

Извлекаем товары из интернет магазина с помощью AJAX и jQuery.

Недавно для освоения AJAX на jQuery придумал себе не сложную задачу. Есть элемент формы select содержащий в себе список категорий товаров. Когда выбираем категорию, под select с категориями появляется select с названиями товаров, выбирая товар из списка ниже появится его описание. Честно сказать, задача на практике не нужная, поскольку такие вещи не индексируются поисковиками, однако для понимания и освоения технологии AJAX очень даже неплохо.

Итак, начнем. Первое, что нам необходимо – таблицы в базе данных MySQL, первая – список категорий, вторая – список товаров, третья – связь товаров с категориями. Зачем так много, если можно обойтись двумя таблицами – спросите вы. Ответ прост: некоторые товары могут принадлежать сразу к нескольким категориям. Когда-то давно я делал так: в таблице, отвечающей за товары, есть ячейка содержащая идентификатор. Однако такая реализация описывает жесткую связь товар-категория. Если же нужно поместить товар в несколько категорий, нужна дополнительная таблица.

Следующее. Как вы (я очень на это надеюсь) знаете, приложения AJAX не работают без так называемого backend-а. Это PHP файл, к которому обращается AJAX приложение. Нам понадобится всего два файла: products.php - реализует вывод selecta товаров, и product.php – вывод описания товара. Основной файл, назовем его, например, ajax_1.php, будет содержать в себе JavaScript функции для изменения списков и описаний товаров. Вот вроде и все. Сразу стоит оговорится: я не буду приводить примеры как выбрать категории или товары из таблицы MySQL, если вас это ставит в затруднительное положение, обратитесь к другим статьям.

Начнем с кода файла ajax_1.php:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>AJAX – вывод товаров</title>
</head>
<body>
<?php
echo '
//печатаем начало элемента <select>
<select name="category_id" onchange="show_products(this.value)">
<option value="0">выберите категорию</option>
';
//выводим список категорий:
foreach ($categoryes as $category) {
echo '<option value="'. $category['category_id']. "'>'.stripslashes($category ['category_name']).'</option>';
}
echo '
</select>
';
?>
<div id="prod_select" style="display:none; background-color:#EEEEEE; padding:4px; margin:5px;"></div>
<div id="prod_details" style="display:none; background-color:#EEEEEE; padding:4px; margin:5px;"></div>
</body>
</html>
```

В скрытых слоях с id prod_select и prod_details будут отображаться select товаров и описание выбранного соответственно. Обратите внимание на атрибут onchange списка категорий, в нем вызывается функция show_products() которая принимает значение идентификатора категории. Если не очень понятно с выводом категорий, массив \$categoryes имеет структуру:

```
$categoryes=array(
1=>array('category_id'=>'идентификатор_категории_1', 'category_name'=>'название_категории_1'),
```

```
2=>array('category_id'=>'идентификатор_категории_2', 'category_name'=>'название_категории_2'),
//и т.д.
);
```

Итак, у нас есть список категорий в элементе select, следующим шагом нужна JavaScript функция, которая будет осуществлять:

```
function show_products(category_id) {
//на всякий случай, вдруг что написано в слое prod_select
$('#prod_select').html("");
//то же самое для prod_details
$('#prod_details').html("");
//если у нас уже было выведено описание какого-либо товара, скрываем его:
$('#prod_details').hide('fast')
//самое интересное. Посылаем запрос к файлу products.php методом post, передаем параметр
//category_id равный переменной category_id, которая передается функции. Когда запрос выполнен
//вызываем функцию onAjaxSuccess (можете назвать ее по другому, от этого суть не изменится.)
$.post( "products.php", {category_id: category_id}, onAjaxSuccess);
function onAjaxSuccess(data)
{
$('#prod_select').html(data);
$('#prod_select').show('fast');
}
}
```

Итак, метод .post принимает параметры:

имя backend-файла

параметры, указанные в фигурных скобках (их может быть сколько угодно), записываются они как пары ИМЯ: ЗНАЧЕНИЕ через запятую

Функция, которая будет выполнена по окончании запроса. Я еще до конца не разобрался, но как я понял, ей передается html код (в отличии от приложений на JsHttpRequest, где AJAX запрос возвращает объект)

Сама функция onAjaxSuccess всего лишь добавляет в слой с id prod_select полученный html код, и показывает слой prod_select методом .show().

Теперь перейдем к самому backend файлу products.php. Его код не сильно отличается от php-кода файла ajax_1.php, за исключением отсутствия в нем тегов <html>, <body> и т.п.:

Код файла products.php:

```
//переменная $cat_id равна переменной массива post category_id, которую передает AJAX запрос
$cat_id=$_POST['category_id'];
//Здесь будет запрос к базе данных, что то вроде:
//SELECT * FROM products as pr, category_xref as cx WHERE
//cx.product_id=pr.product_id
echo '
<select name="product_id" onchange="product_details(this.value)">
<option value="0">выберите товар</option>
';
foreach ($products as $product) {
echo '<option value="'. $product['product_id']. ">'. $product['product_name']. '</option>';
}
echo '
</select>
';
```

Код очень похож на вывод selecta категорий. Вот и все, попробуйте выбрать категорию, и вы увидите как появится слой prod_select в котором будет список товаров выбранной категории.

Полностью аналогично добавляем еще одну функцию, которая вызывается при выборе товара, product_details() и принимает идентификатор товара:

```
function product_details(product_id) {
$.post( "product.php", {product_id: product_id}, onAjaxSuccess);
function onAjaxSuccess(data)
{
$('#prod_details').html(data);
$('#prod_details').show('fast')
}
}
```

Здесь все полностью аналогично выводу списка товаров. Сам backend файл product.php выглядит следующим образом:

```
$product_id=$_POST['product_id'];
//извлекаем товар из базы данных, что то вроде:
//SELECT * FROM products WHERE product_id='$product_id'
echo '
<h1>'.$product['product_name'].'</h1>
<p>'.$product['product_desc'].'</p>
';
```

Вот и все, осталось включить в файл ajax_1.php наши функции, и не забыть подключить саму библиотеку jQuery:

```
<?php
Здесь подключение всех необходимых файлов, для работы с MySQL и выбор категорий.
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>AJAX – вывод товаров</title>
<script src="jquery.js" type="text/javascript"></script>
<script type="text/javascript">
function show_products(category_id) {
$('#prod_select').html("");
$('#prod_details').html("");
$('#prod_details').hide('fast')
$.post( "backend/products.php", {category_id: category_id}, onAjaxSuccess);
function onAjaxSuccess(data)
{
$('#prod_select').html(data);
$('#prod_select').show('fast');
}
}
function product_details(prod_id) {
$.post( "backend/product.php", {product_id: prod_id}, onAjaxSuccess);
function onAjaxSuccess(data)
{
$('#prod_details').html(data);
$('#prod_details').show('fast')
}
}
```

```
}
</script>
</head>
<body>
<?php
echo '
//печатаем начало элемента <select>
<select name="category_id" onchange="show_products(this.value)">
<option value="0">выберите категорию</option>
';
//выводим список категорий:
foreach ($categories as $category) {
echo '<option value="'. $category['category_id']. "'>'.stripslashes($category ['category_name']).'</option>';
}
echo '
</select>
';
?>

<div id="prod_select" style="display:none; background-color:#EEEEEE; padding:4px; margin:5px;"></div>
<div id="prod_details" style="display:none; background-color:#EEEEEE; padding:4px; margin:5px;"></div>
</body>
</html>
```