

ОБ АВТОРЕ

Баланов Антон Николаевич имеет большой опыт руководства и консультирования в сфере ИТ-технологий. Работал топ-менеджером в крупных компаниях — таких, как Industrial and Commercial Bank of China (КНР), Caravan portal (ОАЭ), Банк ВТБ, Сбербанк России, VK; руководил разработками сервиса Gosuslugi.ru. Имеет степень MBA IT (CIA) и сертификации Microsoft, CompTIA, ISACA, PMI, SHRM, ПВА, HRCI, ISO, Six Sigma (Master Black Belt). Преподавал в следующих вузах и учебных центрах: Российском университете дружбы народов, СберУниверситете, Институте бизнеса и делового администрирования и Центре подготовки руководителей и команд цифровой трансформации (на базе Высшей школы государственного управления РАНХиГС). Автор десятков книг и научно-практических публикаций в профессиональных изданиях. Является советником Российской академии естественных наук.

Широкая эрудиция и глубокие профессиональные компетенции автора в сфере ИТ-технологий позволили ему создать книжную серию «Айтишный университет», один из выпусков которой находится перед вами.

ОГЛАВЛЕНИЕ

Глава 1. Оптимизация изображений и видео	11
Введение	11
Выбор оптимального формата изображений и видео.....	12
Сжатие и оптимизация размера файлов изображений и видео	14
Использование ленивой загрузки для улучшения производительности	16
Оптимизация ресурсов для различных устройств и разрешений экранов	19
Заключение	22
 Глава 2. Кэширование и сжатие файлов	24
Введение	24
Использование кэширования для хранения временных копий ресурсов.....	25
Компрессия и сжатие файлов для уменьшения их размера	27
Настройка HTTP-заголовков для эффективного кэширования и сжатия.....	30
Использование инструментов и практик для контроля и управления кэшем	32
Заключение	34
 Глава 3. Минимизация HTTP-запросов.....	36
Введение	36
Сокращение числа HTTP-запросов, необходимых для загрузки страницы	37

8 Оглавление

Объединение и минификация файлов CSS и JavaScript	39
Использование спрайтов и шрифтовых пиктограмм для сокращения запросов.....	41
Разработка эффективных методов загрузки данных и ресурсов на страницу	44
Заключение	46
 Глава 4. Оптимизация кода	 48
Введение	48
Идентификация и оптимизация узких мест в коде приложения.....	49
Использование эффективных алгоритмов и структур данных	51
Оптимизация SQL-запросов и взаимодействия с базами данных	55
Применение современных методов и практик программирования для повышения производительности	57
Заключение	61
 Глава 5. Использование Content Delivery Network (CDN).....	 63
Введение	63
Введение в Content Delivery Network и его преимущества	64
Распределение статических ресурсов по серверам CDN	67
Ускорение загрузки контента через распределенные серверы.....	68
Настройка и интеграция CDN в веб-приложения.....	71
Заключение	73
 Глава 6. Оптимизация мобильной веб-разработки	 75
Введение	75
Принципы и методы оптимизации мобильных веб-приложений	76
Адаптивный и отзывчивый дизайн для разных устройств	78

Оптимизация размера и скорости загрузки на мобильных устройствах.....	81
Улучшение производительности и энергоэффективности на мобильных платформах	83
Заключение	85
Глава 7. Оптимизация SEO и поисковой видимости	88
Введение	88
Понимание принципов SEO и оптимизации контента для поисковых систем	89
Использование правильных метаданных, тегов и структуры URL	91
Улучшение скорости загрузки и удобства использования для повышения рейтинга поиска	94
Оптимизация мобильной доступности и опыта пользователей для улучшения SEO.....	96
Заключение	98
Глава 8. Мониторинг и анализ производительности.....	100
Введение	100
Использование инструментов для мониторинга производительности веб-приложений.....	101
Анализ показателей производительности, включая время загрузки, задержки и использование ресурсов	103
Определение проблемных областей и оптимизация для улучшения производительности	105
Установление метрик и целей производительности, а также контроль и отчетность	107
Заключение	109
Глава 9. Оптимизация безопасности и защита от угроз.....	111
Введение	111
Обеспечение безопасности веб-приложений через оптимизацию кода и настройки	112
Применение лучших практик для защиты от известных уязвимостей.....	114
Регулярное обновление и мониторинг безопасности для предотвращения атак	116

Обеспечение конфиденциальности и целостности данных веб-приложений	118
Заключение	120
 Глава 10. Оптимизация пользователями и удобство использования	
Введение	123
Исследование и анализ поведения пользователей веб-приложений	124
Повышение удобства использования и интерактивности интерфейса	126
Оптимизация навигации и структуры приложения для улучшения пользовательского опыта	129
Тестирование и оптимизация процессов работы с пользователем для достижения лучших показателей удовлетворенности	131
Заключение	133

ГЛАВА 1

ОПТИМИЗАЦИЯ ИЗОБРАЖЕНИЙ И ВИДЕО

ВВЕДЕНИЕ

Глава 1 посвящена оптимизации изображений и видео, что является важным аспектом при разработке веб-приложений. Изображения и видео часто являются неотъемлемой частью веб-содержимого, но неоптимизированные файлы могут существенно замедлять загрузку страницы, увеличивать использование трафика и влиять на общую производительность приложения. В этой главе мы рассмотрим различные методы оптимизации изображений и видео, которые помогут повысить производительность и улучшить пользовательский опыт.

Первым шагом в оптимизации изображений и видео является выбор оптимального формата файлов. Различные форматы, такие как JPEG, PNG, GIF, WebP и другие, имеют свои особенности и применение в зависимости от типа контента и требований к качеству. Выбор правильного формата поможет минимизировать размер файла и сохранить приемлемое качество изображения или видео.

Второй важной задачей является сжатие и оптимизация размера файлов изображений и видео. Применение сжатия позволяет уменьшить размер файлов, не сильно ухудшая качество. Существуют различные алгоритмы сжатия — как с потерями, так и без потерь, которые можно использовать в зависимости от требований проекта. Оптимизация размера файлов помогает сократить время загрузки страницы и уменьшить использование трафика, что особенно важно для пользователей с медленным интернет-соединением или мобильных устройств.

Ленивая загрузка (Lazy Loading) — еще один эффективный метод оптимизации, который используется для улучшения производительности при загрузке изображений и видео. При

ленивой загрузке ресурсы загружаются только тогда, когда они становятся видимыми на экране пользователя. Это позволяет ускорить начальную загрузку страницы и сократить использование ресурсов, так как только видимые изображения и видео загружаются, когда пользователь прокручивает страницу.

Оптимизация ресурсов для различных устройств и разрешений экранов — еще один важный аспект при разработке веб-приложений. С разнообразием устройств и разрешений экранов, необходимо адаптировать изображения и видео под конкретные условия отображения. Использование ресурсов оптимизированных для конкретного устройства или разрешения экрана позволяет достичь лучшего качества и уменьшить размер файлов, что в свою очередь положительно влияет на загрузку и производительность приложения.

В главе 1 мы изучим различные аспекты оптимизации изображений и видео, включая выбор формата файлов, сжатие и оптимизацию размера, использование ленивой загрузки и оптимизацию ресурсов для разных устройств и разрешений экранов. Эти методы помогут улучшить производительность и загрузку веб-приложений, обеспечивая более быстрый и эффективный пользовательский опыт.

ВЫБОР ОПТИМАЛЬНОГО ФОРМАТА ИЗОБРАЖЕНИЙ И ВИДЕО

Выбор оптимального формата изображений и видео является важным аспектом веб-разработки, поскольку он влияет на качество, размер файла и скорость загрузки контента.

Таблица 1.1
Сравнение форматов изображений

Формат	Преимущества	Недостатки
JPEG	<ul style="list-style-type: none"> • Хорошее сжатие • Поддерживается всеми браузерами 	<ul style="list-style-type: none"> • Потеря качества при сжатии • Не подходит для изображений с прозрачностью

Формат	Преимущества	Недостатки
PNG	<ul style="list-style-type: none"> Поддерживает прозрачность Хорошее сжатие без потери качества Подходит для иллюстраций и логотипов 	<ul style="list-style-type: none"> Большой размер файлов Медленная загрузка
GIF	<ul style="list-style-type: none"> Поддерживает анимацию Маленький размер файлов Поддерживается всеми браузерами Подходит для простых графических элементов 	<ul style="list-style-type: none"> Ограниченнная цветовая палитра Низкое качество изображения
WebP	<ul style="list-style-type: none"> Хорошее сжатие без потери качества Поддерживает прозрачность Маленький размер файлов 	<ul style="list-style-type: none"> Не поддерживается всеми браузерами Ограниченнная поддержка платформами

При выборе оптимального формата изображений и видео для веб-приложений необходимо учитывать различные факторы, такие как качество, размер файла и скорость загрузки. Формат JPEG обеспечивает хорошее сжатие изображений, но при этом происходит потеря качества. Он широко поддерживается всеми браузерами, что делает его популярным для фотографий и изображений с реалистичными цветами.

Формат PNG поддерживает прозрачность и обеспечивает хорошее сжатие без потери качества. Однако файлы в формате PNG могут быть крупными, что приводит к медленной загрузке страниц. Этот формат часто используется для иллюстраций, логотипов и изображений с прозрачностью.

Формат GIF позволяет создавать анимированные изображения и имеет маленький размер файлов. Однако он имеет ограниченную цветовую палитру и низкое качество изображения.

GIF поддерживается всеми браузерами и обычно используется для простых графических элементов.

Формат WebP обеспечивает хорошее сжатие без потери качества и поддерживает прозрачность. Однако он не поддерживается всеми браузерами и имеет ограниченную поддержку на различных платформах. WebP обычно используется для оптимизации загрузки страниц и уменьшения размера файлов.

Пример использования таблицы.

Представим, что вы разрабатываете веб-приложение для фотографической галереи. Вам нужно выбрать формат изображений, который обеспечит хорошее качество и приемлемый размер файлов для быстрой загрузки. Из таблицы видно, что формат JPEG предлагает хорошее сжатие и поддерживается всеми браузерами. Этот формат будет отличным выбором для фотографий в вашей галерее.

Обратите внимание, что выбор формата изображений и видео зависит от конкретных требований вашего веб-приложения и его аудитории. Рекомендуется провести тестирование различных форматов и оценить их влияние на качество и скорость загрузки контента перед принятием окончательного решения.

СЖАТИЕ И ОПТИМИЗАЦИЯ РАЗМЕРА ФАЙЛОВ ИЗОБРАЖЕНИЙ И ВИДЕО

Сжатие и оптимизация размера файлов изображений и видео являются важными шагами для улучшения производительности веб-приложений. Большие файлы могут замедлить загрузку страницы, потреблять больше пропускной способности сети и занимать больше места на сервере. В этом разделе рассмотрим различные методы сжатия и оптимизации файлов изображений и видео, а также примеры и таблицы для иллюстрации этих концепций.

1. Сжатие изображений

- *Сжатие без потерь.* Используется для сжатия изображений без потери качества. При этом сжатии можно исполь-

зовать форматы без потерь, такие как PNG или без потерь сжатия в формате JPEG.

- *Сжатие с потерями.* Этот метод сжатия приводит к потере некоторого качества изображения, но позволяет значительно уменьшить размер файла. Формат JPEG с потерями является наиболее популярным форматом для сжатия фотографий с потерями.

Таблица 1.2

Пример сравнения форматов изображений и методов сжатия

<i>Формат изображения</i>	<i>Сжатие без потерь</i>	<i>Сжатие с потерями</i>	<i>Поддержка прозрачности</i>
PNG	Да	Нет	Да
JPEG	Нет	Да	Нет
WebP	Да	Да	Да

2. Оптимизация размера видео

- *Изменение разрешения.* Уменьшение разрешения видео может существенно сократить размер файла. Оптимальное разрешение зависит от требований проекта и устройств, на которых будет воспроизводиться видео.
- *Сжатие видео.* Использование сжатия видео с помощью кодеков, таких как H.264 или VP9, позволяет сократить размер видеофайла при сохранении приемлемого качества.
- *Предзагрузка видео.* Применение техники предзагрузки видео позволяет начать загрузку видеофайла заранее, что снижает время ожидания и улучшает пользовательский опыт (см. Табл. 1.3).

3. Инструменты для сжатия и оптимизации файлов

- *ImageOptim.* Это популярный инструмент для оптимизации файлов изображений без потери качества. Он автом

Таблица 1.3

Пример сравнения разрешений видео и размеров файлов

<i>Разрешение</i>	<i>Размер (без оптимизации)</i>	<i>Размер (с оптимизацией)</i>
1920 × 1080	50 МБ	20 МБ
1280 × 720	30 МБ	10 МБ
640 × 360	15 МБ	5 МБ

матически удаляет метаданные и применяет сжатие без потерь.

- *TinyPNG*. Сервис TinyPNG предлагает сжатие изображений в формате PNG с потерями, сохраняя хорошее качество при существенном уменьшении размера файла.
- *FFmpeg*. Это мощный инструмент для обработки видеофайлов. Он позволяет изменять разрешение, применять сжатие и оптимизировать видео.

Пример использования инструмента ImageOptim для оптимизации файлов изображений (см. Рис. 1.1).

В результате оптимизации размер файла изображения был сокращен до 500 КБ без потери качества.

Сжатие и оптимизация файлов изображений и видео играют важную роль в улучшении производительности и оптимизации загрузки веб-приложений. Правильный выбор формата и методов сжатия, а также использование соответствующих инструментов, помогут снизить размер файлов и улучшить пользовательский опыт.

ИСПОЛЬЗОВАНИЕ ЛЕНИВОЙ ЗАГРУЗКИ ДЛЯ УЛУЧШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ

Использование ленивой загрузки (Lazy Loading) — это техника оптимизации производительности веб-приложений, которая позволяет загружать ресурсы только по мере необходимости, вместо загрузки всех ресурсов сразу при открытии страницы.

Рисунок 1.1

```
makefile

До оптимизации:
Имя файла: image.jpg
Размер: 1.5 МБ
Формат: JPEG

После оптимизации:
Имя файла: image_optimized.jpg
Размер: 500 кб
Формат: JPEG
```

Ниже представлено описание, таблица с примерами ресурсов для ленивой загрузки и их эффект на производительность, а также примеры применения ленивой загрузки.

Использование ленивой загрузки является эффективным способом улучшить производительность веб-приложений, особенно при наличии большого количества ресурсов на странице, таких как изображения, видео, скрипты и стили. Вместо загрузки всех этих ресурсов при открытии страницы, ленивая загрузка позволяет загружать только те ресурсы, которые видимы на экране или будут использованы в ближайшем будущем. Это сокращает время загрузки страницы и снижает нагрузку на сервер (см. Табл. 1.4).

Примеры применения ленивой загрузки.

1. **Ленивая загрузка изображений.** Веб-приложение использует технику ленивой загрузки для изображений на странице. При прокрутке страницы изображения загружаются только тогда, когда они становятся видимыми для пользователя. Это снижает время загрузки страницы и повышает общую производительность приложения.

Таблица 1.4

Примеры ресурсов для ленивой загрузки и их эффект на производительность

<i>Ресурс</i>	<i>Тип</i>	<i>Эффект на производительность</i>
Изображения	JPEG, PNG, GIF	Загрузка изображений только при прокрутке к ним, что ускоряет загрузку страницы и уменьшает объем передаваемых данных.
Видео	MP4, WebM	Загрузка видео только при необходимости проигрывания, что позволяет уменьшить время загрузки страницы и снизить использование сетевого трафика.
Скрипты	JavaScript	Загрузка скриптов только при необходимости, например, при событиях пользователя или по мере прокрутки страницы.
Стили	CSS	Загрузка стилей только для видимых элементов на странице, что позволяет ускорить отображение и уменьшить объем передаваемых данных.

2. *Ленивая загрузка видео.* Веб-приложение применяет ленивую загрузку для видео на странице. Вместо загрузки всех видео сразу, видео загружаются только при необходимости воспроизведения. Это сокращает время загрузки страницы и экономит сетевой трафик.

3. *Ленивая загрузка скриптов.* Веб-приложение использует ленивую загрузку скриптов для оптимизации производительности. Скрипты загружаются только при событиях пользователя или по мере прокрутки страницы. Это ускоряет загрузку страницы и повышает отзывчивость приложения.

4. *Ленивая загрузка стилей.* Веб-приложение применяет ленивую загрузку стилей для оптимизации производительности. Стили загружаются только для видимых элементов на странице, что ускоряет отображение и снижает объем передаваемых данных.

Применение ленивой загрузки веб-приложений позволяет сократить время загрузки страницы, снизить нагрузку на сервер и улучшить общую производительность приложения. Определение и загрузка только необходимых ресурсов повышает отзывчивость приложения и улучшает пользовательский опыт.

ОПТИМИЗАЦИЯ РЕСУРСОВ ДЛЯ РАЗЛИЧНЫХ УСТРОЙСТВ И РАЗРЕШЕНИЙ ЭКРАНОВ

Выбор оптимального формата изображений и видео является важной задачей при разработке веб-приложений. Различные форматы имеют свои особенности, преимущества и недостатки, и выбор правильного формата зависит от типа контента, требуемого качества, поддерживаемых функций и ограничений среды, в которой будет использоваться приложение.

Один из наиболее распространенных форматов изображений — JPEG (Joint Photographic Experts Group). Он обеспечивает хорошую компромисс между качеством и размером файла. JPEG подходит для фотографий и изображений с множеством цветов и плавных переходов. Он использует алгоритм сжатия с потерями, что может привести к некоторой потере качества, особенно при высокой степени сжатия.

Еще один популярный формат — PNG (Portable Network Graphics). PNG подходит для изображений с прозрачностью и логотипов с плоскими цветами. Он использует без потерь сжатие, что означает, что качество изображения сохраняется без видимых потерь. Однако PNG-файлы могут быть большего размера по сравнению с JPEG, особенно для изображений с большим количеством цветов.

Формат GIF (Graphics Interchange Format) обычно используется для создания анимированных изображений или небольших пиктограмм. Он поддерживает анимацию и прозрачность, но ограничен в количестве цветов и может иметь большой размер файла.

WebP — относительно новый формат, разработанный Google, который обеспечивает хорошее сочетание между качеством и размером файла. WebP поддерживает как сжатие с потерями,

так и без потерь и может быть эффективным для различных типов изображений. Однако поддержка WebP в различных браузерах может быть ограничена.

При выборе формата видео, одним из наиболее распространенных является формат MP4 (MPEG-4 Part 14). Он обеспечивает хорошее сжатие и качество видео и широко поддерживается различными платформами и устройствами. Для видео с прозрачностью можно использовать формат WebM.

Важно учитывать требования проекта и целевую аудиторию при выборе формата изображений и видео. Если качество является наивысшим приоритетом, следует использовать форматы без потерь, такие как PNG или WebP без сжатия. Если важен малый размер файла и достаточное качество, то можно выбрать JPEG или WebP с потерями. Для анимаций или прозрачных изображений можно использовать формат GIF или WebP.

Также необходимо учитывать совместимость форматов с различными браузерами и устройствами. Некоторые форматы могут не поддерживаться в старых версиях браузеров или на некоторых plataформах, поэтому важно убедиться в доступности формата для целевой аудитории.

Оптимизация ресурсов для различных устройств и разрешений экранов является важной задачей веб-разработки.

Таблица 1.5

Совместимость разрешений экранов

<i>Разрешение экрана</i>	<i>Тип устройства</i>
320 × 480	Смартфоны и планшеты
375 × 667	Смартфоны и планшеты
768 × 1024	Планшеты и ноутбуки
1024 × 768	Планшеты и ноутбуки
1366 × 768	Ноутбуки и десктопы
1920 × 1080	Десктопы и ноутбуки

Конец ознакомительного фрагмента.

Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru