

# AJAX

ASYNCHRONOUS JAVASCRIPT AND XML



## Что такое AJAX?

AJAX – технология обращения к серверу без перезагрузки страницы.

## Что я могу сделать с помощью AJAX?

- Взаимодействие с элементами интерфейса
- Динамическая подгрузка данных
- Живой поиск

# AJAX

```
var xhr = new XMLHttpRequest();  
xhr.open(method, URL, async?, user?, password?);  
xhr.send(body?);  
//xhr.abort(); Прерывает выполнение запроса
```

# AJAX

```
xhr.open(method, URL, async?, user?, password?);
```

**method** – HTTP-метод. Как правило, используется GET либо POST, хотя доступны и более экзотические, вроде DELETE/PUT и т.п.

**URL** – адрес запроса.

**async** – если установлено в false, то запрос производится синхронно, если true – асинхронно.

**user, password** – логин и пароль для HTTP-авторизации, если нужны.



## Ответ: `status`, `statusText`, `responseText`

### `status`

HTTP-код ответа: 200, 404, 403 и так далее. Может быть также равен 0, если сервер не ответил или при запросе на другой домен.

### `statusText`

Текстовое описание статуса от сервера: *OK*, *Not Found*, *Forbidden* и так далее.

### `responseText`

Текст ответа сервера.

Обычно используют JSON. То есть, сервер возвращает JSON в виде текста, который браузер превращает в объект  
вызовом **`JSON.parse(xhr.responseText)`**.

# AJAX

```
"{"id":21262,"group":1610,"name":"Seitmagambet Olzhas"}"
```

```
JSON.parse(xhr.responseText);
```

```
{  
  id: 21262,  
  group: 1610,  
  name: "Seitmagambet Olzhas",  
}
```

## Событие `readystatechange`

Событие **`readystatechange`** происходит несколько раз в процессе отсылки и получения ответа. При этом можно посмотреть «текущее состояние запроса» в свойстве **`xhr.readyState`**.

- 0 // начальное состояние
- 1 // вызван `open`
- 2 // получены заголовки
- 3 // загружается тело (получен очередной пакет данных)
- 4 // запрос завершён

Запрос проходит их в порядке  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow 3 \rightarrow 4$ , состояние 3 повторяется при каждом получении очередного пакета данных по сети

# AJAX

```
var xhr = new XMLHttpRequest();
xhr.open(method, URL, async, user, password);
xhr.send(body);

xhr.onreadystatechange = function() {
    if (xhr.readyState !== 4) return;

    if (xhr.status !== 200) {
        console.log(xhr.status + ': ' + xhr.statusText);
    } else {
        console.log(JSON.parse(xhr.responseText));
    }
};
```



# AJAX

+ fetch (современная замена XMLHttpRequest)

 **Сторонние библиотеки**

+ jQuery AJAX

+ axios

# AXIOS

```
axios({  
  method: 'post',  
  url: '/offers/create',  
  data: formData  
});
```

## HTTP Method Aliases

**axios.post(url[, data[, config]])**

**axios.get(url[, config])**

...



## Результат запроса

```
{
  // `data` is the response that was provided by the server
  data: {},

  // `status` is the HTTP status code from the server response
  status: 200,

  // `statusText` is the HTTP status message from the server response
  statusText: 'OK',

  // `headers` the headers that the server responded with
  // All header names are lower cased
  headers: {},

  // `config` is the config that was provided to `axios` for the request
  config: {},

  // `request` is the request that generated this response
  // It is the last ClientRequest instance in node.js (in redirects)
  // and an XMLHttpRequest instance the browser
  request: {}
}
```

# AXIOS

```
axios.get('/offers/1')  
  .then(function(response) {  
    console.log(response.data);  
  });
```

```
/*
```

```
...
```

```
*/
```

```
var formData = new FormData(offerForm);
```

```
axios.post('/offers/create', formData)  
  .then(function(response) {  
    console.log(response.data);  
  });
```

# AJAX

## Материалы



### Документация

<https://learn.javascript.ru/ajax>

<https://github.com/axios/axios>

<https://developer.mozilla.org/ru/docs/Web/API/XMLHttpRequest>



### Код

<https://github.com/winexy/laravel-ajax>