

# — JAVASCRIPT —

*HTTP. AJAX. XMLHttpRequest*

# HTTP

*HyperText Transfer Protocol* — «протокол передачи гипертекста»

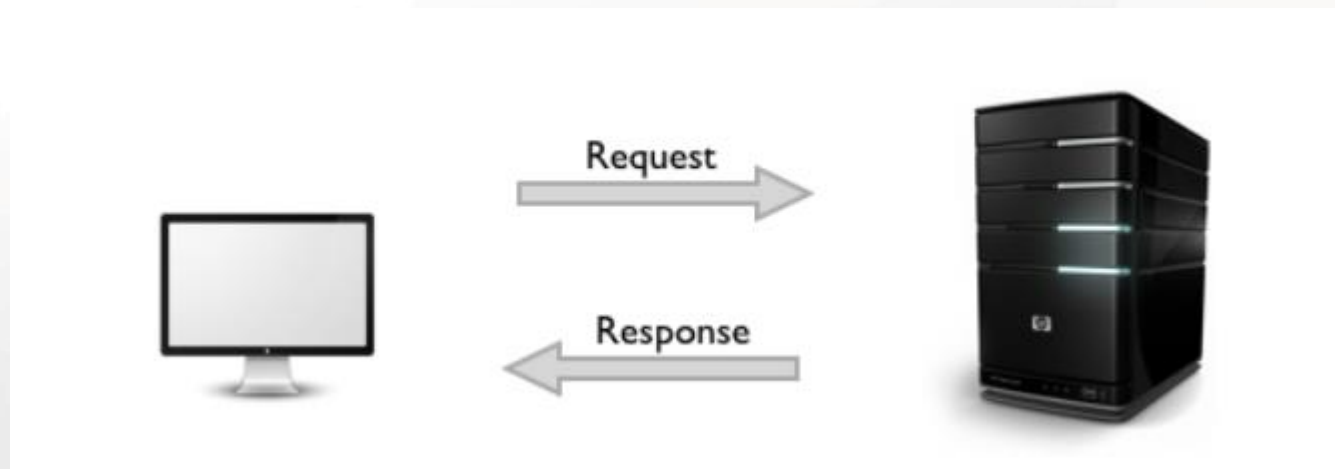


Diagram illustrating the components of an HTTP URL:

`http://www.domain.com:1234/path/to/resource?a=b&x=y`

The components are labeled as follows:

- protocol**: `http`
- host**: `www.domain.com`
- port**: `1234`
- resource path**: `/path/to/resource`
- query**: `?a=b&x=y`

# HTTP

## HTTP-сообщение: структура

**Стартовая строка** (англ. *Starting line*)

определяет тип сообщения;

**Заголовки** (англ. *Headers*)

характеризуют тело сообщения, параметры передачи и прочие сведения;

**Тело сообщения** (англ. *Message Body*)

непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

```
GET /wiki/страница HTTP/1.1
```

```
Host: ru.wikipedia.org
```

```
User-Agent: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509 Firefox/3.0b5
```

```
Accept: text/html
```

# HTTP

## Методы

### **GET:**

получить доступ к существующему ресурсу. В URL перечислена вся необходимая информация, чтобы сервер смог найти и вернуть в качестве ответа искомый ресурс.

### **POST:**

используется для создания нового ресурса. POST запрос обычно содержит в себе всю нужную информацию для создания нового ресурса.

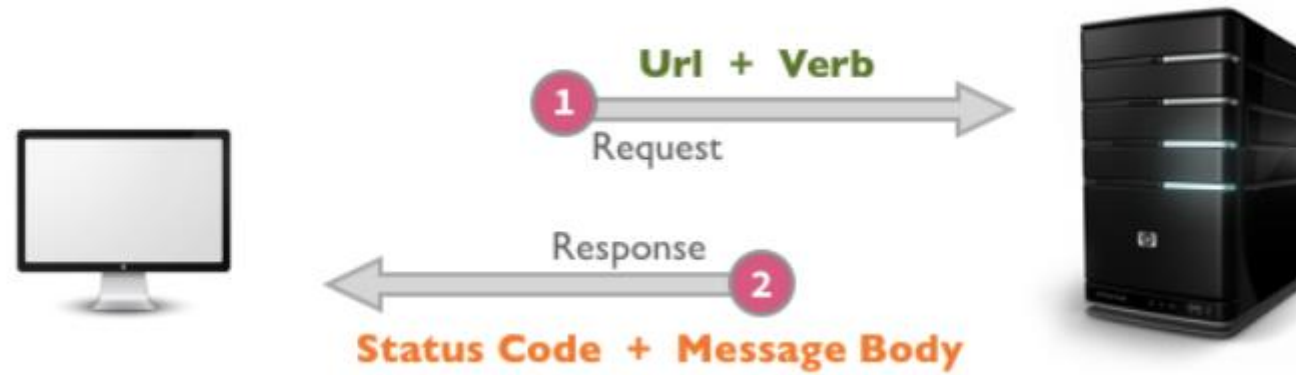
### **PUT:**

обновить текущий ресурс. PUT запрос содержит обновляемые данные.

### **DELETE:**

служит для удаления существующего ресурса.

# HTTP



*HTTP/1.0* **200 OK**



sync

# 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 }
```

## **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

async

# XMLHttpRequest

```
1 var xhr = new XMLHttpRequest();
2
3 xhr.open('GET', 'phones.json', true);
4
5 xhr.send(); // (1)
6
7 xhr.onreadystatechange = function() { // (3)
8     if (xhr.readyState != 4) return;
9
10    button.innerHTML = 'Готово!';
11
12    if (xhr.status != 200) {
13        alert(xhr.status + ': ' + xhr.statusText);
14    } else {
15        alert(xhr.responseText);
16    }
17
18 }
19
20 button.innerHTML = 'Загружаю...'; // (2)
21 button.disabled = true;
```

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

```
const unsigned short UNSENT = 0; // начальное состояние
const unsigned short OPENED = 1; // вызван open
const unsigned short HEADERS_RECEIVED = 2; // получены заголовки
const unsigned short LOADING = 3; // загружается тело (получен очередной пакет данных)
const unsigned short DONE = 4; // запрос завершён
```



async

# XMLHttpRequest

- `loadstart` – запрос начат.
- `progress` – браузер получил очередной пакет данных, можно прочитать текущие полученные данные в `responseText`.
- `abort` – запрос был отменён вызовом `xhr.abort()`.
- `error` – произошла ошибка.
- `load` – запрос был успешно (без ошибок) завершён.
- `timeout` – запрос был прекращён по таймауту.
- `loadend` – запрос был завершён (успешно или неуспешно)

## События:

- `onreadystatechange`
- `ontimeout`
- `onerror`
- `onload`
- `onprogress`
- `onabort`
- `onloadstart`
- `onloadend`

# XMLHttpRequest

HTTP-заголовки

## `setRequestHeader(name, value)`

Устанавливает заголовок `name` запроса со значением `value`.

```
xhr.setRequestHeader('Content-Type', 'application/json');
```

## `getResponseHeader(name)`

Возвращает значение заголовка ответа `name`, кроме `Set-Cookie` и `Set-Cookie2`.

```
xhr.getResponseHeader('Content-Type')
```

## `getAllResponseHeaders()`

Возвращает все заголовки ответа, кроме `Set-Cookie` и `Set-Cookie2`.

```
Cache-Control: max-age=31536000  
Content-Length: 4260  
Content-Type: image/png  
Date: Sat, 08 Sep 2012 16:53:16 GMT
```

# — JAVASCRIPT —

*HTTP. AJAX. XMLHttpRequest*