

Что такое AJAX

Как добавить что-то новое на страницу без перезагрузки

Когда мы делали [форму обратной связи на сайте](#), то она работала так:

- вы вводите сообщение и нажимаете «отправить»;
- загружается новая страница с подтверждением отправки.

Часто это неудобно: хочется оставить посетителя на странице, а не показывать что-то новое. Чтобы сделать именно так, используют AJAX.

AJAX (программисты называют его «аякс») — это сокращение от Asynchronous Javascript and XML, а на русский это переводится как «Асинхронный JavaScript и XML». Смысл здесь — в асинхронности действий браузера и сервера. Разберём эти понятия.

Синхронные запросы

Обычные запросы к серверу, чтобы получить и сформировать страницу, работают так:

1. Посетитель нажимает в форме обратной связи кнопку «Отправить».
2. Браузер видит, что по кнопке нужно отправить запрос на сервер. Он пакует запрос и отправляет.
3. Сервер получает запрос, обрабатывает и даёт ответ в виде новой веб-страницы.
4. Браузер загружает эту новую страницу с нуля. На ней уже все нужные пользователю данные, но есть ощущение «вспышки»: ты был на одной странице, потом экран побелел, и ты оказался на новой странице. Синхронность здесь в том, что все данные кучей отправляются на сервер, а потом вся страница такой же охапкой возвращается пользователю.

Именно так работает [форма обратной связи](#): вы нажимаете на кнопку и браузер, чтобы показать новые данные, начинает загружать новую страницу. Но есть и другой способ.

Асинхронные запросы

AJAX работает иначе: вместо того чтобы синхронно загружать новые данные с новой страницей, он запрашивает у сервера недостающую информацию и добавляет её на страницу. А сама страница не перезагружается.

1. Посетитель нажимает в форме обратной связи кнопку «Отправить».
2. Браузер запускает скрипт, привязанный к кнопке.
3. Этот скрипт отправляет запрос на сервер и получает в ответ **новую порцию информации от сервера**. Страница не перезагружается, всё происходит внутри скрипта.
4. Скрипт смотрит, что ответил сервер, и вживляет новую информацию в старую страницу.

5. Страница не перезагрузилась, посетитель остался там же, где и был, только с новыми данными.

Примеры таких запросов:

- Получить список новых сообщений в чате, не перезагружая весь чат.
- Ютуб этим пользуется, чтобы свернуть видео в маленький плеер в углу. .
- Бесконечные ленты в соцсетях — тоже отсюда. Когда вы доскролливаете до конца, на сервер улетает новый AJAX-запрос, и в ответ приходит новый контент.

Как устроен AJAX

Основа AJAX — объект XMLHttpRequest, который и отвечает за все запросы. Раньше его поддерживали не все браузеры, поэтому нужно было подключить отдельно библиотеку jQuery, в которой этот объект есть. Сейчас почти все браузеры научились поддерживать этот объект напрямую.

Вся магия асинхронных запросов AJAX основана на работе с **DOM-объектами** страницы в браузере:

- Когда браузер получает исходный код страницы, он внутри себя строит много виртуальных элементов на основе этого кода: заголовки, картинки, абзацы, ссылки и всё остальное.
- К каждому элементу модели можно обратиться отдельно, чтобы поменять его свойства или содержимое. Например, через JavaScript можно поменять текст в заголовке или перекрасить фон страницы, не перезагружая страницу.
- JavaScript делает запрос на сервер, получает ответ и дальше творит со страницей то, что нужно, в зависимости от самой программы. Отправить запрос и получить ответ ничем принципиально не отличается от чтения и записи переменных.

Плюсы и минусы AJAX

✔ **Удобство для посетителя и быстрый интерфейс.** Гораздо лучше видеть моментальное обновление данных на той же странице, чем загружать новую при каждой новой порции данных.

✔ **Меньше трафика.** Если нужно показать что-то новое, браузер получает с сервера не всю страницу целиком, а только то, чего нет на исходной странице.

✔ **Можно снизить нагрузку на сервер.** Если сайт формирует все страницы на лету, например, с помощью PHP, то можно один раз загрузить стандартные части страницы (шапку, меню и подвал), а остальное подгружать по необходимости. Меньше данных в запросе — быстрее ответ от сервера и базы.

✗ Нужен включённый JavaScript. Если в браузере выключить JS, то ничего не работает — данные не придут с сервера и интерактивной страницы не получится.

✗ Поисковые движки не видят AJAX-контент. Если содержимое страницы формируется на AJAX-запросах, то поисковые роботы не смогут его увидеть. Смысл в том, что поисковики смотрят на исходный код, а не на то, что приходит с сервера. Поисковая оптимизация таких страниц — головная боль для сеошника.

Что такое SEO

✗ Растёт сложность проекта. Работа с такими запросами требует определённой квалификации от программиста, чтобы он мог предусмотреть разные нештатные ситуации и обработать их заранее. А ещё нужно продумать бэкенд — поведение сервера в ответ на разные запросы.

✗ Ошибки при нестабильной связи. Если интернет работает с перебоями, то AJAX может не дожидаться ответа от сервера или не сможет отправить свой запрос. В итоге логика работы страницы может нарушиться — в итоге придётся перезагружать её полностью и начинать работу с нуля.

<https://ruseller.com/lessons.php?id=1452>