**AJAX** — «асинхронный [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) и [XML](https://ru.wikipedia.org/wiki/XML) — подход к построению интерактивных [пользовательских интерфейсов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), заключающийся в «фоновом» [обмене данными](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D0%B5%D0%BD_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8) [браузера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) с [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). В результате, при обновлении данных [веб-страница](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) не перезагружается полностью, и веб-приложения становятся быстрее и удобнее. В классической модели веб-приложения:

* Пользователь заходит на веб-страницу и нажимает на какой-нибудь её элемент.
* [Браузер](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) формирует и отправляет [запрос](https://ru.wikipedia.org/wiki/HTTP) [серверу](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).
* В ответ сервер генерирует совершенно новую веб-страницу и отправляет её браузеру и т. д., после чего браузер полностью перезагружает всю страницу.

При использовании AJAX:

* Пользователь заходит на веб-страницу и нажимает на какой-нибудь её элемент.
* [Скрипт](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) (на стороне клиента) определяет, какая информация необходима для обновления страницы.
* [Браузер](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) отправляет соответствующий запрос на [сервер](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)).
* Сервер возвращает только ту часть документа, на которую пришёл запрос.
* [Скрипт](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) вносит изменения с учётом полученной информации (без полной перезагрузки страницы).

В качестве формата передачи данных могут использоваться фрагменты простого текста, [HTML](https://ru.wikipedia.org/wiki/HTML)-кода, [JSON](https://ru.wikipedia.org/wiki/JSON), [XML](https://ru.wikipedia.org/wiki/XML) или двоичные данные.

## **Преимущества**

**Экономия трафика**

Использование AJAX позволяет [сократить трафик](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D1%8F_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0&action=edit&redlink=1) при работе с веб-приложением благодаря тому, что вместо загрузки всей страницы достаточно загрузить только изменившуюся часть или вообще только получить/передать набор данных в формате [JSON](https://ru.wikipedia.org/wiki/JSON) или [XML](https://ru.wikipedia.org/wiki/XML), а затем изменить содержимое страницы с помощью JavaScript.

**Уменьшение нагрузки на сервер**

При правильной реализации AJAX позволяет снизить нагрузку на сервер в несколько раз.

В частности, все страницы сайта чаще всего генерируются по одному шаблону, включая неизменные элементы («шапка», «навигационная панель», «подвал» и т. д.), для генерации которых требуются обращения к разным файлам, время на обработку скриптов (а иногда и запросы к БД) — всё это можно опустить, если заменить полную загрузку страницы генерацией и передачей лишь содержательной части.

**Ускорение реакции интерфейса**

Поскольку загрузка изменившейся части значительно быстрее, то пользователь видит результат своих действий быстрее и без мерцания страницы (возникающего при полной перезагрузке).

**Возможности для интерактивной обработки**

Например, при вводе поискового запроса в [Google](https://ru.wikipedia.org/wiki/Google_(%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)) выводится подсказка с возможными вариантами запроса. На многих сайтах при регистрации пользователь вводит имя, и сразу же видит, доступно это имя или нет. AJAX удобен для программирования [чатов](https://ru.wikipedia.org/wiki/%D0%A7%D0%B0%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)), [административных панелей](https://ru.wikipedia.org/w/index.php?title=%D0%90%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BD%D0%B5%D0%BB%D1%8C&action=edit&redlink=1) и других инструментов, которые выводят меняющиеся со временем данные.

## **Недостатки**

**Отсутствие интеграции со стандартными инструментами браузера**

Динамически создаваемые страницы не регистрируются браузером в истории посещения страниц, поэтому не работает кнопка «Назад», предоставляющая пользователям возможность вернуться к просмотренным ранее страницам, но существуют скрипты, которые могут решить эту проблему.

Другой недостаток изменения содержимого страницы при постоянном [URL](https://ru.wikipedia.org/wiki/URL) заключается в невозможности сохранения закладки на желаемый материал. Проблему можно успешно решить с помощью History.pushState[[5]](https://ru.wikipedia.org/wiki/AJAX" \l "cite_note-5).

**Старые методы учёта статистики сайтов становятся неактуальными**

Многие сервисы статистики ведут учёт просмотров новых страниц сайта. Для сайтов, страницы которых широко используют AJAX, такая статистика теряет актуальность.

**Усложнение проекта**

Перераспределяется логика обработки данных — происходит выделение и частичный перенос на сторону клиента процессов первичного форматирования данных. Это усложняет контроль целостности форматов и типов. Конечный эффект технологии может быть нивелирован необоснованным ростом затрат на кодирование и управление проектом, а также риском снижения доступности сервиса для конечных пользователей.

**Требуется включённый JavaScript в браузере**

JavaScript может быть выключен из соображений безопасности. И, конечно же, AJAX-страницы труднодоступны неполнофункциональным браузерам, роботам и [веб-архивам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B0%D1%80%D1%85%D0%B8%D0%B2).

**Низкая скорость при грубом программировании**

Казалось бы, AJAX предназначен именно для повышения скорости. Но когда AJAX-запросов на одной странице много и, например, по каждому щелчку подгружается список, AJAX-страница становится даже медленнее традиционной.

**Плохое поведение на ненадёжных соединениях**

Если связь часто теряется (из-за потери несущей или перегрузки канала), обычную страницу можно как минимум перезагрузить. AJAX-страницу (например, с «бесконечной» прокруткой) приходится перезагружать с самого начала и искать, где остановился. Параллельная работа — особенность AJAX — здесь оказывает медвежью услугу, деля и без того узкий канал на множество маленьких соединений, и велика вероятность, что какое-то будет разорвано. Частично решается API истории.

**Риск** [**фабрикации запросов другими сайтами**](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%B4%D0%B5%D0%BB%D0%BA%D0%B0_%D0%BC%D0%B5%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D1%8B%D1%85_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2)

**История AJAX и популярные реализации: https://www.ibm.com/developerworks/ru/library/wa-aj-ajaxhistory/index.html**