

## ХРАНИЛИЩЕ ДАННЫХ ДЛЯ WEB ПРИЛОЖЕНИЙ

### Аннотация

Данная статья посвящена анализу возможностей хранения данных для web - приложений.

### Ключевые слова

web, web - приложения, web - технологии, протоколы, программирование.

Интернет - соединения могут быть нестабильными или отсутствовать на ходу, поэтому офлайн - поддержка и надежная работа являются обычными особенностями прогрессивных веб - приложений. Даже в идеальных беспроводных средах разумное использование кэширования и других методов хранения может существенно улучшить взаимодействие с пользователем.

Для сетевых ресурсов, необходимых для загрузки вашего приложения и файлового контента, используйте Cache Storage API [1] (часть сервис - воркеров).

Для других данных используйте IndexedDB (с оболочкой обещаний).

IndexedDB и Cache Storage API поддерживаются всеми современными браузерами. Они оба асинхронны и не блокируют основной поток. Они доступны из объекта window, веб - воркеров и сервисных воркеров, что упрощает их использование в любом месте кода.

В браузере доступно несколько других механизмов хранения, но они имеют ограниченное использование и могут вызвать серьезные проблемы с производительностью.

SessionStorage зависит от вкладки и ограничен временем существования вкладки. Это может быть полезно для хранения небольших объемов информации о сеансе, например ключа IndexedDB [2]. Его следует использовать с осторожностью, поскольку он синхронный и блокирует основной поток. Он ограничен примерно 5 МБ и может содержать только строки. Поскольку он зависит от вкладки, он недоступен для веб - воркеров или сервис - воркеров.

Следует избегать использования LocalStorage, поскольку он синхронный и блокирует основной поток. Он ограничен примерно 5 МБ и может содержать только строки. LocalStorage недоступен для веб - воркеров или сервис - воркеров.

Файлы cookie имеют свое применение, но не должны использоваться для хранения. Файлы cookie отправляются с каждым HTTP - запросом, поэтому хранение чего - либо, кроме небольшого количества данных, значительно увеличит размер каждого веб - запроса. Они синхронны и недоступны для веб - воркеров. Как и LocalStorage и SessionStorage, файлы cookie ограничены только строками.

File System API и FileWriter API предоставляют методы для чтения и записи файлов в изолированную файловую систему. Хотя это асинхронно, это не рекомендуется, потому что оно доступно только в браузерах на основе Chromium.

File System Access API был разработан, чтобы упростить пользователям чтение и редактирование файлов в их локальной файловой системе. Пользователь должен предоставить разрешение, прежде чем страница сможет читать или записывать в любой локальный файл, и при этом разрешения не сохраняются между сеансами.

Не следует использовать WebSQL [3], а существующее использование следует перенести на IndexedDB. Поддержка была удалена почти из всех основных браузеров. W3C прекратил поддержку спецификации Web SQL в 2010 году, и дальнейших обновлений не планируется.

Кэш приложения не следует использовать, а существующее использование следует перенести на сервис - воркеры и Cache API так как он устарел, и в будущем поддержка в браузерах будет прекращена.

Во время разработки вы можете использовать DevTools своего браузера для проверки различных типов хранилищ и легко очистить все сохраненные данные.

В Chrome 88 была добавлена новая функция, которая позволяет переопределить квоту хранилища сайта на панели хранилища. Эта функция дает вам возможность моделировать различные устройства и тестировать поведение ваших приложений в сценариях с низкой доступностью диска. Перейдите в Application, затем Storage, установите флажок Simulate custom storage quota и введите любое допустимое число для имитации квоты хранилища. Пример внешнего вида данного окна представлена на рисунке 1.

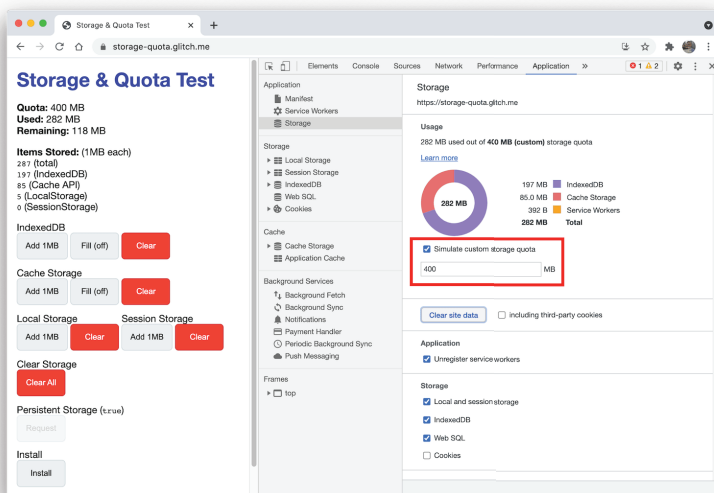


Рисунок 1 – Переопределение квоты хранения данных

Прошли те времена, когда хранилище было ограничено, и пользователю предлагалось хранить все больше и больше данных. Сайты могут эффективно хранить все ресурсы и данные, необходимые для работы. Используя StorageManager API, вы можете определить, сколько вам доступно и сколько вы использовали. А с постоянным хранилищем, если пользователь не удалит его, вы можете защитить его от потери заэкшированной информации.

### **Список использованных источников и литературы:**

1. Cache Storage API [Электронный ресурс]. URL: <https://developer.mozilla.org/ru/docs/Web/API/CacheStorage> (Дата обращения 24.12.2020).
2. LocalStorage и sessionStorage [Электронный ресурс]. URL: <https://learn.javascript.ru/localstorage> (Дата обращения 24.12.2020).
3. WEBSQL [Электронный ресурс]. URL: <http://html5ru.com/websql-relyacionnaya-baza-dannyx-na-veb-stranice.html> (Дата обращения 24.12.2020).
4. StorageManager API [Электронный ресурс]. URL: <https://developer.mozilla.org/ru/docs/Web/API/StorageManager>