

Web-разработка на PHP-технологиях

Курс лекций и семинаров для студентов,
желающих научиться основам Web-разработки на PHP

Осень-Зима 2014

Лекция №6

PHP и Cookies. Сессии в PHP.

Автор: Дмитрий Левин, Senior PHP Developer, Sibers

Sibers®

PHP и Cookies

Cookies — это механизм хранения данных браузером удаленного компьютера для идентификации возвращающихся посетителей и хранения параметров веб-страниц (например, переменных)

Файлы Cookies представляют собой обыкновенные текстовые файлы, которые хранятся на диске у посетителей сайтов. Файлы Cookies и содержат ту информацию, которая была в них записана сервером.

PHP и Cookies

Для установки Cookies используется функция **SetCookie()**.

Для этой функции можно указать шесть параметров, один из которых является обязательным:

- **name** — задает имя (строка), закрепленное за Cookie;
- **value** — определяет значение переменной (строка);
- **expire** — время «жизни» переменной (целое число). Если данный параметр не указать, то Cookie будут «жить» до конца сессии, то есть до закрытия браузера. Если время указано, то, когда оно наступит, Cookie самоуничтожится.
- **path** — путь к Cookie (строка);
- **domain** — домен (строка). В качестве значения устанавливается имя хоста, с которого Cookie был установлен;
- **secure** — передача Cookie через защищенное HTTPS-соединение.

PHP и Cookies

При использовании Cookies необходимо иметь в виду, что Cookies должны устанавливаться до первого вывода информации в браузер (например, оперетором echo или выводом какой-либо функции). Поэтому желательно устанавливать Cookies в самом начале скрипта.

Cookies устанавливаются с помощью определенного заголовка сервера, а если скрипт выводит что-либо, то это означает, что начинается тело документа.

В результате Cookies не будут установлены и может быть выведено

```
<?php

// Устанавливаем Cookie до конца сессии:
SetCookie("Test", "Value");

// Устанавливаем Cookie на один час после установки:
SetCookie("My_Cookie", "Value", time()+3600);
```

PHP и Cookies

Чтение значений Cookies

Получить доступ к Cookies и их значениям достаточно просто. Они хранятся в суперглобальных массивах и `$_COOKIE` и `$HTTP_COOKIE_VARS`.

```
<?php
// Устанавливаем Cookie 'test' со значением 'Hello' на один час:
setcookie("test","Hello",time()+3600);
// При следующем запросе скрипта выводит 'Hello':
echo @$_COOKIE['test'];
?>
```

При чтении значений Cookies обращайте внимание на проверку существования Cookies, например, используя оператор `isset()`

PHP и Cookies

Удаление Cookies

Иногда возникает необходимость удаления Cookies. Сделать это несложно, необходимо лишь вновь установить Cookie с идентичным именем и пустым параметром

```
<?php
// Удаляем Cookie 'Test':
SetCookie("Test", "");
?>
```

PHP и Cookies

Установка массива Cookies и его чтение

Мы можем установить массив Cookies, используя квадратные скобки в именах Cookies [], а затем прочитать массив Cookies и значения этого массива:

```
<?php
// Устанавливаем массив Cookies:
setcookie("cookie[1]", "Первый");
setcookie("cookie[2]", "Второй");
setcookie("cookie[3]", "Третий");

// После перезагрузки страницы мы отобразим
// Состав массива Cookies 'cookie':
if (isset($_COOKIE['cookie'])) {
    foreach ($_COOKIE['cookie'] as $name => $value) {
        echo "$name : $value <br>";
    }
}
?>
```

PHP и Cookies

Преимущества использования Cookies неоспоримы. Однако существуют и некоторые проблемы их использования.

Первая из них заключается в том, что посетитель может блокировать прием Cookies браузером или попросту удалить все Cookies или их часть.

Таким образом, мы можем иметь некоторые проблемы в идентификации таких посетителей

Сессии в PHP

Сессии в PHP

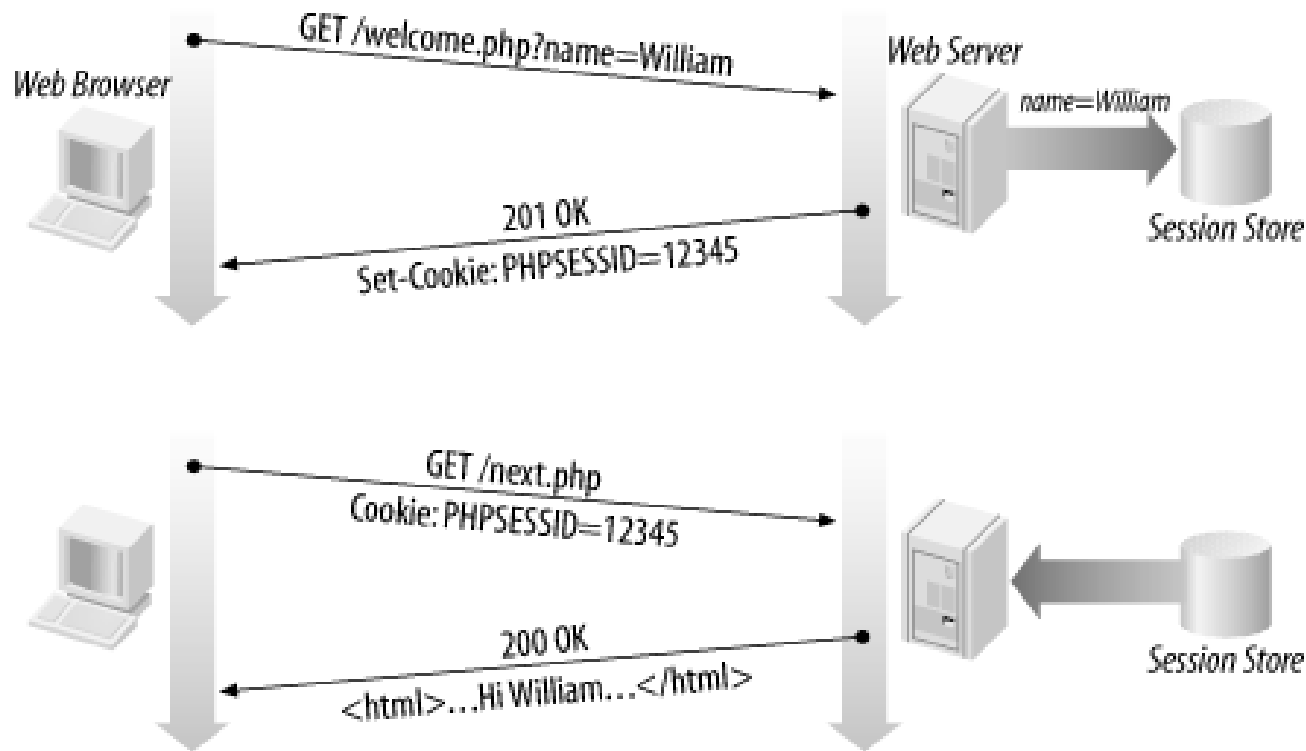
Возможно, вы знаете, что веб-сервер не поддерживает постоянного соединения с клиентом, и каждый запрос обрабатывается, как новый, без связи с предыдущими.

То есть, нельзя ни отследить запросы от одного и того же посетителя, ни сохранить для него переменные между просмотрами отдельных страниц. Вот для решения этих двух задач и были изобретены сессии.

Собственно, сессии, если в двух словах — это механизм, позволяющий однозначно идентифицировать браузер и создающий для этого браузера файл на сервере, в котором хранятся переменные сеанса.

Для начала надо как-то идентифицировать браузер. Для этого надо выдать ему уникальный идентификатор и попросить передавать его с каждым запросом.

Сессии в PHP



Сессии в PHP

Массив `$_SESSION` — особенный. В нем, собственно, и находятся переменные, которые мы ходим сделать доступными в различных скриптах.

Чтобы поместить переменную в сессию, достаточно присвоить ее элементу массива `$_SESSION`.

```
session_start();  
$_SESSION['test']='Hello world!';
```

```
<?  
session_start();  
if (!isset($_SESSION['counter'])) $_SESSION['counter']=0;  
echo "Вы обновили эту страницу ".$_SESSION['counter']++." раз. ";  
echo "<br><a href=".$_SERVER['PHP_SELF']."'>обновить";
```

Сессии в PHP

Для того, чтобы иметь доступ к переменным сессии на любых страницах сайта, необходимо вызвать функцию старта сессий в самом начале файла, в котором нам нужны сессии.

```
session_start();
```

Удаление переменных из сессии

```
unset($_SESSION['var']);
```



Заключение:

В шестой лекции мы рассмотрели Cookies и функцию для их установки, познакомились с сессиями в PHP

В следующей лекции:

PHP – Основы ООП