Quota Management API : Fast Facts

By Eiji Kitamura

Developer Advocate in Tokyo

Warning: This page is deprecated. Please see Persistent Storage instead.

There's various offline related features introduced to modern browsers through HTML5. While offline is convenient, its concept of quota has been left untouched for a long time. The latest version of Chrome browser has the first concept and its implementation of *Quota Management API*. It handles quota for AppCache, IndexedDB, WebSQL and File System API. Here's a list of things you should keep in mind when working with Quota Management API in the latest Chrome.

(The specific numbers and details noted below are not a part of HTML5 but the facts in the current Chrome implementation. The API and Quota management in Chrome is still evolving and the details may change over the time.)

- On HTML5 storage, there's a notion of `TEMPORARY` storage and `PERSISTENT` storage.
  - `TEMPORARY` storage can be used without requesting quota, but may be deleted at the browser's discretion.
  - `PERSISTENT` storage is never deleted without the user's instruction, but usually requires up-front quota request to use.
- For `TEMPORARY` storage, it is shared between all applications and websites run in the browser.
  - `TEMPORARY` storage has a default quota of 50% of available disk as a shared pool. (50GB => 25GB) (Not restricted to 1GB anymore)
    [To be more specific, TEMP quota is calucalated by *(remaining disk space + TEMP storage space) * 50%*. Therefore if apps are using 25GB TEMP storage in total and the current remaining disk space is 25GB that's already full.]
  - Each application has a limitation to have 20% of the available `TEMPORARY` storage pool (i.e. 20% of 50% of available disk). (Not restricted to 5Mb anymore)
  - When `TEMPORARY` storage quota is exceeded, *all the data (incl. AppCache, IndexedDB, WebSQL, File System API) stored for oldest used origin gets deleted* .
    [Note: since each app can only use up to 20% of the pool, this won't happen very frequently unless the user has more than 5 active offline apps.]

    - If app tries to make a modification which will result in exceeding `TEMPORARY` storage quota (20% of pool), an error will be thrown.
    - Each application can query how much data is stored or how much more space is available for the app by calling `queryUsageAndQuota()` method of Quota API.
    - Requesting more quota against `TEMPORARY` storage doesn't do anything.
    - Requesting quota for `TEMPORARY` storage using `webkitRequestFileSystem()` doesn't actually allocate / change quota.

  webkitStorageInfo.queryUsageAndQuota(webkitStorageInfo.TEMPORARY, // or PERSISTENT usageCallback, errorCallback);

- For `PERSISTENT` storage, its default quota is 0 and it needs to be explicitly requested per application using `requestQuota()` of Quota API.
    - The allocated space can be used only by File System API (for `PERSISTENT` type filesystem) and there's no such thing like `PERSISTENT` storage on *IndexedDB, WebSQL DB or AppCache* (yet).

  webkitStorageInfo.requestQuota(webkitStorageInfo.PERSISTENT newQuotaInBytes, quotaCallback, errorCallback);

- `unlimitedStorage` on manifest.json of Chrome Web App has been brought as a temporary solution for apps to work without Quota API. So, there's no guarantee that Chrome will support this feature forever.
- The API is described in WebIDL here: https://groups.google.com/a/chromium.org/group/chromium-html5/msg/5261d24266ba4366?pli=1

Last updated July 2, 2018.