



EBook Gratis

APRENDIZAJE indexeddb

Free unaffiliated eBook created from
Stack Overflow contributors.

#indexeddb

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con indexeddb	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Esquema de DB indexado.....	2
Visión general.....	3
Lo esencial	3
Async vs Sync	4
Apoyo	4
Aprende más	4
Resumen de solicitud de base de datos indexada.....	4
Capítulo 2: Ejemplos de DB indexados	6
Examples.....	6
Abrir una base de datos.....	6
Inicializar base de datos - crear tabla - con versión conocida de db.....	6
Inicializar base de datos - crear tabla sin saber la versión de db.....	6
Creditos	8

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [indexeddb](#)

It is an unofficial and free indexeddb ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official indexeddb.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con indexeddb

Observaciones

Esta sección proporciona una descripción general de qué es indexeddb y por qué un desarrollador puede querer usarlo.

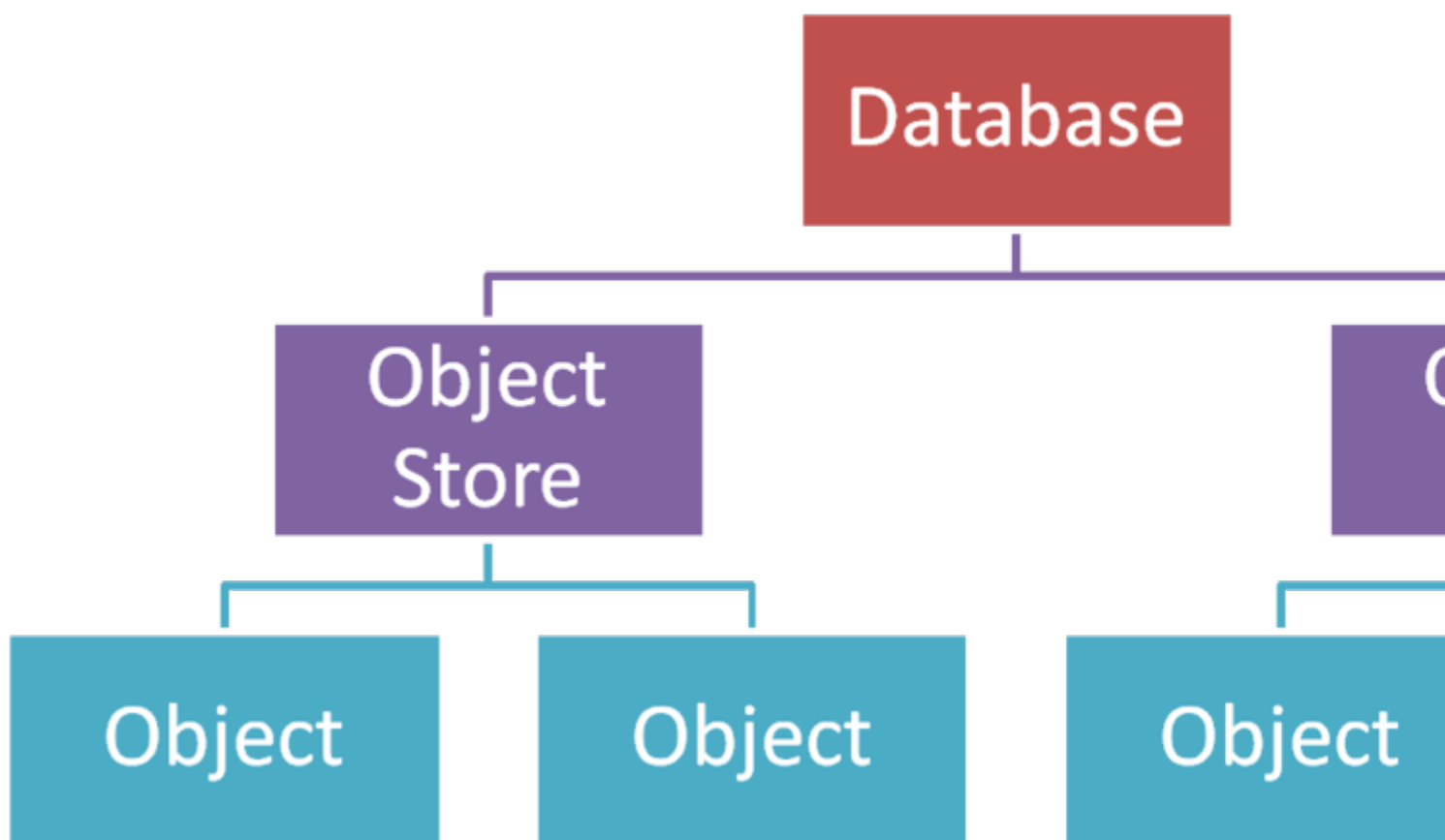
También debe mencionar cualquier tema grande dentro de indexeddb, y vincular a los temas relacionados. Dado que la Documentación para indexeddb es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas para configurar o instalar indexeddb.

Esquema de DB indexado



Como se puede ver en la imagen de arriba, en una sola aplicación podemos crear:

- Múltiples bases de datos

- Cada base de datos puede tener múltiples almacenes de objetos (tablas)
- Cada almacén de objetos puede tener almacenados múltiples objetos.

Visión general

IndexedDB es una API de bajo nivel para el almacenamiento del lado del cliente de cantidades significativas de datos estructurados, incluidos archivos / blobs. Esta API utiliza índices para permitir búsquedas de alto rendimiento de estos datos. Si bien el almacenamiento web es útil para almacenar cantidades más pequeñas de datos, es menos útil para almacenar cantidades más grandes de datos estructurados. El estándar IndexedDB se creó para permitir el almacenamiento y la recuperación escalables y eficaces de objetos Javascript en un navegador.

Lo esencial

indexedDB está diseñado para almacenar literales de objetos Javascript como `{prop1 : value, prop2 : value}`. Además, las implementaciones más recientes admiten el almacenamiento de objetos binarios grandes (BLOB), como imágenes, archivos de audio y archivos de video. Además, indexedDB puede almacenar objetos que contienen otros objetos (objetos anidados), como `{prop1 : value, prop2 : {nestedprop1 : value, nestedprop2 : value}}`.

Los siguientes son algunos conceptos básicos:

- **Base de datos** : un contenedor de almacenes de objetos e índices. Cada base de datos tiene un nombre y una versión.
- **Almacén de objetos** : un contenedor de objetos. Esto es análogo a una tabla en una base de datos relacional. En indexedDB, los registros corresponden a objetos de Javascript y las columnas corresponden a las propiedades de objetos de Javascript. Los objetos agregados a la tienda se almacenan en el orden agregado. Las consultas contra la tienda recuperan objetos en el mismo orden. Puede insertar, actualizar o eliminar objetos en un almacén de objetos.
- **Índice** : un contenedor especial para objetos específicos contenidos en un almacén de objetos. Los índices también son análogos a las tablas, y pueden entenderse como almacenes de objetos con restricciones especiales. Cuando se inserta un objeto en un almacén de objetos, puede, si cumple con ciertos criterios, también insertarse en un almacén de índice correspondiente. Los objetos en un índice se almacenan en un orden definido por el índice. Las consultas contra un índice recuperan objetos en el orden definido por el índice (aunque las consultas se pueden configurar para que funcionen de manera diferente). No puede insertar, actualizar o eliminar objetos en un índice (solo puede hacerlo indirectamente insertando el objeto en la tienda en la que se basa el índice).
- **Cursor** : los cursores son análogos a las consultas. Un cursor recorre los objetos en un almacén de objetos o en un índice. Los cursores pueden moverse hacia delante o hacia atrás, buscar (saltar o avanzar objetos pasados) y saltar al objeto "único" anterior o siguiente en la tienda / índice subyacente.
- **Ruta clave**: las rutas clave son análogas a las claves primarias (o claves primarias compuestas) de una tabla en una base de datos relacional. En el caso general, cuando le indica a indexedDB que cree un almacén de objetos en una base de datos particular,

también define la ruta de acceso clave para el almacén. Puede usar la ruta de acceso de clave para obtener rápidamente un objeto en particular, que es similar a usar una clave principal para seleccionar un registro en una tabla relacional. Opcionalmente, puede usar claves para asegurarse de que los intentos posteriores de insertar un objeto en un almacén de objetos que ya contiene un objeto con la misma clave producirá un error.

- **Transacciones y solicitudes** : las solicitudes son análogas a las consultas SQL individuales. Existen métodos de API específicos para insertar un objeto, eliminar un objeto, actualizar un objeto e iterar sobre uno o más objetos. Cada llamada de método corresponde a una sola solicitud. Cada solicitud se produce dentro del contexto de una transacción. En otras palabras, múltiples solicitudes pueden ocurrir en una transacción. Las solicitudes individuales pueden fallar por una variedad de razones. Al realizar múltiples solicitudes en una sola transacción, las solicitudes no se confirman completamente hasta que todas las solicitudes se consideran exitosas. De esta manera, si se produce un problema en una solicitud posterior, la transacción completa se puede "revertir" para que el estado del almacén de objetos subyacente sea el mismo que antes de que se produjera la primera solicitud en la transacción.

Async vs Sync

La API de Javascript de indexedDB utiliza técnicas asíncronas. Cuando interactúa directamente con la API, y no con una biblioteca de terceros de nivel superior, la API requiere el uso de devoluciones de llamada de Javascript. El diseño asíncrono ayuda a evitar que las operaciones de procesamiento de datos más grandes bloqueen el subproceso principal de Javascript, lo que ayuda a evitar que la interfaz de usuario (lo que se ve en el navegador) aparezca congelada / entrecortada / desfigurada.

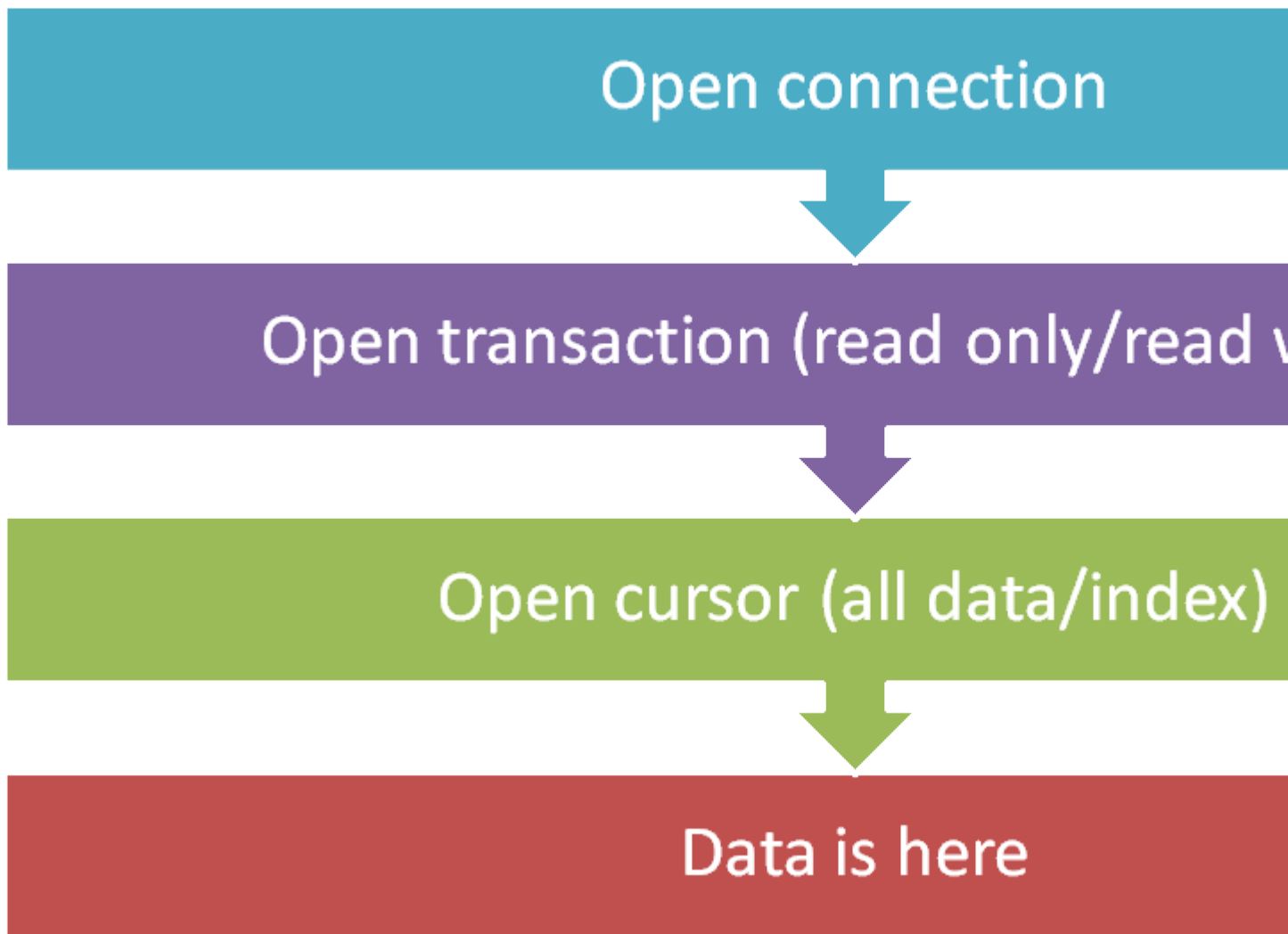
Apoyo

Visite <http://caniuse.com/#feat=indexeddb> .

Aprende más

- [Usando el artículo IndexedDB en MDN](#)
- [Especificaciones W3](#)
- [Ejemplo en html5rocks.com](#) . (**Advertencia:** este ejemplo está desactualizado, por ejemplo, utiliza `setVersion` lugar de `onupgradeneeded` , y por lo tanto puede que no se ejecute en navegadores modernos).

Resumen de solicitud de base de datos indexada



Como se puede ver en la imagen de arriba, en la base de datos indexada para acceder a los datos que necesita tener:

1. Abra una conexión a la base de datos deseada
2. Abra una transacción que puede ser solo de lectura o lectura de escritura
3. Abra un cursor o índice que se puede utilizar para filtrar los datos
4. En el evento de solicitud de cursor - onsuccess puede acceder a sus datos

Lea Empezando con indexeddb en línea:

<https://riptutorial.com/es/indexeddb/topic/10356/empezando-con-indexeddb>

Capítulo 2: Ejemplos de DB indexados

Examples

Abrir una base de datos

```
function openDatabase(dbName) {
  var request = indexedDB.open(dbName);
  request.onsuccess = function (e) {
    var database = request.result;
    if (database) {
      console.log("Database initialized.");
    } else {
      console.error("Database is not initialized!");
    }

    //Your code goes here
    database.close();
  }
  request.onerror = function (e) {
    console.error( e.target.error.message);
  }
}
```

Inicializar base de datos - crear tabla - con versión conocida de db

Para desencadenar un evento "actualizado", debe solicitar la base de datos con una versión más alta que la versión actual; de lo contrario, el evento no se activará.

```
function createTable(dbName, dbversion, tableName) {
  var request = indexedDB.open(dbName, dbversion);
  request.onupgradeneeded = function (e) {
    var database = e.target.result;
    var objectStore = database.createObjectStore(tableName, {
      keyPath: 'id'
    });
    console.log("Object Store Created");
  };
  request.onsuccess = function (e) {
    var database = e.target.result;

    //code to verify that the table was created
    database.objectStoreNames.contains(storeName);

    database.close();
  }
  request.onerror = function (e) {
    console.error(e.target.error.message);
  }
}
```

Inicializar base de datos - crear tabla sin saber la versión de db

Esta es una solución más genérica aplicable en el sistema donde el usuario tiene la opción de agregar índices a la tabla que usa:

```
function createTable(dbName, tableName) {
  var request = indexedDB.open(dbName);
  request.onsuccess = function (e){
    var database = e.target.result;
    var version = parseInt(database.version);
    database.close();
    var secondRequest = indexedDB.open(dbName, version+1);
    secondRequest.onupgradeneeded = function (e) {
      var database = e.target.result;
      var objectStore = database.createObjectStore(tableName, {
        keyPath: 'id'
      });
    };
    secondRequest.onsuccess = function (e) {
      e.target.result.close();
    }
  }
}
```

Lea Ejemplos de DB indexados en línea:

<https://riptutorial.com/es/indexeddb/topic/10446/ejemplos-de-db-indexados>

Creditos

S. No	Capítulos	Contributors
1	Empezando con indexeddb	Community , Deni Spasovski
2	Ejemplos de DB indexados	Deni Spasovski