МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**Факультет программной инженерии и компьютерной техники**

**Направление (специальность) ⎯ 09.04.02 Информационные системы и технологии**

**Специализация ⎯ Веб-технологии**

**Дисциплина — Исследование экосистем веб-языков и веб-технологий**

**Курсовой проект (работа)**

**ТЕМА: Исследование веб-хранилищ данных типа «ключ-значение»**

(Наименование сайта)

ВЫПОЛНИЛ

Студент группы   P4162 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шувалов Д.М.

№  группы подпись, дата ФИО

ПРОВЕРИЛ к.п.н., доцент ФПИиКТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Государев И.Б.

ученая степень, должность подпись, дата ФИО

**САНКТ-ПЕТЕРБУРГ**

**2019г.**

Оглавление

[введение 3](#_Toc10405566)

[1. Браузерные хранилища данных 5](#_Toc10405567)

[2. Web storage 6](#_Toc10405568)

[3. web db (Indexed db) 9](#_Toc10405569)

[Заключение 12](#_Toc10405570)

[Список литературы 14](#_Toc10405571)

# введение

Двадцать первый век – век информационных технологий, создания и развития глобальных Интернет гигантов. Интернет в первое десятилетие двадцать первого века из военной разработки превратился в коммуникационную, информационную, образовательную и развлекательную платформу для всего мира.

Современный Интернет невозможно представить без онлайн-магазинов, сайтов-агрегаторов туристических путевок, билетов на различные средства передвижения или массовые мероприятия (концерты, шоу, кино, театральные представления и т.п.), образовательных платформ, интернет-СМИ и т.д. В основе перечисленных видов веб-сайтов лежит клиент-серверная архитектура, где с одной стороны в качестве клиента выступает пользователь и браузер, а с другой стороны веб-сервер, выполняющий различные вычисления в ответ на запросы или действия пользователя в веб-приложении.

За счет своего развития в течение последних лет и значительного роста в производительности современные веб-браузеры позволяют реализовать клиент-серверные веб-приложения, решающие сложные задачи высоконагруженных систем, связанных с банковскими, трейдинговыми, проектными, финансовыми и другими сферами мирового бизнеса.

Традиционно данные веб-приложения хранятся на веб-сервере, например, конфиденциальная информация, личные данные, данные банковских карт, совершенных транзакциях и т.д. Такая система хранения данных является актуальной и по сей день – она надежная, безопасная и эффективная. Но современные веб-приложения также зависят от метаданных и второстепенной информации, которые легче хранить на клиенте, то есть в веб-браузере пользователя. Такой информацией, например, могут быть пользовательские настройки (язык, цветовая тема) или состояние приложения (снимок текущего состояния веб-приложения), чтобы посетитель мог продолжить его выполнение с того же самого места позже. [1]

До выхода стандарта HTML5 единственные хранилищем являлись cookies. Поскольку cookies были разработаны для обмена небольшими объемами идентифицирующей информации между веб-серверами и веб-браузерами и имеют ряд ограничений связанными с отсутствием нативного JavaScript API для работы с cookies данными и их временем хранения. [1][2]

В стандарте HTML5 вводятся концепция браузерного хранилища (web storage), которая включают в себя localStorage и sessionStorage. [3][4][5] Также вне стандарта HTML5 появилась связанная технология – индексированная база данных (IndexedDB) API. [6]

**Целью курсовой работы** является исследование веб-хранилищ данных типа ключ-значение.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучить официальные технологические стандарты для браузерных веб-хранилищ.
2. Исследовать возможности веб-хранилища.
3. Определить преимущества и недостатки для каждого типа веб-хранилищ.
4. Исследовать технологии объединяющие возможности веб-хранилищ.

# Браузерные хранилища данных

Изначально web storage (*далее* веб-хранилище) были частью спецификации HTML5, но сейчас имеют отдельную спецификацию. [7]

Согласно стандарту, веб-хранилище состоит из двух компонентов: sessionStorage и localStorage. Session storage представляет временное хранилище информации, которая удаляется после закрытия браузера. Local storage представляет хранилище для данных на постоянной основе. Функциональность веб-хранилища позволяет веб-приложением сохранять данные на компьютере пользователя.

IndexedDB – это новая концепция, которая была представлена отдельной спецификацией в рамках публикации HTML5 стандарта. Технология IndexedDB является более мощным хранилищем, чем веб аналоги (localStorage и sessionStorage), и полезен для приложений, которым требуется хранить большой объем данных на стороне клиента. [6][8] IndexedDB был представлен W3C для замены устаревшего WebSQL (спецификация по данной технологии больше не поддерживается – *прим.*). [9]

IndexedDB можно охарактеризовать как полноценную браузерную базу данных, она обладает схожими преимуществами, что и традиционные реляционные базы данных: структурированность, поддержка транзакций, индексация полей, кэширование данных и т.д. Также стоить отметить, что в виду ограничений для веб-хранилища на объем хранимых данных, IndexedDB не обладает таковым, а ограничивается объемом доступной памяти на компьютере пользователя.

Локальные хранилища являются NoSQL подобными, так как хранят данные в виде ключ-значение и хранят данные в виде строк или объектов.

# Web storage

Локальные хранилища, встроенные в веб-браузеры, такие как localStorage, sessionStorage и cookies имеют узконаправленную сферу применения.

SessionStorage хранит данные ограниченное время, они удаляются сразу после того, как пользователь завершает свой сеанс работы с веб-приложением или закроет браузер.

LocalStorage хранит данные, которые не имеют ограничений по времени и сеанса. Данные, хранящиеся в localStorage, доступны при каждом посещении сайта и хранятся на компьютере, ноутбуке и других устройствах с которого осуществляется вход.

SessionStorage и localStorage имеют удобное JavaScript API для чтения и записи данных, но имеют ограничение в объеме хранилища 5-10 Мб в зависимости от браузера. [10]

Cookies в свою очередь являются инструментом для хранения различных конфиденциальных данных для веб-приложения: CSRF токен, JWT, идентификаторы пользователей и т.д. [10] Cookies являются безопасным хранилищем для персональных данных, cookie данные хранятся ограниченное время (в конце сеанса или по истечении указанной даты). [11] Из минусов cookies стоит отметить отсутствие нативного API для работы с данными, а также небольшой размер хранилища – 4 Кб.

SessionStorage, localStorage и cookies хранят данные с помощью элементов (пар "ключ-значение"). Ключ представляет собой некоторый идентификатор, который связан со значением. То есть для того, чтобы записать или получить некоторое значение необходимо знать его ключ. Значение представляет собой строку, это необходимо учитывать при работе с ними в коде JavaScript.

В таблице 1 представлено сравнение веб-хранилища и cookies.

Таблица 1. Сравнение веб-хранилища и cookies

|  |  |  |
| --- | --- | --- |
| **Критерий сравнения/**  **Тип хранилища** | **Cookie** | **localStorage и sessionStorage** |
| **Где храниться?** | На компьютере пользователя | На компьютере пользователя |
| **Доступность данных** | Данные доступны как со стороны сервера, так и со стороны клиента | Данные доступны только со стороны браузера посредством JavaScript API |
| **Максимальный размер данных** | 4 Кбайт | 5 или 10 Мбайт в зависимости от браузера |
| **Удобство методов для работы с данными** | Отсутствие нативного API | JavaScript API |
| **Время хранения данных** | Ограниченное время (в конце сеанса или по истечении указанной даты) | Неограниченное время (localStorage) или после завершения сеанса (sessionStorage) |
| **Кроссбраузерность** | Все браузеры кроме OperaMini (покрытие 98%) | |

Скорость чтения/записи данных c веб-хранилища и cookies практически одинаковая в среднем разница составляет 1-2 мс.

Программные интерфейсы localStorage и sessionStorage ограничивают доступ к данным тем доменом с учетом протокола и номера порта, в котором находится данная страница. То есть данные веб-приложения доступны только тем веб-страницам, которые принадлежат домену. Страницы, которые не расположены в рамках домена не могут прочитать или удалить данные хранилища. Данное преимущество превращается в недостаток, если при обращении к зараженному веб-приложению или веб-сайту вместе со статичными файлами, пользователь может получить вредоносный JavaScript-код, который может беспрепятственно получить доступ к хранилищу, поскольку никакой защиты для такого сценария не предусмотрено.

Для того, чтобы хранить конфиденциальные/авторизационные данные на стороне клиента используются cookies. Cookies является криптографически зашифрованным хранилищем конфиденциальной информации. [10] Cookies не имеет собственного JavaScript API, но на просторах интернета достаточно информации как организовать работу с данным хранилищем. [2] Также при определенных настройках cookies на стороне веб-сервера, можно полностью ограничить доступ к ним на стороне клиента. [10]

Существенным недостатком при работе с localStorage и sessionStorage является их синхронность. Это означает, что каждая операция, связанная с хранилищем, будет выполняться последовательно. Для сложных приложений это критично, поскольку может замедлить скорость его работы. Данное ограничение не позволяет использовать хранилища для web workers, которые предоставляют преимущества фоновой обработки для производительности (расширения для Chrome, ServiceWorker и пр.). [10]

Для хранения сложных данных, например, JSON, объекты, массивы объектов, а также данных не являющихся конфиденциальными, лучше использовать встроенные в веб-браузеры браузерные базы данных – Web DB.

# web db (Indexed db)

IndexedDB – это низкоуровневый API-интерфейс для хранения структурированных данных на стороне клиента, включая файлы и большие объекты. IndexedDB является хранилищем ключей/значений (база данных NoSQL), которое использует индексы для обеспечения высокопроизводительного поиска данных.

IndexedDB была представлена W3C в рамках стандарта HTML5 для замены устаревшей технологии WebSQL. [4][8] IndexedDB является более мощным и функциональным, чем локальное хранилище, и полезен для приложений, которым требуется хранить большой объем данных. [6]

Среди основных преимуществ данного стандарта можно отметить следующие качества:

* **Не зависит от доступности сети.** Хранилище имеет полную функциональность, даже если подключение к сети недоступно, а также не будет терять данные при разрывах сетевого соединения;
* **Кэширование.** За счет индексов кэширует как запросы, так и данные;
* **Меньше трафика.** Генерирует меньше трафика чем WebSQL база данных;
* **Скорость**. Операции записи/чтения происходят мгновенно, так как манипуляции с данными происходят на компьютере, а не на веб-сервере;
* **Асинхронность**. IndexedDB является асинхронным API. Это означает, что выполнение приоритетных операций не будет блокировать поток пользовательского интерфейса; [13]
* **Гибкость данных.** Позволяет хранить объекты в отличие от строкового формата для веб-хранилища и cookies. [5][10]

Основными терминами, на которых основан стандарт IndexedDB, являются хранилище объектов, индекс, транзакция и курсор.

**Хранилище объектов** – главный механизм хранения данных в базе данных IndexedDB. Любая база данных представляете коллекцию именованных хранилищ объектов. Каждый объект должен иметь ключ, под которым он сохраняется и извлекается из хранилища. Ключи должны быть уникальными, и они должны иметь естественный порядок следования, чтобы их можно было сортировать. IndexedDB может автоматически генерировать уникальные ключи для каждого объекта, добавляемого в базу данных. Однако, часто объекты, сохраняемые в хранилище объектов, уже будут иметь свойство, пригодное для использования в качестве ключа. В этом случае при создании хранилища объектов достаточно просто определить ключевое свойство. [13][14]

Помимо возможности извлекать объекты из хранилища по значению первичного ключа существует также возможность выполнять поиск по значениям других свойств объекта. Чтобы обеспечить эту возможность, в хранилище объектов можно определить любое количество **индексов**. Каждый индекс определяет вторичный ключ хранимых объектов. Эти индексы в целом могут быть неуникальными, и одному и тому же ключу может соответствовать множество объектов. [13][15]

**Транзакция** является оберткой вокруг операции или группы операций, которая обеспечивает целостность базы данных. Все операции чтения или записи в IndexedDB должны быть частью транзакции. Если одно из действий в транзакции завершается неудачно, ни одно из них не применяется, и база данных возвращается в состояние, в котором она находилась до начала транзакции. Это позволяет выполнять атомарные операции чтения-изменения-записи, не беспокоясь о других потоках, одновременно работающих с базой данных. [14]

В IndexedDB имеется поддержка **курсора**. Курсор – это специальный объект, который проходит через хранилище объектов по заданному запросу и возвращает один ключ/значение за раз, тем самым экономя память. Поскольку хранилище объектов отсортировано по ключу, курсор перемещается по хранилищу в порядке расположения ключей (по умолчанию в порядке возрастания). [6][14][15]

Также как веб-хранилище и cookies, каждая база данных IndexedDB уникальна для источника (обычно это домен или поддомен сайта), а значит, она не может получить доступ или быть доступной для любого другого источника.

В таблице 2 представлено сравнение веб-хранилища и IndexedDB.

Таблица 2. Сравнение веб-хранилища и IndexedDB

|  |  |  |
| --- | --- | --- |
| **API** | **Веб-хранилище** | **IndexedDB** |
| **Получение значений в порядке их ключей** | Да | Да |
| **Хранение нескольких значений на ключ** | Нет | Да, только в другом хранилище |
| **Сеансовое хранилище** | Да | Нет |
| **Асинхронность API** | Нет | Да |
| **Атомарность обновлений** | Нет | Да |
| **Транзакции** | Нет | Да |
| **Ограничение по типу данных** | Да | Нет |
| **Кроссбраузерность** | 98% (кроме Opera Mini) | 90% (кроме Opera Mini, частичная поддержка в IE и Edge) |

По состоянию на май 2019 года IndexedDB API имеет версию 1.0 с поддержкой в браузерах около 90%, а также версию 2.0 в качестве релиз кандидата с поддержкой в браузерах около 85%. [16]

# Заключение

С каждым годом популярность веб-приложений увеличивается, наблюдается переход от традиционных (десктопных) приложений к веб версиям. С ростом популярности растут потребности и требования к разрабатываемым продуктам, в подтверждение данного факта свидетельствует ежегодное появление новых фреймворков, различных библиотек и дополнительных инструментов для разработчиков, а также развитие и прогрессирование уже существующих технологий.

Стандарт HTML5 представил веб миру новый технологии и новые стандарты. Новый стандарт внес новый набор технологий и стандартов, позволяющих делать функциональные и гибкие веб-приложения и веб-сайта. Также новые спецификации предоставили разработчикам мощные инструменты для работы с возможностями современных браузеров и заложили основы для дальнейшего развития веба.

Одним из нововведений были веб-хранилище и веб базы данных. Согласно стандарту, веб-хранилище состоит из двух хранилищ: localStorage и sessionStorage. Изначально данная технология были частью стандарта HTML5, но в дальнейшем обзавелась отдельной спецификацией. На сегодняшний день веб-хранилище является гибким инструментов для хранения различных метаданных веб-приложений, а также кэширования пользовательских настроек или данных от сторонних API, например, результаты запросов Google Maps API. Веб-хранилище имеет превосходную поддержку в браузерах – 98%.

Также в рамках стандарта HTML5 был представлен стандарт индексируемой базы данных или IndexedDB. Данная технология предоставляет возможность хранения больших объемов данных в браузерах пользователей. Приложениям, которые отправляют большой объем данных по сети, данная технология дает возможность хранить данные на стороне клиента, тем самым предоставляя возможность кэширования или дублирования данных. Данная особенность позволяет использовать IndexedDB для разработки offline first веб-приложений. IndexedDB также предоставляет мощный API для поиска данных, работающий на основе индексов и курсоров, для получения необходимых данных. А наличие транзакции в реализации данной базы данных позволяет сохранить атомарность хранимых данных.

Современные веб-браузеры, а также технологии и стандарты, реализованные в них, позволяют реализовывать гибкие хранилища данных на стороне пользователя. Комбинация таких технологий, как Web Storage и IndexedDB, позволяет создавать эффективные веб-приложения, которые могут работать в offline, кэшировать пользовательские данные и метаинформацию.

# Список литературы

1. HTML5 Web Storage | Обзор веб-хранилища – Программирование на языке C#, платформа .NET Framework [Электронный ресурс]. Режим доступа: <https://professorweb.ru/my/html/html5/level5/5_1.php>, свободный (дата обращения: 25.05.2019). Язык доступа: русский.
2. Куки, document.cookie – Современный учебник JavaScript [Электронный ресурс]. Режим доступа: <https://learn.javascript.ru/cookie>, свободный (дата обращения: 25.05.2019). Язык доступа: русский.
3. HTML 5.2 – World Wide Web Consortium (W3C) [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/html5/>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.
4. HTML5 – Веб-документация MDN [Электронный ресурс]. Режим доступа: <https://developer.mozilla.org/ru/docs/HTML/HTML5>, свободный (дата обращения: 25.05.2019). Язык доступа: русский.
5. Web Storage API. Web APIs – Веб-документация MDN [Электронный ресурс]. Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.
6. Indexed Database API – World Wide Web Consortium (W3C) [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/2015/REC-IndexedDB-20150108/>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.
7. Web Storage (Second Edition) – World Wide Web Consortium (W3C) [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/webstorage/>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.
8. HTML5 — Википедия [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/HTML5>, свободный (дата обращения: 25.05.2019). Язык доступа: русский.
9. Web SQL Database – World Wide Web Consortium (W3C) [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/webdatabase/>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.
10. Почему не стоит использовать LocalStorage – Хабр [Электронный ресурс]. Режим доступа: <https://habr.com/ru/post/349164/>, свободный (дата обращения: 10.05.2019). Язык доступа: русский.
11. JavaScript - sessionStorage и localStorage – ИТ Шеф [Электронный ресурс]. Режим доступа: <https://itchief.ru/lessons/javascript/javascript-sessionstorage-and-localstorage>, свободный (дата обращения: 10.05.2019). Язык доступа: русский.
12. Использование IndexedDB [Электронный ресурс]. Режим доступа: <http://cultofdigits.com/dart-language/ispolzovanie-indexeddb/>, свободный (дата обращения: 10.05.2019). Язык доступа: русский.
13. Готовим IndexedDB – Хабр [Электронный ресурс]. Режим доступа: <https://habr.com/ru/post/213515/>, свободный (дата обращения: 10.05.2019). Язык доступа: русский.
14. Working with IndexedDB – Google Developers [Электронный ресурс]. Режим доступа: <https://developers.google.com/web/ilt/pwa/working-with-indexeddb>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.
15. Работа с базой IndexedDB – Free How-To Tutorials & Online Courses by Envato Tuts+ [Электронный ресурс]. Режим доступа: <https://code.tutsplus.com/ru/tutorials/working-with-indexeddb--net-34673>, свободный (дата обращения: 25.05.2019). Язык доступа: русский.
16. Can I use... Support tables for HTML5, CSS3, etc [Электронный ресурс]. Режим доступа: <https://caniuse.com/#search=indexeddb>, свободный (дата обращения: 25.05.2019). Язык доступа: английский.