

**Язык JavaScript (Отправка
формы GET запросом, Ajax)**

JS

ТЕГ <FORM>

Основное предназначение тега <form> - это обмен данными между пользователем и сервером.

Форма отправляется на сервер по нажатию кнопки submit (или кнопки enter – имитирует submit).

Браузер подготавливает информацию для отправки в виде пары «имя=значение», где имя определяется атрибутом name тега **<input>**, а значение введено пользователем (например, атрибут value) или установлено в поле формы по умолчанию.

Действия при отправки формы: Клиент (браузер) отправляет HTTP-запрос на сервер с данными из формы. Затем сервер возвращает ответ клиенту. Ответ содержит сведения о состоянии запроса, а также может содержать запрошенное содержимое. !!!Страница перерисовывается, а в строке URL будет путь по которому отправлялись данные.

ТЕГ <FORM>

Основные атрибуты тега <form> настраивающие передачу данных серверу:

- ❖ **action** – URL адрес сервера по которому будут отправлены данные формы;
- ❖ **enctype** – способ кодирования данных формы;
- ❖ **method** – метод протокола HTTP (тег <form> поддерживает GET и POST).

ОТПРАВКА ФОРМЫ GET ЗАПРОСОМ

При отправки формы GET запросом есть следующие особенности:

- ❖ **Используется только тип кодировки:** application/x-www-form-urlencoded;
- ❖ При отправке данных **метод Get добавляет данные в URL-адрес** и длина URL ограничена (максимальная длина URL составляет 2048 символов). Вот пример формируемой URL: **/test/demo.html?name1=value1&name2=value2**

Параметры перечисляются после вопросительного знака, разделяются между собой символом амперсанда (&).

Нелатинские символы преобразуются в шестнадцатеричное представление (в форме %HH, где HH — шестнадцатеричный код для значения ASCII-символа), пробел заменяется на плюс (+).

- ❖ **Данные видны всем в URL**

AJAX

Изначально термин **Ajax** был аббревиатурой выражения **Asynchronous JavaScript and XML** (асинхронный JavaScript и XML), но сейчас используется как **обобщение для асинхронного стиля обмена сообщениями независимо от используемых технологий** (асинхронный обмен сообщениями означает, что приложение, отправив запрос, не должно ждать ответа от сервера, прежде чем приступить к выполнению других задач).

Аjax в настоящий момент — стандартный механизм для фоновой отправки и получения данных.

AJAX

В основе технологии **Ajax** лежит API **XMLHttpRequest**.

Объект XMLHttpRequest (или, как его кратко называют, «XHR») дает возможность из JavaScript делать HTTP-запросы к серверу без перезагрузки страницы.

Несмотря на слово «XML» в названии, XMLHttpRequest может работать с любыми данными, а не только с XML.

XMLHttpRequest СИНХРОННЫЙ

// 1. Создаём новый объект XMLHttpRequest

```
var xhr = new XMLHttpRequest();
```

// 2. Конфигурируем его: GET-запрос на URL 'phones.json'

```
xhr.open('GET', 'phones.json', false); //false - запрос синхронный
```

// 3. Отсылаем запрос

```
xhr.send();
```

// 4. Если код ответа сервера не 200, то это ошибка

```
if (xhr.status != 200) {
```

```
    // обработать ошибку
```

```
    alert( xhr.status + ': ' + xhr.statusText ); // пример вывода: 404: Not Found
```

```
} else {
```

```
    // вывести результат
```

```
    alert( xhr.responseText ); // responseText -- текст ответа.
```

```
}
```

XMLHttpRequest АСИНХРОННЫЙ

// 1. Создаём новый объект XMLHttpRequest

```
var xhr = new XMLHttpRequest();
```

// 2. Конфигурируем его: GET-запрос на URL 'phones.json'

```
xhr.open('GET', 'phones.json', true); //true - запрос синхронный
```

// 3. Отсылаем запрос

```
xhr.send();
```

/* 4. Обработчик изменения статуса: 0 - начальное состояние, 1 - вызван open, 2 - получены заголовки, 3 - загружается тело (получен очередной пакет данных), 4 - запрос завершён */

```
xhr.onreadystatechange = function() { // (0 → 1 → 2 → 3 → ... → 3 → 4)
```

```
    if (xhr.readyState != 4) return;
```

```
    if (xhr.status != 200) {
```

```
        alert( xhr.status + ': ' + xhr.statusText ); // обработать ошибку
```

```
    } else {
```

```
        alert(xhr.responseText); // вывести результат
```

```
    }
```

```
}
```


JQUERY AJAX АСИНХРОННЫЙ

```
$.ajax({  
    type: 'GET',  
    url: 'phones.json',  
    success: function(response) {  
        //ответ от сервера в переменной response  
        alert(response);  
    }  
});
```

КРОСС-ДОМЕННЫЕ ЗАПРОСЫ

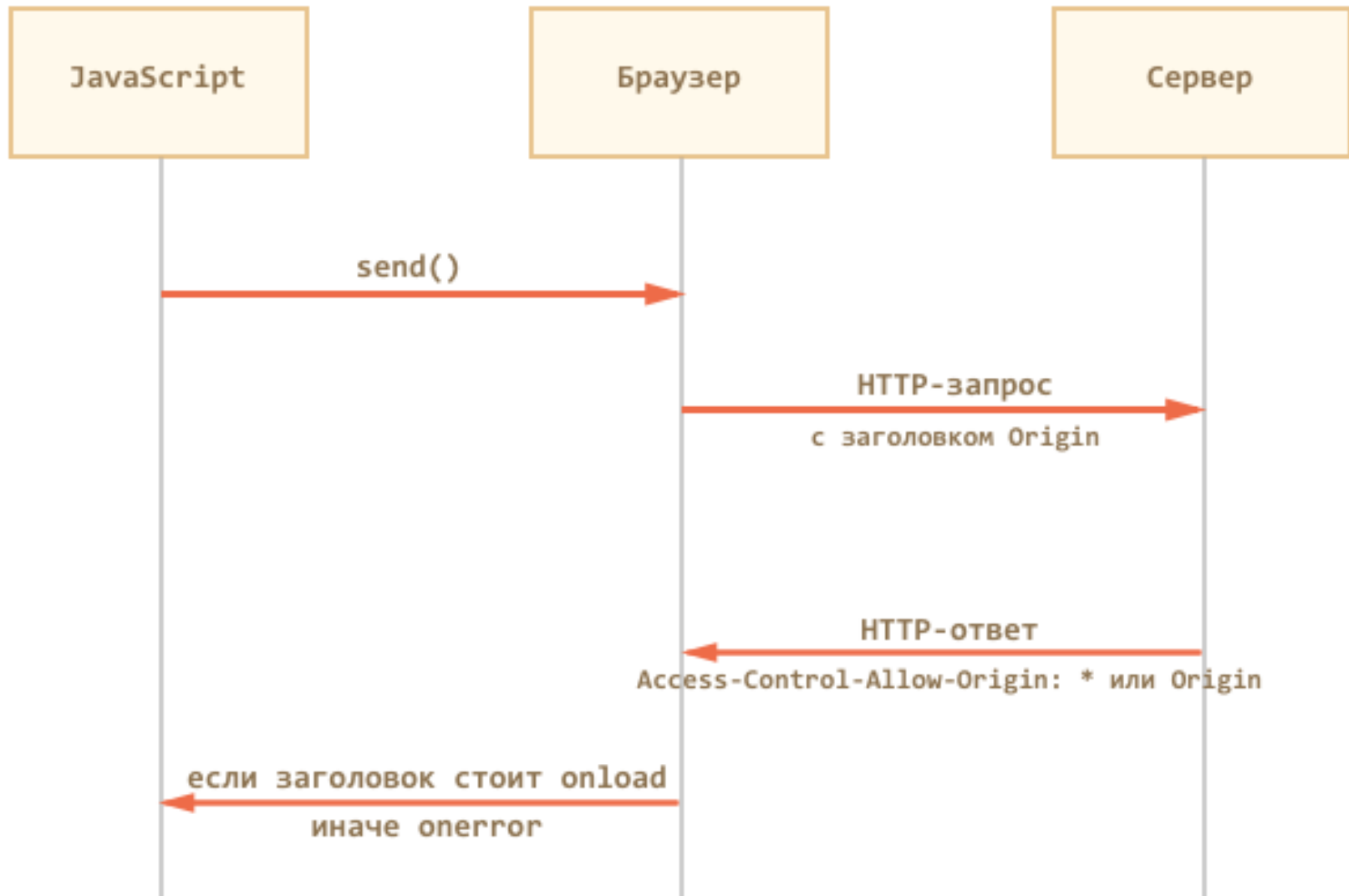
//Пример №1

```
$.ajax({  
    type: 'GET',  
    url: 'https://xml.meteoservice.ru/export/gismeteo/point/69.xml',  
    success: function(response) {  
        alert(response); //ответ от сервера в response  
    }  
});
```

//Пример №2

```
$.ajax({  
    type: 'GET',  
    url: 'http://www.nbrb.by/API/ExRates/Currencies',  
    success: function(response) {  
        alert(response); //ответ от сервера в response  
    }  
});
```

КРОСС-ДОМЕННЫЕ ЗАПРОСЫ



ПОЛЕЗНЫЕ ССЫЛКИ

<http://jquery.page2page.ru/index.php5/Ajax> - сайт посвященный jQuery на русском языке, раздел ajax