

AJAX

Asynchronous Javascript And Xml

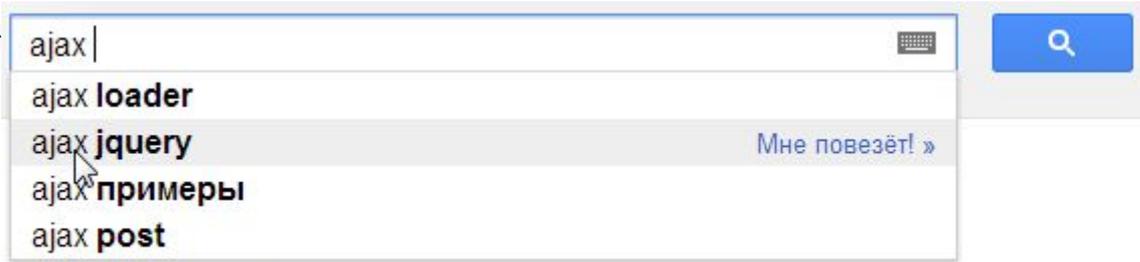
Что такое AJAX?

AJAX (аббревиатура от «Asynchronous Javascript And Xml») – технология обращения к серверу без перезагрузки страницы.

Несмотря на то, что в названии технологии присутствует буква X (от слова XML), использовать XML вовсе не обязательно. Под AJAX подразумевают любое общение с сервером без перезагрузки страницы, организованное при помощи JavaScript.

Что я могу сделать с помощью AJAX?

- элементы интерфейса
- динамическая подгрузка данных

- A screenshot of a search bar with a dropdown menu. The search bar contains the text 'ajax |' and a search icon. The dropdown menu lists several search results: 'ajax loader', 'ajax jquery' (highlighted), 'ajax примеры', and 'ajax post'. A blue button with a search icon is located to the right of the search bar. Below the dropdown menu, there is a text prompt: 'Для поиска нажмите "Ввод"'.

ajax |

ajax loader

ajax **jquery** [Мне повезёт! »](#)

ajax **примеры**

ajax **post**

Для поиска нажмите "Ввод"

XMLHttpRequest

```
1 // 1. Создаём новый объект XMLHttpRequest
2 var xhr = new XMLHttpRequest();
3
4 // 2. Конфигурируем его: GET-запрос на URL 'phones.json'
5 xhr.open('GET', 'phones.json', false);
6
7 // 3. Отсылаем запрос
8 xhr.send();
9
10 // 4. Если код ответа сервера не 200, то это ошибка
11 if (xhr.status != 200) {
12     // обработать ошибку
13     alert( xhr.status + ': ' + xhr.statusText ); // пример вывода: 404: Not Found
14 } else {
15     // вывести результат
16     alert( xhr.responseText ); // responseText -- текст ответа.
17 }
```

XMLHttpRequest

Синтаксис:

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

Задаёт основные параметры запроса:

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

`URL` – адрес запроса. Можно использовать не только http/https, но и другие протоколы, например `ftp://` и `file://`.

При этом есть ограничения безопасности, называемые «Same Origin Policy»: запрос со страницы можно отправлять только на тот же `протокол://домен:порт`, с которого она пришла. В следующих главах мы рассмотрим, как их можно обойти.



JSON Server <https://github.com/typicode/json-server>

Create a `db.json` file

```
{
  "posts": [
    { "id": 1, "title": "json-server", "author": "typicode" }
  ],
  "comments": [
    { "id": 1, "body": "some comment", "postId": 1 }
  ],
  "profile": { "name": "typicode" }
}
```

Start JSON Server

```
$ json-server --watch db.json
```

Now if you go to <http://localhost:3000/posts/1>, you'll get

```
{ "id": 1, "title": "json-server", "author": "typicode" }
```

JSON Server <https://github.com/typicode/json-server>

Install

```
$ npm install -g json-server
```

Routes

Based on the previous `db.json` file, here are all the default routes. You can also add [other routes](#) using `--routes .`

Plural routes

```
GET    /posts
GET    /posts/1
POST   /posts
PUT    /posts/1
PATCH /posts/1
DELETE /posts/1
```

Задание

Сделай страницу со списком городов, которые можно редактировать, удалять и добавлять новые. При ввода города должна работать фильтрация, то есть если набрать например «Сан», появляется «Санкт-Петербург». Задание должно быть реализовано с использованием уаrn, json-server, ajax

Город	
Москва	<input type="button" value="Изменить"/> <input type="button" value="Удалить"/>
Ростов-на-Дону	<input type="button" value="Изменить"/> <input type="button" value="Удалить"/>
Санкт-Петебург	<input type="button" value="Изменить"/> <input type="button" value="Удалить"/>

Вопросы



Dmitry Anikin

CTO of Roonyx

E-mail: dima@roonyx.tech

GitHub: <https://github.com/d-anikin>