# Клиентская оптимизация

## Уменьшение количества HTTP-запросов

80% загрузки страницы ориентировано на загрузку компонентов страницы: скриптов, фотографий, файлов CSS, flash. Спецификация HTTP/1.1 советует, чтобы браузеры параллельно загружали не более 2-х компонентов веб-страницы с одного хоста. Уменьшив количество этих компонентов мы уменьшаем количество HTTP-запросов к серверу и как результат увеличиваем скорость загрузки страницы.

* Использование CSS-спрайтов.
* Использование Inline-картинок. Inline-картинки используют URL-схему data: для встраивания картинки в саму страницу. Это, однако, увеличит размер HTML-документа. Встраивая inline-картинки в ваши таблицы стилей вы добьетесь уменьшения запросов к серверу, а размер HTML останется прежним.
* Объединение нескольких файлов в один.

## Помещение CSS файлов в начале страницы

Помещая подключение к css файлам в хедере страницы мы получаем постепенный рендеринг страницы, т.е. страница будет загружаться постепенно — сначала заголовок, потом лого наверху, навигация и т.д. – а это в свою очередь служит отличным индикатором загрузки страницы для пользователя и улучшает общее впечатление от сайта.

Если размещать CSS файлы внизу страницы, то это не позволяет многим браузерам рендерить страницу постепенно. Это объясняется тем, что браузер «не хочет» перерисовывать элементы, у которых после загрузки страницы может измениться стиль. Так что все свои CSS файлы всегда подключайте в верхней части страницы в секции HEAD.

## Помещение javascript в конец страницы

Помещая javascript-файлы вниз страницы мы позволяем браузеру загрузить страницу с контентом в первую очередь, а уже потом начать загрузку javascript-файлов. Если же Ваш сайт идет в ногу со временем и содержит все возможные интерактивные «примочки», то этих javascript-файлов может быть несколько и весить они могут несколько сотен килобайт, поэтому перед загрузкой страницы заставлять пользователя ждать пока загрузятся все Ваши javascipt-файлы губительно.

Кроме того, внешние .js-файлы блокируют параллельную загрузку. Спецификация HTTP/1.1 советует, чтобы браузеры параллельно загружали не более 2-х компонентов веб-страницы с одного хоста. Таким образом, если картинки для вашего сайта располагаются на разных хостах, вы получите более 2-х параллельных загрузок. А когда загружается скрипт, браузер не будет начинать никаких других загрузок, даже с других хостов.

Минимизировать css и javascript

Минимизация файла — это удаление из кода всех несущественных символов с целью уменьшения объема файла и ускорения его загрузки. В минимизированном файле удаляются все комментарии и незначащие пробелы, переносы строк, символы табуляции. Здесь все просто. Чем меньше объем файла, тем меньше времени понадобится браузеру на его загрузку.

## Использывание поддоменов для параллельного скачивания

Согласно спецификации HTTP/1.1 на браузеры накладываются ограничения на количество одновременно загружаемых компонентов сайта, а именно не более 2-х компонентов с одного хоста. Поэтому если на Вашем сайте много графики, то ее лучше вынести на отдельный поддомен или поддомены. Для Вас это будет один и тот же сервер, а для браузера – разные. Чем больше поддоменов Вы создадите, тем больше файлов браузер сможет одновременно загрузить и тем быстрее загрузится вся страница сайта. Вам остается лишь изменить адрес картинок на новый. Очень простой, но действенный способ.

## Использывание кэша браузера

Кеширование становится крайне важным для современных веб-сайтов, которые используют обширное подключение JavaScript и CSS. Дело в том, что когда посетитель зашел на Ваш сайт в первый раз, то браузер выполнит загрузку всех javascript и css-файлов, также загрузит всю графику и флэш, однако правильно выставив HTTP-заголовок Expires, вы сделаете компоненты страницы кешируемыми

Поэтому выставляйте HTTP-заголовок Expires везде, где только это возможно, на несколько дней или даже месяцев вперед. Для того, чтобы веб-сервер Apache отдавал соответствующие рекомендациям HTTP-заголовки Expires, необходимо добавить в файл .htaccess, находящийся в корневой папке сайта, следующие строки:

<IfModule mod\_expires.c>

Header append Cache-Control "public"

FileETag MTime Size

ExpiresActive On

ExpiresDefault "access plus 0 minutes"

ExpiresByType image/ico "access plus 1 years"

ExpiresByType text/css "access plus 1 years"

ExpiresByType text/javascript "access plus 1 years"

ExpiresByType image/gif "access plus 1 years"

ExpiresByType image/jpg "access plus 1 years"

ExpiresByType image/jpeg "access plus 1 years"

ExpiresByType image/bmp "access plus 1 years"

ExpiresByType image/png "access plus 1 years"

</IfModule>

Данный фрагмент файла конфигурации Веб-сервера Apache проверяет наличие модуля mod\_expires и, если модуль mod\_expires доступен, включает отдачу HTTP-заголовков Expires, которые устанавливают срок хранения перечисленных выше объектов в кэше браузеров и прокси-серверов равный одному году с момента первой загрузки. Установив такой срок жизни кэша браузера, может возникнуть сложность с обновлением файлов. Поэтому если Вы изменили содержимое css или javascript-файла и хотите, чтобы эти изменения обновились в кэше браузера, то необходимо изменить название самого файла. Обычно в название файла добавляют его версию, например так: styles.v1.css

## Использывание CDN для загрузки популярных JavaScript библиотек

Если на Вашем сайте используется популярный javascript фреймворк, например jQuery, то для его подключения лучше использовать CDN.

CDN (Content Delivery Network) — это множество веб-серверов, разнесенных географически для достижения максимальной скорости отдачи контента клиенту. Сервер, который непосредственно будет отдавать контент пользователю, выбирается на основании некоторых показателей. Например, выбирается сервер с наименьшим числом промежуточных хопов до него либо с наименьшим временем отклика. Кроме того браузер кэширует javascript-файлы, и если Вы посещали сайты на котором используется такой метод, то эта библиотека уже есть в кэше Вашего браузера, и он не будет загружать её снова.

Одним из таких CDN – является *Google Libraries*. Это CDN для популярных open-source JavaScript библиотек. Загрузка популярных javascript фреймверков с Google Libraries позволяет увеличить скорость загрузки страницы и снизит траффик на ваш сервер.

## Оптимизация изображений

Необходимо определять подходящий формат для ваших изображений. Выбор неверного формата изображения, может существенно увеличить размер файла.

GIF – идеально подходят для изображений с несколькими цветами, например логотип.

JPEG – отлично подходят для детализированых изображений с большим количеством цветов, такие как фотографии.

PNG – ваш выбор, когда вам нужно высококачественное изображение с прозрачностью.

## Не масштабируйте изображения

Не изменяйте размер изображения при помощи атрибутов width и height тега , либо при помощи CSS. Это тоже негативно влияет на скорость загрузки страницы. Если у Вас есть изображение размером 500x500px, а вставить на сайт Вы хотите изображение с размером 100x100px, то лучше изменить размер оригинальной картинки при помощи графического редактора Photoshop, или любого другого. Чем меньший вес картинки, тем меньше времени потребуется для её загрузки.

## Используйте Gzip- сжатие

Как показали проведенные исследования, gzip-сжатие текстового файла «на лету» в 95–98% случаев позволяет сократить время на передачу файла браузеру. Если хранить архивированные копии файлов на сервере (в памяти proxy-сервера или просто на диске), то соединение в общем случае удается освободить в 3-4 раза быстрее.

Начиная с версии протокола HTTP/1.1, веб-клиенты указывают, какие типы сжатия они поддерживают, устанавливая заголовок Accept-Encoding в HTTP-запросе.

Accept-Encoding: gzip, deflate

Если веб-сервер видит такой заголовок в запросе, он может применить сжатие ответа одним из методов, перечисленных клиентом. При выдаче ответа посредством заголовка Content-Encoding сервер уведомляет клиента о том, каким методом сжимался ответ.

Content-Encoding: gzip

Переданные таким образом данные меньше первоначальных примерно в 5 раз, и это существенно ускоряет их доставку. Однако здесь есть один недостаток: увеличивается нагрузка на веб-сервер. Но вопрос с сервером всегда можно решить. Так что не будем обращать на это внимание.

Для того, чтобы включить GZIP-сжатие на своем сайте, необходимо в файле .htaccess прописать следующие строки кода:

<IfModule mod\_deflate.c>

AddOutputFilterByType DEFLATE text/html

AddOutputFilterByType DEFLATE application/javascript

AddOutputFilterByType DEFLATE text/javascript

AddOutputFilterByType DEFLATE text/css

<IfModule mod\_setenvif.c>

BrowserMatch ^Mozilla/4 gzip-only-text/html

BrowserMatch ^Mozilla/4\.0[678] no-gzip

BrowserMatch \bMSIE !no-gzip !gzip-only-text/html

</IfModule>

</IfModule>

Но опять таки, данный код работает не на всех серверах, поэтому лучше обратиться в службу поддержки Вашего хостинг-провайдера и уточнить этот вопрос.