Defining Scope In IndexedDB Transactions

By Ido Green

Ido is a Developer Advocate and a runner

What is IndexedDB? IndexedDB is an evolving web standard for storage of significant amounts of structured data in the browser and for high performance searches on this data using indexes. In other words, IndexedDB is an object store. It is not the same as a relational database, which has tables with collections rows and columns. It is an important and fundamental difference that affects the way that you design and build your applications (more on the - basic concepts).

So what is new? Changes my friends... we have some changes that are going to throw some errors if we don't handle them with simple syntax change.

From version 17 onwards, Chrome will now throw an error if an IndexedDB transaction is not scoped to an object store. Since all reading and writing of data are done within transactions, we need to create a transaction on a database, specify the scope (such as which object stores you want to access) and determine the kind of access (read only or write).

What does it means in code? Well, instead of passing an empty array to our database.transaction:

```
var transaction = db.transaction([], IDBTransaction.READ_ONLY);
```

You should scope to a particular object store, or list of object stores:

```
// all stores (equivalent to what use to be marked as empty array. )
var transaction = db.transaction(db.objectStoreNames, IDBTransaction.READ_ONLY);

// multiple stores:
var transaction = db.transaction(['ObjectStoreName1', 'ObjectStoreName2'],
    IDBTransaction.READ_ONLY);

// single store - these are equivalent
var transaction = db.transaction(['ObjectStoreName'], IDBTransaction.READ_ONLY);
var transaction = db.transaction('ObjectStoreName', IDBTransaction.READ_ONLY);
```

You can speed up data access by using the right scope and mode in the transaction. Here's a couple of tips: When defining the scope, specify only the object stores you need. This way, you can run multiple transactions with non-overlapping scopes concurrently. Only specify a `READ_WRITE` transaction mode when necessary. You can concurrently run multiple `READ_ONLY` transactions with overlapping scopes, but you can have only one `READ_WRITE` transaction for an object store.

Other sources:

- The update on chromium.org
- More details on IndexedDB at MDN
- A full example on how to use indexedDB in a simple ToDo list web app

So until next time... keep pushing the web to near territories.

Last updated July 2, 2018.