**16. Программный интерфейс приложения API**

API — это интерфейс, позволяющий двум независимым компонентам программного обеспечения обмениваться информацией. Проще говоря, это то, что обеспечивает эффективный процесс коммуникаций между программами, использующими функции и ресурсы друг друга. API играет роль посредника между внутренними и внешними программными функциями, обеспечивая настолько эффективный обмен информацией, что конечные пользователи обычно его просто не замечают. Говоря по-простому, API действует как виртуальный посредник и передает информацию из одного интерфейса, например мобильного приложения, в другой. API связывает различные части программной платформы, чтобы передаваемая информация дошла до места назначения.Эти связующие узлы не только выполняют роль внутренних каналов связи, но и позволяют внешним инструментам получать доступ к этой же информации. Таким образом API-интерфейсы могут относиться к одной из двух категорий:

**Внутренние/частные** API(Частные API доступны только разработчикам и пользователям из числа сотрудников организации. Такие API обычно связывают внутренние процессы для уменьшения разрозненности рабочих данных и оптимизации совместной работы.)

**Внешние/открытые** API(Открытые API, в свою очередь, позволяют внешним разработчикам получать доступ к информации и интегрировать информацию, которая передается из одного программного инструмента в другой. Открытые или частные API экономят время разработчиков, позволяя им объединять платформы с имеющимися инструментами и устраняя необходимость в создании нового функционала с нуля.)

Вот несколько распространенных примеров использования API:

* Обмен информацией о рейсах между авиакомпаниями и туристическими сайтами
* Использование Google Maps в приложении для совместных поездок (райдшеринга)
* Создание виртуальных собеседников в службе обмена сообщениями
* Встраивание видеоклипов с YouTube на веб-странице
* Автоматизация рабочих процессов в программных инструментах для B2B-сектора

1. **Веб-сервера. Статический и Динамический контент**

Понятие «веб-сервер» может относиться как к аппаратной начинке, так и к программному обеспечению. Или даже к обеим частям, работающим совместно.

1. С точки зрения "железа", «веб-сервер» — это компьютер, который хранит файлы сайта и доставляет их на устройство конечного пользователя (веб-браузер и т.д.). Он подключен к сети Интернет и может быть доступен через доменное имя.
2. С точки зрения ПО, *веб-сервер* включает в себя несколько компонентов, которые контролируют доступ веб-пользователей к размещенным на сервере файлам, как минимум — это *HTTP-сервер*. *HTTP-сервер* — это часть ПО, которая понимает веб-адреса и HTTP.

На самом базовом уровне, когда браузеру нужен файл, размещенный на веб-сервере, браузер запрашивает его через HTTP-протокол. Когда запрос достигает нужного веб-сервера ("железо"), сервер HTTP (ПО) принимает запрос, находит запрашиваемый документ и отправляет обратно, также через HTTP. Чтобы опубликовать веб-сайт, необходим либо статический, либо динамический веб-сервер.

**Статический веб-сервер**, или стек, состоит из компьютера ("железо") с сервером HTTP (ПО). Мы называем это «статикой», потому что сервер посылает размещенные файлы в браузер «как есть».

**Динамический веб-сервер** состоит из статического веб-сервера и дополнительного программного обеспечения, чаще всего *сервера приложения* и *базы данных*. Мы называем его «динамическим», потому что сервер приложений изменяет исходные файлы перед отправкой в ваш браузер по HTTP.

**«Динамическое»** означает, что сервер обрабатывает данные или даже генерирует их на лету из базы данных. Это обеспечивает большую гибкость, но технически сложнее в реализации и обслуживании. **Динамический сайт** — состоит из динамичных страниц - контента, скриптов и прочего, в большинстве случаев в виде отдельных файлов. Страница сайта, показываемая в итоге браузеру пользователя, формируется на стороне сервера динамически, по запросу, из страницы-шаблона и отдельно хранимого содержимого (информации, скриптов и др.). Как правило, для отображения любого количества однотипных страниц используется одна страница-шаблон, в которую подгружается соответствующее содержимое.

**Процесс получения содержимого сайта обычно выглядит следующим образом:**

1. Генерация содержимого на стороне сервера;
2. Передача сгенерированной странички клиенту;
3. Генерация содержимого на стороне клиента.

### Генерация содержимого на стороне сервера

Сервер получает запрос от Клиента (например, ***page.ru/index.php***) и запускает обработку файла-скрипта (в данном случае — ***index.php***) интерпретатором. Языки программирования на Серверной стороне используются разные, наиболее часто встречаются, например: Java, PHP, Perl и другие.

Именно на этой стадии происходит **выборка** необходимой информации из баз данных и **наполнение** ею страницы, после чего готовая страница передаётся Клиенту.

### Генерация содержимого на стороне клиента

После того, как страница получена Клиентом с Сервера, программа-браузер обрабатывает её и отображает Пользователю, при этом *исполняя скрипты* Клиентской стороны, если они были указаны в странице и получены. На Клиентской стороне используется JavaScript

### Комбинированная генерация

Чаще всего в жизни встречается именно комбинация этих двух методов генерации — весь «новый интернет» основан на нём, это, и «умная строка с подсказкой вариантов» у поисковых систем, и всплывающие меню, и многое другое.

4

[**Джино.Журнал**](https://zen.yandex.ru/jino_journal)

276 подписчиков

Подписаться

# Статические и динамические сайты сегодня: какие лучше и почему

22 августа 2019

159 дочитываний

1,5 мин.

292 просмотра. Уникальные посетители страницы.

159 дочитываний, 54%. Пользователи, дочитавшие до конца.

1,5 мин. Среднее время дочитывания публикации.



Казалось бы, статические сайты некоторое время назад ушли в прошлое и простые в разработке динамические заменили их уже навсегда. Но с появлением разных генераторов статических сайтов и специальных фреймворков споры о том, какой тип сайтов оптимальнее, разгорелись с новой силой. Сначала мы рассмотрим различия статических и динамических страниц, а затем — их достоинства и недостатки, чтобы понять, за каким типом сайтов будущее.

**Статические сайты** состоят из неизменяемых страниц. Это значит, что сайт имеет один и тот же внешний вид, а также одно и то же наполнение для всех посетителей. При запросе такого сайта в браузере сервер сразу предоставляет готовый HTML-документ в исходном виде, в котором он и был создан. Кроме HTML, в коде таких страниц используется разве что CSS и JavaScript, что обеспечивает их легкость и быструю загрузку.

Чаще всего статическими бывают сайты с минимальным количеством страниц или с контентом, который не нужно регулярно обновлять, а именно **сайты-визитки**, **каталоги продукции**, **справочники технической документации**. Однако с помощью сторонних инструментов существует возможность добавить на такие страницы отдельные динамические элементы (комментарии, личный кабинет для пользователей, поиск).

**Динамические сайты**, в свою очередь, имеют изменяемые страницы, адаптирующиеся под конкретного пользователя. Такие страницы не размещены на сервере в готовом виде, а собираются заново по каждому новому запросу. Сначала сервер находит нужный документ и отправляет его интерпретатору, который выполняет код из HTML-документа и сверяется с файлами и базой данных. После этого документ возвращается на сервер и затем отображается в браузере. Для интерпретации страниц на серверной стороне используются языки программирования Java, PHP, ASP и другие.

Самыми яркими примерами динамических сайтов являются страницы, созданные на основе систем управления контентом (CMS). Среди них чаще всего встречаются **интернет-магазины**, а также **форумы**, **страницы с отзывами** и другие ресурсы с возможностью размещения контента посетителями.