные на языке HTML, можно написать и отладить на локальном ПК пользователя с помощью любого браузера. То же самое можно сказать и про динамический сайт, содержащий скрипты, написанные на JavaScript, потому что интерпретатор такого рода скриптов также входит в состав любого браузера

Ключевые слова: HTML, JavaScript, PHP, СУБД, MySQL

One or multipage dynamic sites which writes o HTML, also can be written on local user's PC with helping of any browser. Same about dynamic site which includes scripts, written on JavaScript, because interpreter of the same scripts also includes in any browser

Key words: HTML, JavaScript, MySQL

Назарцев Максим Сергеевич

Преподаватель

Лермонтовский региональный многопрофильный колледж

г. Лермонтов, ул. Комсомольская, 18

Nazarcev Maksim Sergeevich

Teacher

Lermontov regional multidisciplinary college

Lermontov, Komsomolskaya st., 18

Приставка Виктор Александрович

Студент

Лермонтовский региональный многопрофильный колледж

г. Лермонтов, ул. Комсомольская, 18

Pristavka Viktor Alexandrovich

Student

Lermontov regional multidisciplinary college

Lermontov, Komsomolskaya st., 18

Шельпов Алексей Владимирович

Студент

Лермонтовский региональный многопрофильный колледж

г. Лермонтов, ул. Комсомольская, 18

Shelpov Alexey Vladimirovich

Student

Lermontov regional multidisciplinary college

Lermontov, Komsomolskaya st., 18

Куприенков Владимир Дмитриевич

Студент

Лермонтовский региональный многопрофильный колледж

г. Лермонтов, ул. Комсомольская, 18

Kuprienkov Vladimir Dmitrievich

Student

Lermontov regional multidisciplinary college

Lermontov, Komsomolskaya st., 18

Одно- или многостраничные динамические сайты, написанные на языке *HTML*, можно написать и отладить на локальном ПК пользователя с помощью любого браузера. То же самое можно сказать и про динамический, сайт, содержащий скрипты, написанные на *JavaScript*, потому что интерпретатор такого рода скриптов также входит в состав любого браузера.

Совсем по другому дела обстоят в случае необходимости передачи определенных параметров при переходе со страницы на страницу в многостраничном web-приложении. Здесь уже не обойтись без так называемого серверного программирования, т.е. без создания специальных программ, реализуемых исключительно на сервере. Такого рода программы пишутся на специальном языке *PHP*, который не может выполняться браузером, по той простой причине в составе браузера нет интерпретатора языка *PHP*.

Необходимость в межстраничном обмене информацией возникает, например, в том случае, когда web-приложения представлены в виде некой обучающей *Online* методики, пользуясь которой пользователь самостоятельно решает конкретные свои проблемы, например, психологического или организационного плана.

Такая методика в режиме *Online* диалога с пользователем конкретизирует его проблемы, выявляет его потенциальные возможности и формирует, так называемый паспорт, по этапной реализации данной проблемы.

Как правило, структура такого приложения достаточно сложная, а результаты диалога, которые формируются практически на каждой странице web-приложения, должны сохраняться до самого конца работы приложения, так как именно на завершающем этапе осуществляется формирование и выдача рекомендаций, путем анализа параметров, полученных от пользователя на всех предшествующих работы приложения.

Для реализации приложения такой конструкции можно пойти по традиционному пути, создав на сервере небольшую базу данных. К тому же в среде Web-программирования уже имеется готовая и прекрасно себя зарекомендовавшая СУБД *MySQL*, обладающая оптимальным соотношением цены, скорости работы и устойчивости к ошибкам.

База данных *MySQL* состоит из набора таблиц, каждая из которых, в свою очередь, состоит из столбцов и строк, на пересечении которых образуются ячейки. Столбцы, как правило, предназначены для хранения однотипной информации. Одним из главных преимуществ работы с такой базой данных считается простота манипулирования хранящейся в ней информации, путем, так называемых *запросов*, реализуемых путем использования специального языка запросов *SQL*.

Однако, специфика нашего приложения заключается в том, что все передаваемые параметры, которые возникают на той или иной странице нашего приложения, как правило, единичны. Другими словами в процессе формирования в базе данных соответствующей таблицы, она будет состоять из большого числа столбцов и всего одной строки.

Поэтому использование для нашего приложения языка запросов обработки информации в рамках СУБД *MySQL* – это «пальба из пушки по воробьям», так как все преимущества работы с таблицами подобного типа сведены на нет.

Рассмотрим альтернативный и на наш взгляд, более рациональный путь решения этой задачи, воспользовавшись элементом *hidden* (скрытое поле), входящего в состав элементов атрибута *type* тега *<input>*. Тег *<input>* – один из элементов формы, обеспечивающий создание различных элементов интерфейса взаимодействия с пользователем на многостраничных сайтах. Он может передавать различные типы объектов: текстовые поля, кнопки, флажки, переключатели и т.д.

Основным атрибутом тега *<input>* является атрибут *type*, с помощью которого задается вид передаваемого через форму объекта. Элементами атрибута *type* являются: текстовое поле (*text*), поле с паролем (*password*), скрытое поле (*hidden*), поле для отправки файла (*file*) переключатель (*radio*), флажок (*checkbox*), кнопка (*button*) и т.д.

Воспользовавшись элементом *hidden* можно отключить отображение объекта (в том числе и текстового поля) в окне браузера. Это весьма удобный механизм отладки, позволяющий отключать изображение объекта уже после полной отладки данной веб-страницы.

Для решения задачи передачи параметров в многостраничном диалоговом веб-приложении, воспользуемся уже описанными нами замечательными свойствами скрытого поля *hidden*. Для реализации механизма доступа к конкретному элементу контейнера, организованного под эгидой элемента *hidden*, будем после каждого введенного в этот контейнер параметра, заносить туда же и специальный знак-разделитель, например, обратный *slash* (**). Это позволит в дальнейшем разработать простейшую программу на кроссплатформенном языке *PHP*, используемый для генерации динамических *HTML*-страниц на веб-сервере [1].

Для обеспечения корректности динамических операций подстрахуем нашу динамику следующим скриптом, который оформлен в виде одного из атрибутов тега *body*:

```
<body style="background-image :url(images/135.jpg)" onload = "document.getElement  
ById('vibor').focus()">
```

Событие *Onload* используется как указатель на, что веб-страница полностью загружена, включая содержание, изображения, стилевые файлы и внешние скрипты.

Формирование *php*-контейнера начнем с того, что занесем в него метку разделения переменных – обратный слэш. Символ обратный слэш имеет несколько возможных применений в *php*. В первую очередь, если он предшествует не буквенно-цифровому символу, то снимает с него специальное значение. Поэтому для занесения в контейнер самого символа обратный *slash*, необходимо использовать запись `'\'`:

```
<input type = "text/hidden" name="php_container" ID="php_container" size=1 value="\\" ">
```

Атрибут *type* в теге *input* после отладки должен принять значение *type = "hidden"*, однако в процесс отладки очень удобно выводить текущее состояние контейнера на экран – в этом случае необходимо использовать запись *type = "text"*.

Считаем, что макет контейнера у нас уже имеется. Следующая задача – собирать в этот контейнер всю информацию от пользователя, преобразовывать ее в удобную для нас форму, и, в зависимости от результатов этого анализа, продолжать тот или иной вариант диалога с пользователем.

С этой целью разработаем *php*-программу, обеспечивающую обработку *post* запросов, формируемых в режиме диалога с пользователем, с последующим их занесением в наш контейнер. Рассмотрим пример, когда контейнер уже частично заполнен, и необходимо добавить в него еще один параметр [2]:

```
.....
<? php
//Настройка локали
Setlocale (LC_ALL, 'ru_RU.65001', 'rus_RUS.65001', 'Russian_Russia. 65001', 'russian');
//Функция isset позволяет узнать существует ли переменная, указанная в скобках
if (isset($_POST['Punct_3']))
{
    $h_v = $_POST ['php_container']; // Текущее содержание php-контейнера

    //Функция explode возвращает массив строк с разделителем
    $Ar_hidden=explode ("\\",$h_v);

    $t_r = $_POST ['Punct_3']; //Переменная, подготовленная для занесения в контейнер
}
else $t_r="";
?>
.....
```

Вывод параметров, занесенных в контейнер ранее (Пункт №1 и Пункт №2), осуществляется путем обращения к соответствующим строкам массива *\$Ar_hidden* следующим образом: *<?echo \$Ar_hidden[1];?>* и *<?echo\$Ar_hidden[2];?>* соответственно.

Обращение к параметру, сформированному на предшествующей странице можно оформить и так: *<?echo \$t_r;?>*.

```

*****

<table border = 0 align=center><tr><td background=images/31.jpg><b><big><font color =
'blue'> Сформулированные раннее пункты вашего проекта</font></b></td></tr> </table>

<table border = 1 align=center background=images/31.jpg>
<tr><td ><b><font color = 'blue'>Пункт №1. Я хочу: </font></b></td> <td><font color =
'blue'><?echo $Ar_hidden[1];?></font></td></tr>

<tr><td ><b><font color = 'blue'>Пункт №2. Когда: </font></b></td> <td><font color =
'blue'><?echo $Ar_hidden[2];?></font></td></tr>

<tr><td ><b><font color = 'blue'>Пункт №3. Зачем: </font></b></td> <td><font color =
'blue'><?echo $t_r;?></font></td></tr>

</table>

*****

```

Подготовка параметра, формируемого на данной странице сайта и параметра, который необходимо добавить в контейнер вместе с символом разделителем осуществляется следующим образом:

```

*****

<table border = 0 align=center><tr><td background=images/31.jpg><b><big><font color =
'blue'>ПУНКТ №4: </b> Что вы готовы для этого делать? </font></big></td></tr> </table>

<table border = 0 align=center><tr><td background=images/31.jpg><b>Ваш ответ (не
менее 5 символов) поместите сюда:</b></td></tr> </table>

<!--Ответ пользователя -->

<textarea name="Punct_4" Id=" Punct_4" cols=70 rows=7 ></textarea>

*****

<input name="php_container" ID="php_container" size=1 value="<?echo $h_v.t_r."<\">";?>">

*****

```

Содержание контейнера, сформированного к моменту перехода на следующую страницу сайта, представлено ниже.

```

*****

<input name="php_container" ID="php_container" size=1 value="\Получить высшее образо-
вание\В ближайшие 5 лет\От этого зависит успех моей карьеры">

*****

```


На рисунке показан фрагмент 5 страницы динамического web-приложения со всеми параметрами, сформированными ранее на страницах 1-4 в режиме диалога с пользователем [3].

Рис. 1. Фрагмент страницы 5 динамического web-приложения

Список используемых источников:

1. Иван И.А. О коучинге просто. Программирование Web-приложений. URL: <http://prosto-coach.ru/>
2. Шкрыль А.А. PHP – это просто. Програмируем для Web-сайта. СПб.: БХВ-Петербург, 2006. 368 с.
3. Электронный коуч. Рубрика «Исполнитель желаний». URL: http://prosto-coach.ru/category/el_coach/

© 2016, Назарцев М.С., Приставка В.А., Шельпов А.В.,
Куприенков В.Д.

Технология разработки многостраничных web-
приложений с межстраничной передачей
параметров

© 2016, Nazarcev M.S., Pristavka V.A., Shelpov A.V.,
Kuprienkov V.D.

Technology of development multipage web-
application with cross-pass parameters