

# Простой запрос Ajax с помощью jQuery и PHP

Когда я начинал использовать [[JavaScript]] и [[jQuery]], то был разочарован отсутствием простых примеров использования [[Ajax]] в [[jQuery]].

Нужно понимать, что Ajax-запрос такой же, как и любой другой запрос. Единственным его отличием является то, что пользователю не нужно перезагружать страницу. Например, это позволит пользователю отправить данные формы без перехода на другую веб-страницу.

## Пример запроса GET, отправленного с помощью jQuery и Ajax:

```
$.ajax({
  type: "GET",
  url: 'test.php',
  success: function(data){
    alert(data);
  }
});
```

В примере кода передаются три параметра:

1. `type`: тип HTTP запроса. В этом примере я отправляю запрос GET. Если вы хотите отправить запрос POST, измените "GET" на "POST".
2. `url`: адрес, на который вы хотите отправить Ajax запрос. В нашем случае это страница "test.php". Помните, что URL-адрес указывается относительно текущей страницы.
3. `success`: функция, которая вызывается, если запрос был успешным. Она принимает параметр `data`, который будет содержать вывод страницы test.php. То есть, если test.php выводит строку "OK", то параметр `data` будет содержать строку "OK".

Поэкспериментируйте с приведённым выше кодом. Например, замените "GET" на "POST" и измените URL-адрес. Можно использовать инструменты разработчика, встроенные в Firefox / Chrome для отладки Ajax-запросов. Они доступны на вкладке «Сеть». Инструменты разработчика позволяют визуализировать запрос.

## Что если мы захотим добавить параметры запроса GET к Ajax-запросу?

```
$.ajax({
  type: "GET",
  url: 'test.php',
  data: {name: 'Wayne'},
  success: function(data){
    alert(data);
  }
});
```

Я добавил новый параметр data. Это объект JavaScript, содержащий данные, которые передаем в запросе. Поскольку мы отправляем запрос GET, эти параметры будут автоматически добавлены к строке запроса: test.php?name=Wayne

## Пример добавления нескольких параметров GET:

```
$.ajax({
  type: "GET",
  url: 'test.php',
  data: {name: 'Wayne', age: 27, country: 'Ireland'},
  success: function(data){
    alert(data);
  }
});
```

Приведенный выше код отправит запрос GET к test.php?name=Wayne&age=27&country=Ireland

Поэкспериментируйте с кодом, чтобы разобраться в нём.

Попробуйте добавлять и удалять параметры, чтобы понять, как отправлять данные с помощью jQuery и Ajax.

А если мы хотим отправить запрос POST к файлу `submission.php`?

```
$.ajax({
    type: "POST",
    url: 'submission.php',
    data: {name:'Wayne', age: 27},
    success: function(data){
        alert(data);
    }
});
```

Я изменил тип метода (с GET на POST) и URL-адрес (на `submission.php`). В этом примере параметры `name` и `age` будут отправлены как переменные POST. Это означает, что на странице `submission.php` их можно получить с помощью следующего кода:

```
$name = $_POST['name'];
$age = $_POST['age'];
```

Если использовать запрос GET, то переменные будут отправлены через строку запроса. Обратиться к ним можно следующим образом:

```
$name = $_GET['name'];
$age = $_GET['age'];
```

Последней частью паззла является функция `success()`. Она вызывается, если Ajax-запрос завершается успешно. В примере, приведенном выше, я вывожу результат в сообщении. Но в большинстве случаев вы будете использовать эту функцию для отображения уведомлений на странице. Например, если пользователь отправляет форму с помощью Ajax, тогда эта функция применяется для проверки ответа. А также для вывода сообщения о том, что данные формы успешно отправлены на сервер.

Надеюсь, эта статья помогла вам разобраться с основами Ajax-запросов и теперь задачи практикума, где надо вводить исходные данные Вы легко решите!