Opening a Filesystem

A web application obtains access to the HTML5 Filesystem by requesting a `LocalFileSystem` object using a global method, `window.requestFileSystem()`:

```
window.requestFileSystem(type, size, successCallback, opt_errorCallback)
```

Warning

This method is currently vendor prefixed as `window.webkitRequestFileSystem`.

Its parameters are described below:

`type`

: Whether the storage should be persistent. Possible values are `TEMPORARY` or `PERSISTENT`. Data stored using `TEMPORARY` can be removed at the browser's discretion (for example if more space is needed). `PERSISTENT` storage cannot cleared unless explicitly authorized by the user or the application.

`size`

: An indicator of how much storage space, in bytes, the application expects to need.

`successCallback`

: A callback function that is called when the user agent successfully provides a filesystem. Its argument is a `FileSystem` object.

`opt_errorCallback`

: An optional callback function which is called when an error occurs, or the request for a filesystem is denied. Its argument is a `FileError` object.

Calling `window.requestFileSystem()` for the first time creates a new sandboxed storage space for the app and origin that requested it. A filesystem is restricted to a single application and cannot access another application's stored data. This also means that an application cannot read/write files to an arbitrary folder on the user's hard drive (such as My Pictures or My Documents). Each filesystem is isolated.

Example 3-1. Requesting ...