

The HTML5 Filesystem API gives applications the facility to write and store actual files in JavaScript. That is amazing, but with great power comes great responsibility. Websites now have the potential to store large amounts of binary data on a user’s system. It is important that applications do not abuse such a gift by, for example, eating up large amounts of disk space without the user’s knowledge or consent. The last thing users want is to have 20 GB of data stored on their system just by visiting a URL.

At the time of writing, Chrome has a limited UI settings page for users to manage the storage space for applications that save data on their behalf. It is accessible via Preferences → Under the Hood → All Cookies and Site Data (or by opening *chrome://settings/cookies*). Users can only delete data from this menu. As a result of this limited UI, write operations (such as creating a folder and writing to a file) require an application to ask for the estimated size, in bytes, they expect to use. The same practice is true for other offline storage APIs, like WebSQL DB, where one opens a database with a particular size:

```
var db = window.openDatabase(
  'MyDB',           // dbName
  '1.0',           // version
  'test database',  // description
  2 * 1024 * 1024, // estimatedSize in bytes (2MB)
  function(db) {}  // optional creationCallback
);
```

## Storage Types

A normal web application can request storage space under two classifications: temporary or persistent. In addition to these types, Chrome Extensions ...