

Отправка данных в формате JSON с помощью JavaScript и jQuery

Владимир | 6 мая, 2009 | [Ajax](http://www.simplecoding.org/category/aj) (<http://www.simplecoding.org/category/aj>), [JavaScript](http://www.simplecoding.org/category/js) (<http://www.simplecoding.org/category/js>), [PHP](http://www.simplecoding.org/category/p) (<http://www.simplecoding.org/category/p>), [Web разработка](http://www.simplecoding.org/category/web-razrabotka) (<http://www.simplecoding.org/category/web-razrabotka>).

Твитнуть



Нравится



В прошлых статьях я рассказывал и приводил примеры получения и обработки данных в формате **JSON** на стороне клиента. Но примера отправки данных серверу в этом формате не было. Сегодня я попробую исправить это упущение.

Прежде всего, кратко напомним **принцип передачи данных в JSON формате от сервера браузеру**.

- 1) Серверный ([PHP](http://www.simplecoding.org/yazyk-programmirovaniya-php) (<http://www.simplecoding.org/yazyk-programmirovaniya-php>)) скрипт формирует массив с данными, преобразует их в **формат JSON** (используется функция `json_encode`) и отправляет браузеру.
- 2) На стороне браузера [JS](http://www.simplecoding.org/javascript) (<http://www.simplecoding.org/javascript>) скрипт обрабатывает полученные данные. Тут для преобразования данных используется `eval`.

То есть все преобразования данных сводятся к использованию двух функций.

Отправка данных серверу немного сложнее.

Прежде всего, нужно четко понимать, что отправить данные в формате JSON можно в параметрах запроса (GET или POST), т.е. точно также как и любые другие данные.

Т.к. формат JSON представляет собой обычную строку, то **алгоритм отправки** будет следующим.

- 1) Формируем JavaScript объект с данными, которые нужно отправить, например, со значениями, введенными пользователем в поля формы.
- 2) Преобразовываем этот объект в JSON формат.
- 3) Отправляем запрос серверу. В одном из параметров запроса передаем строку с JSON данными.
- 4) На стороне сервера читаем переданные данные (они придут в массиве `$_GET` или `$_POST`).
- 5) С помощью `json_decode` преобразуем JSON данные в массив.

Теперь **рассмотрим небольшой пример**.

Допустим, у нас есть страница с формой, которая содержит 2 поля: `field1` и `field2`. Необходимо отправить эти данные в формате JSON серверному скрипту и прочитать его ответ.

Разметку формы я приводить не буду, она самая обычная и в конце статьи я дам ссылку на архив с примером.

Сразу перейдем к JavaScript коду.

```
1  $(function() {  
2      $("#myForm").submit(function() {  
3          var formData = {  
4              "field1":$("#field1").val()  
5              , "field2":$("#field2").val()  
6          };  
7          $.ajax({  
8              url:'dataparser.php'  
9              , type:'POST'  
10             , data:'jsonData=' + $.toJSON(formData)  
11             , success: function(res) {  
12                 alert(res);  
13             }  
14         });  
15         return false;  
16     });  
17 });
```

В этом примере я использовал библиотеку [jQuery \(http://jquery.com/\)](http://jquery.com/) и плагин [jquery-json \(http://code.google.com/p/jquery-json/\)](http://code.google.com/p/jquery-json/). Этот плагин значительно упрощает преобразование данных в JSON формат, т.е. сводит всю работу к вызову одного метода `$.toJSON(...)`.

Рассмотрим подробнее этот пример.

Прежде всего, мы **назначили обработчик** события `submit` для формы. Обратите внимание, что этот обработчик возвращает `false` (строка 15). Таким образом, мы предотвращаем отправку данных формы обычным способом.

В строках 3-6 мы **формируем JavaScript объект** и заполняем его введенными в форму данными.

Затем **отправляем AJAX запрос** (строка 7). В параметре `url` указываем название PHP скрипта, который выполняет обработку данных. В параметре `type` – способ передачи данных.

Третий параметр (`data`) самый интересный. Здесь указываем имя параметра, в котором будут отправлены данные (`jsonData`), и сами данные (преобразуем JavaScript объект в JSON строку с помощью `$.toJSON()`).

После этого, в параметре `success` указываем функцию, которая будет обрабатывать полученные данные.

В результате серверу будет отправлен только один параметр

jsonData {«field1»: «111», «field2»: «222»}

Теперь **рассмотрим серверный скрипт**.

```
1  <?php  
2  $data = json_decode($_POST['jsonData']);  
3  $response = 'Получено параметров '.count($data).'\n';  
4  foreach ($data as $key=>$value) {  
5      $response .= 'Параметр: '.$key.'; Значение: '.$value.'\n';  
6  }  
7  echo $response;  
8  ?>
```

Я решил его сделать максимально простым, т.е. опустил все возможные проверки.

Чтение и преобразование данных выполняется одной строчкой кода (строка 2). В результате получаем массив (`$data`) с данными формы.

После этого скрипт формирует строку с этими же данными и отправляет её назад браузеру.

Как видите, пример получился довольно простой.

Если хотите поэкспериментировать с этим примером, качайте архив.