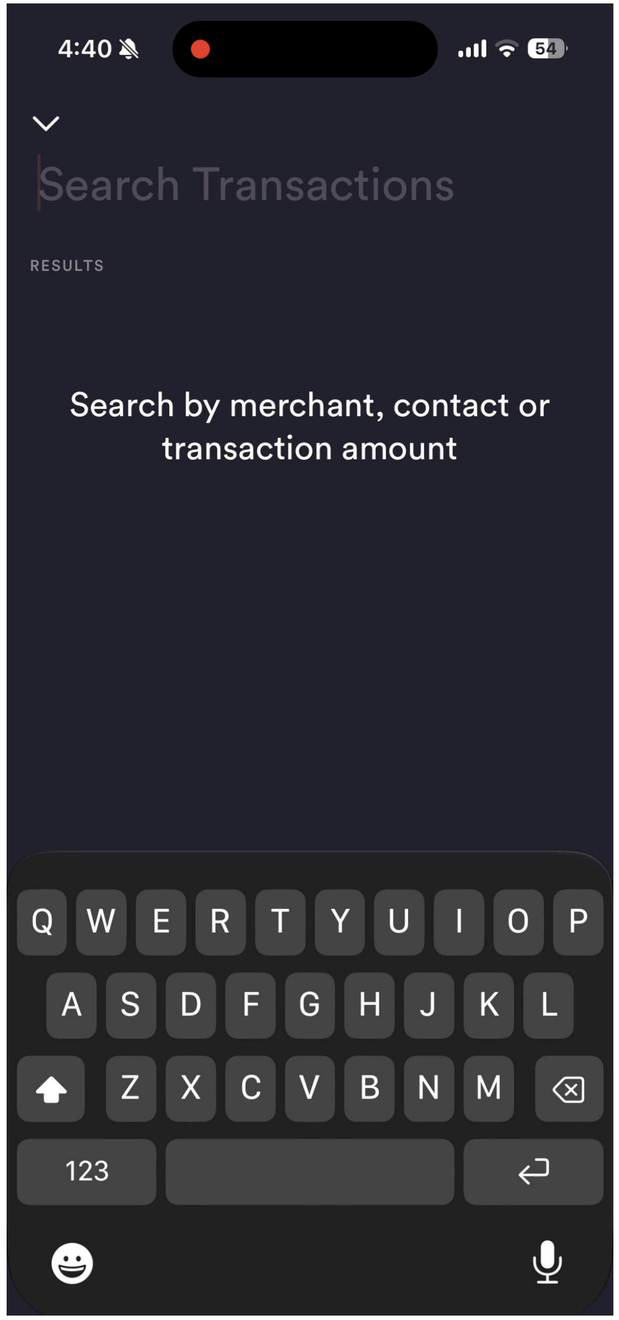
Local-first web apps

steveharrison.dev

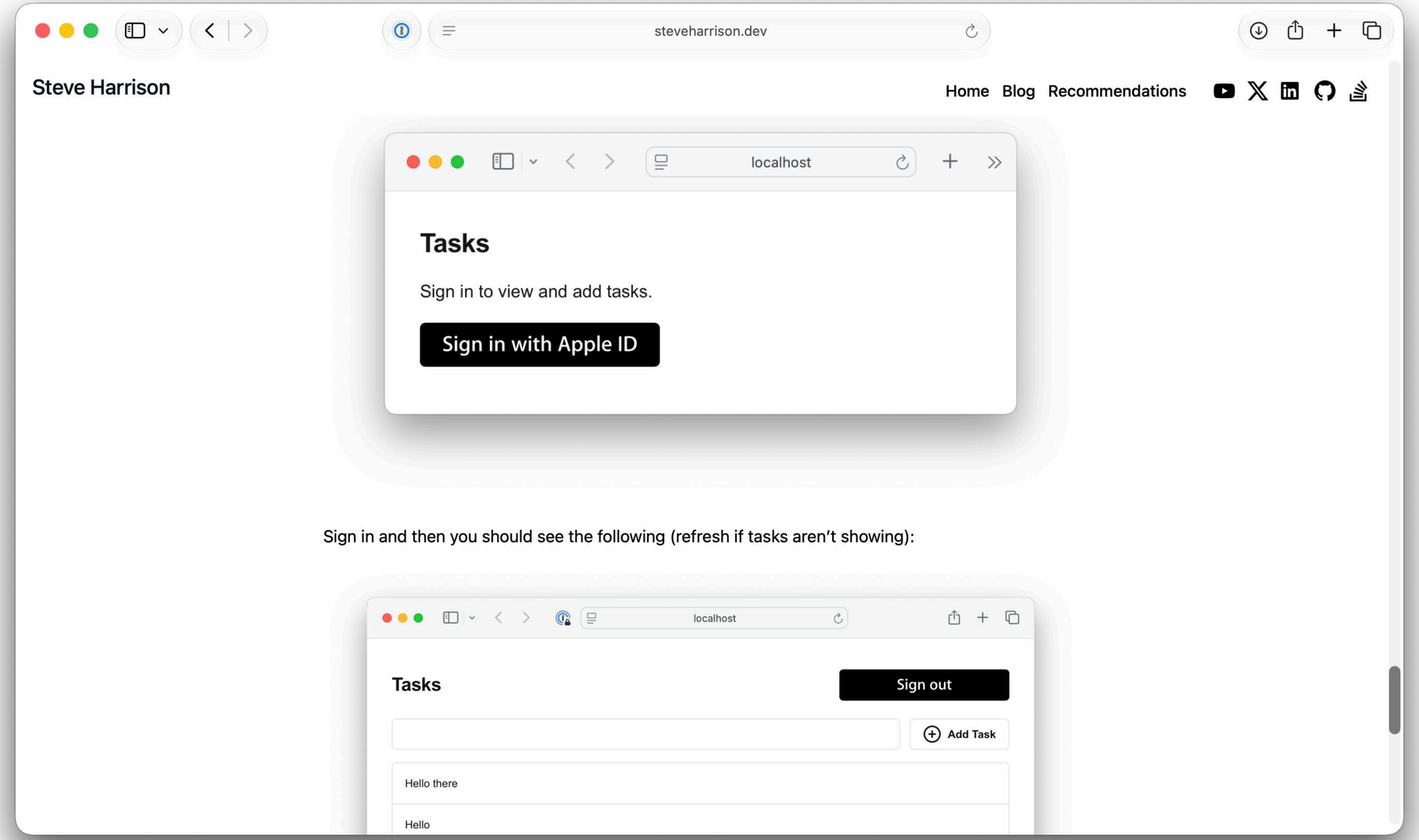
“How could I build a personal finance app where all the data is stored on my Mac?”


4:40
54
Search Transactions
RESULTS
Search by merchant, contact or transaction amount
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123

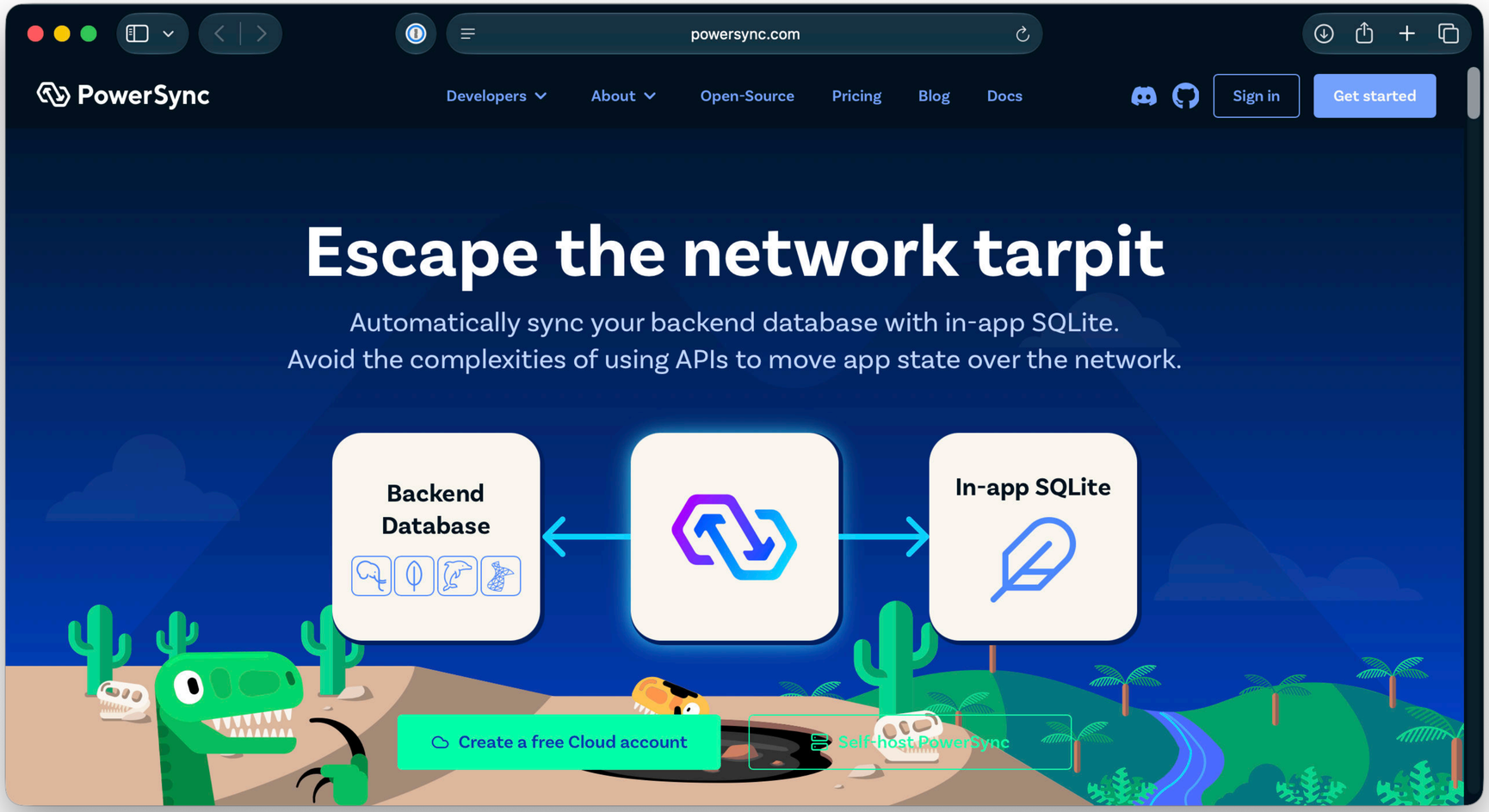
What is local-first?

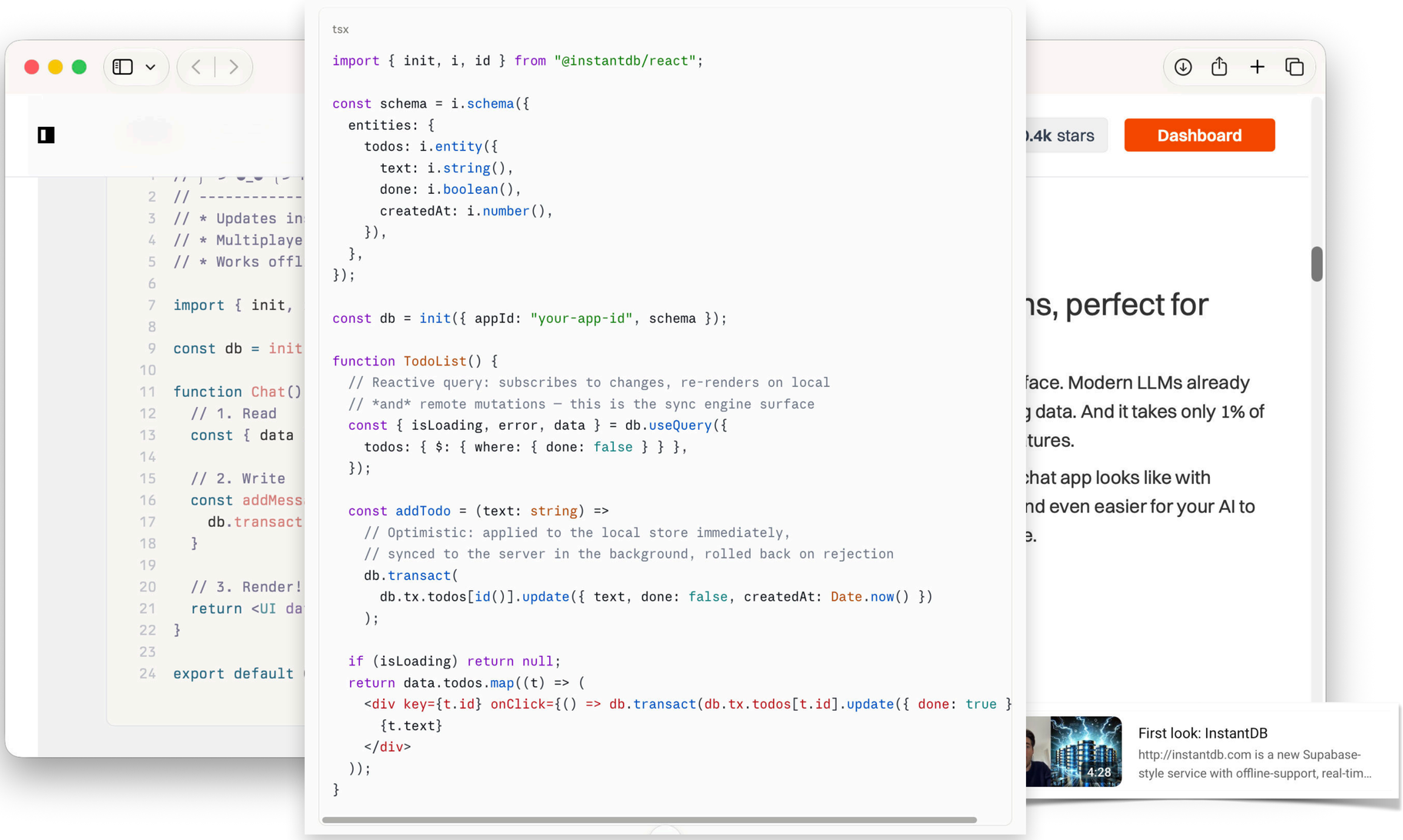
- You own your data, in spite of the cloud
- Seven ideals for local-first software:
    - 1. No spinners: your work at your fingertips
    - 2. Your work is not trapped on one device
    - 3. The network is optional
    - 4. Seamless collaboration with your colleagues
    - 5. The Long Now
    - 6. Security and privacy by default
    - 7. You retain ultimate ownership and control
- https://www.inkandswitch.com/essay/local-first/
- https://lofi.so/

Sync Engines

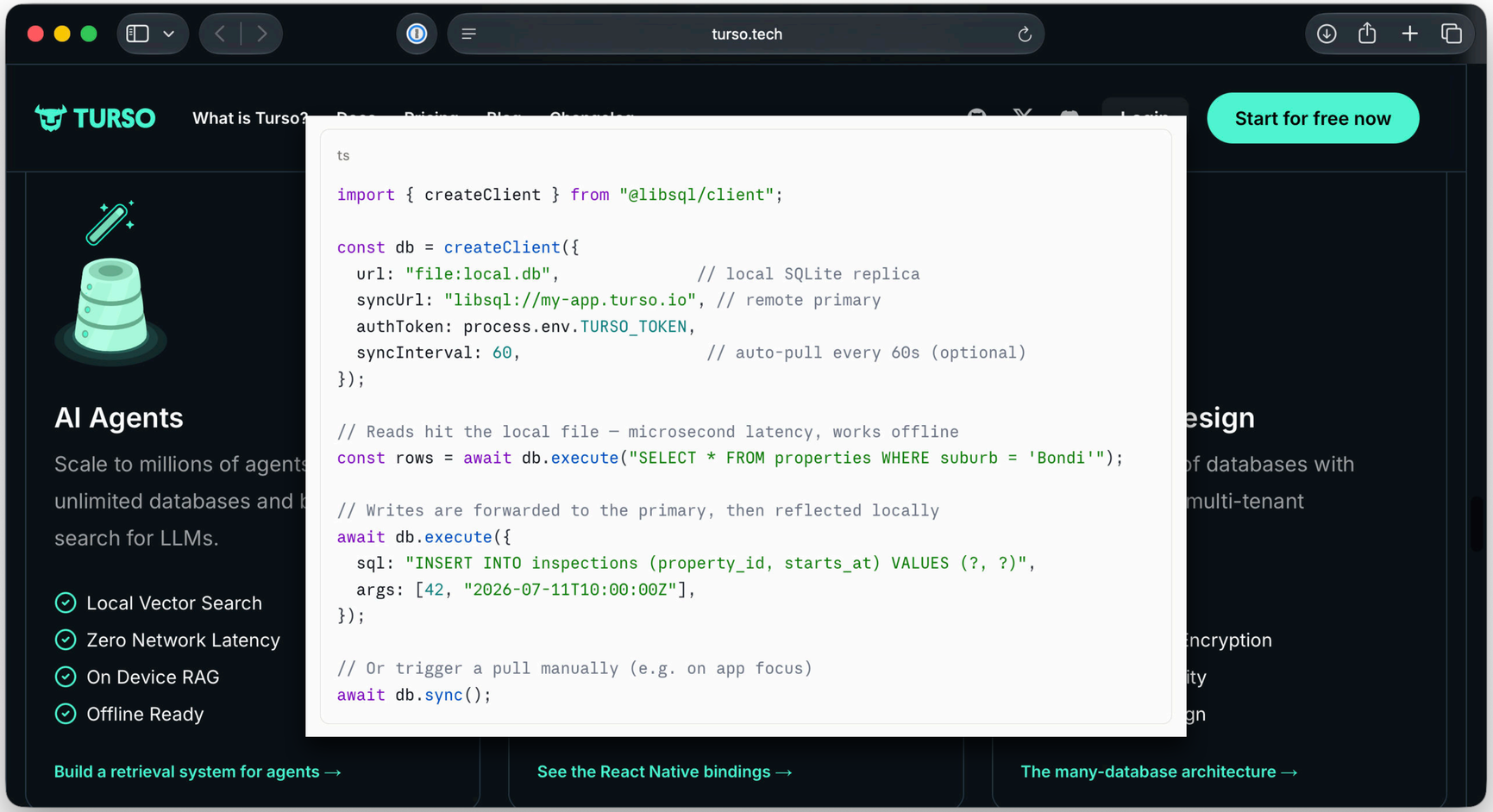


Sign in and then you should see the following (refresh if tasks aren't showing):

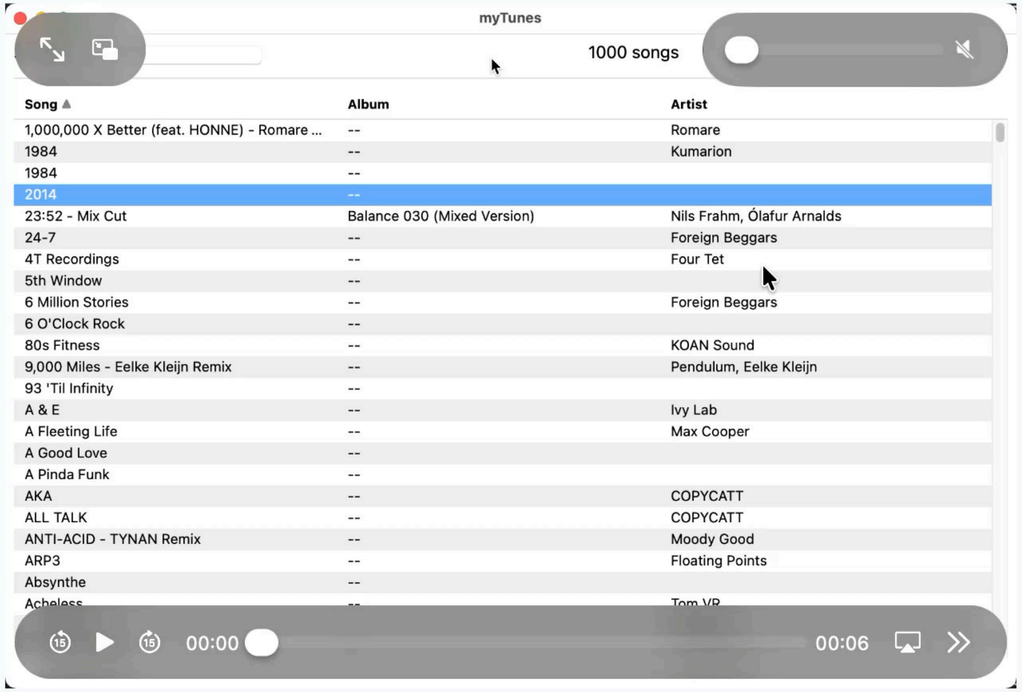

powersync.com
PowerSync
Developers
About
Open-Source
Pricing
Blog
Docs
Sign in
Get started
Escape the network tarpit
Automatically sync your backend database with in-app SQLite.
Avoid the complexities of using APIs to move app state over the network.
Backend Database
In-app SQLite
Create a free Cloud account
Self-host PowerSync



Local data stored in IndexedDB.


turso.tech
TURSO
What is Turso?
Start for free now
ts
import { createClient } from "@libsql/client";
const db = createClient({
url: "file:local.db", // local SQLite replica
syncUrl: "libsql://my-app.turso.io", // remote primary
authToken: process.env.TURSO_TOKEN,
syncInterval: 60, // auto-pull every 60s (optional)
});
// Reads hit the local file — microsecond latency, works offline
const rows = await db.execute("SELECT * FROM properties WHERE suburb = 'Bondi'");
// Writes are forwarded to the primary, then reflected locally
await db.execute({
sql: "INSERT INTO inspections (property_id, starts_at) VALUES (?, ?)",
args: [42, "2026-07-11T10:00:00Z"],
});
// Or trigger a pull manually (e.g. on app focus)
await db.sync();
AI Agents
Scale to millions of agents
unlimited databases and b
search for LLMs.
Local Vector Search
Zero Network Latency
On Device RAG
Offline Ready
Build a retrieval system for agents →
See the React Native bindings →
esign
of databases with
multi-tenant
ncryption
ity
gn
The many-database architecture →

In our experience so far, SQLite can run most queries fast enough to make this approach work. (We discuss what can be done in cases where it isn't fast enough later.) As another example of how fast a local datastore can be, we can store the currently selected track in the database. Selecting tracks with the mouse or keyboard feels responsive, even though it's round-tripping through the database every time the selection changes:



Virtualized list rendering

Personal music collections can get large—it's not uncommon for one person to collect hundreds of thousands of songs over time. With a large collection, it's too slow to

What if users could use the SQLite database *outside* of the app?

developer.mozilla.org

Improve collaboration and visibility with one DevSecOps platform. Learn more from our team. Talk to Our Team

AD

Filter

setInterval()
setResizable()
setTimeout()
showDirectoryPicker()
showOpenFilePicker()
showSaveFilePicker()
sizeToContent()
stop()
structuredClone()
webkitConvertPointFromNodeToPage()

Window: showDirectoryPicker() method

Limited availability

Secure context: This feature is available only in secure contexts (HTTPS), in some or all supporting browsers.

Experimental: This is an experimental technology
Check the Browser compatibility table carefully before using this in production.

The **`showDirectoryPicker()`** method of the `Window` interface displays a directory picker which allows the user to select a directory.

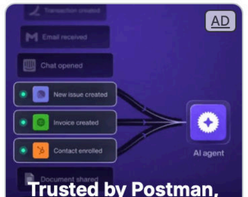
Syntax

JS Copy

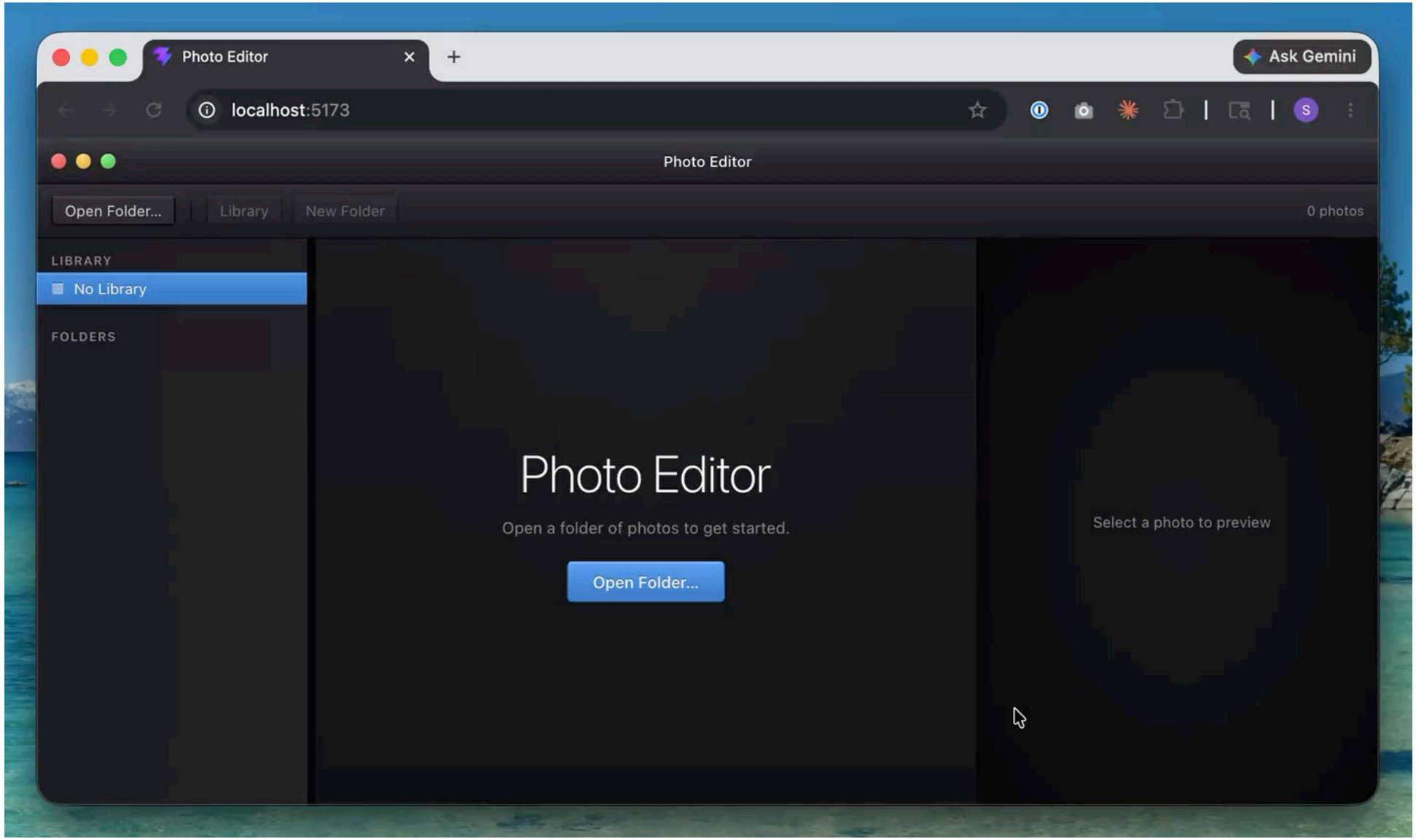
In this article

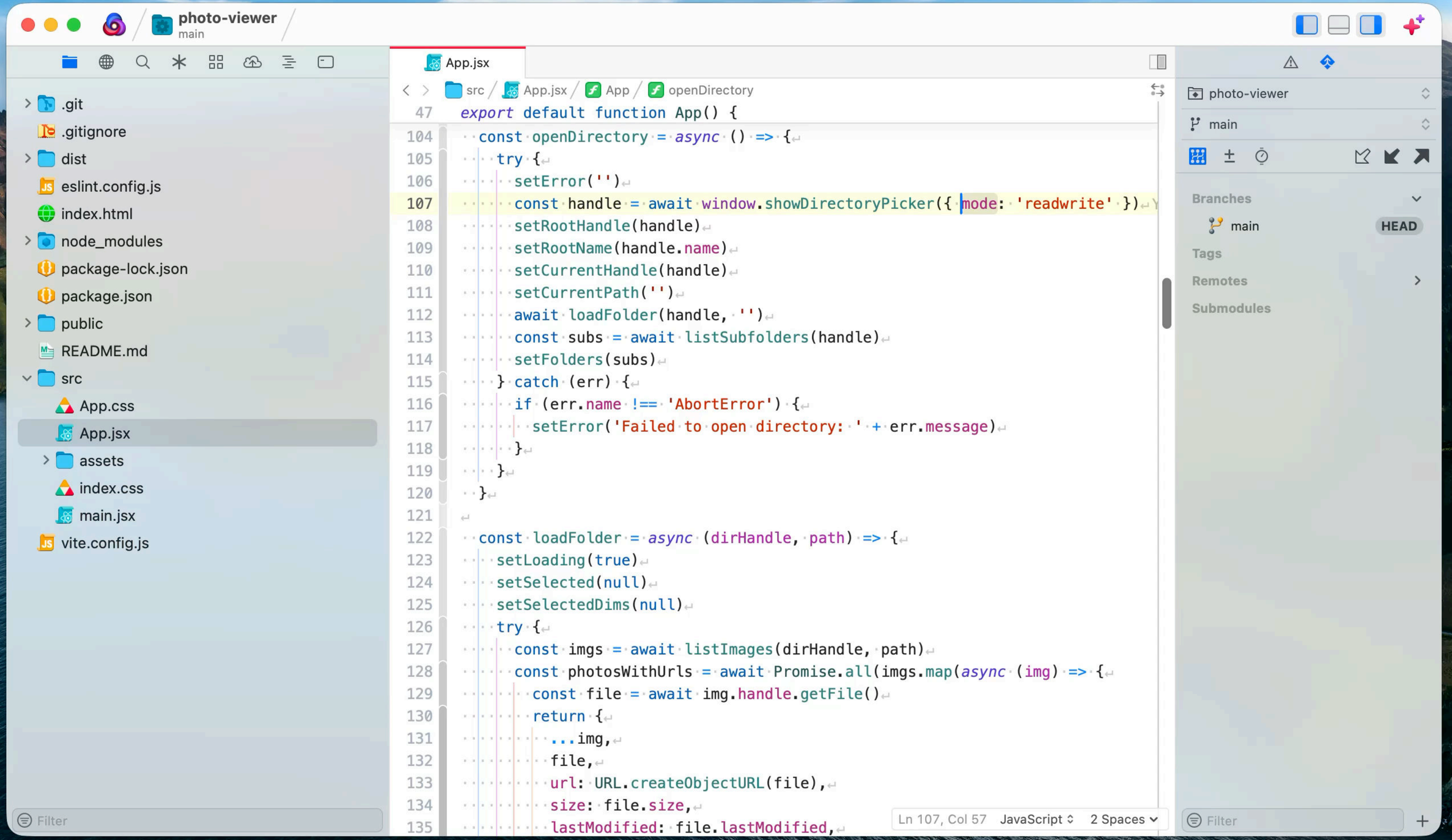
Syntax
Security
Examples
Specifications
Browser compatibility
See also

AD



Trusted by Postman, Copy.ai, and Zendesk.
The most integration-heavy teams chose Paragon. See why Copy.ai didn't build this in-house.


Photo Editor
Ask Gemini
localhost:5173
Photo Editor
Open Folder...
Library
New Folder
0 photos
LIBRARY
No Library
FOLDERS
Photo Editor
Open a folder of photos to get started.
Open Folder...
Select a photo to preview


photo-viewer
main
App.jsx
src / App.jsx / App / openDirectory
export default function App() {
const openDirectory = async () => {
try {
setError('')
const handle = await window.showDirectoryPicker({ mode: 'readwrite' })
setRootHandle(handle)
setRootName(handle.name)
setCurrentHandle(handle)
setCurrentPath('')
await loadFolder(handle, '')
const subs = await listSubfolders(handle)
setFolders(subs)
} catch (err) {
if (err.name !== 'AbortError') {
setError('Failed to open directory: ' + err.message)
}
}
}
const loadFolder = async (dirHandle, path) => {
setLoading(true)
setSelected(null)
setSelectedDims(null)
try {
const imgs = await listImages(dirHandle, path)
const photosWithUrls = await Promise.all(imgs.map(async (img) => {
const file = await img.handle.getFile()
return {
...img,
file,
url: URL.createObjectURL(file),
size: file.size,
lastModified: file.lastModified,
.git
.gitignore
dist
eslint.config.js
index.html
node_modules
package-lock.json
package.json
public
README.md
src
App.css
App.jsx
assets
index.css
main.jsx
vite.config.js
Filter
photo-viewer
main
Branches
main
HEAD
Tags
Remotes
Submodules
Filter
Ln 107, Col 57
JavaScript
2 Spaces

▲ znpy 14 days ago | prev | next [–]

> Chrome introduced a new API, window.showDirectoryPicker() that allows the user to grant access to a directory on their computer and allow a website to read/write everything inside.

I mean, what could go wrong?

It's not like an user is tricked into uploading a file from a folder (let's say, the main "Documents" folder) and some malicious website steals all the files over there.

▲ bigrocketapps 14 days ago | parent | next [–]

If you tried this out in Linux, for example, the system would block you from selecting folders that contain files that are flagged as dangerous or "system files". I'm assuming this was implemented across all OSes.

▲ blharr 14 days ago | root | parent | next [–]

That does not sound at all reassuring, that the only safeguard is the *system* blocking access and that the API has no safeguard.

Its also easily possible to have sensitive files misplaced, especially for a general non-technical user that would be the one falling for a browser hijacking attack

▲ bigrocketapps 14 days ago | root | parent | next [–]

I have not checked the source code to tell you if the system is the one blocking access or the browser-level API itself. I'm guessing it's the browser. The only reason I mentioned Linux is that's where I tested it and I'm sure there are differences across OSes.

My biggest concern here is the write permission.

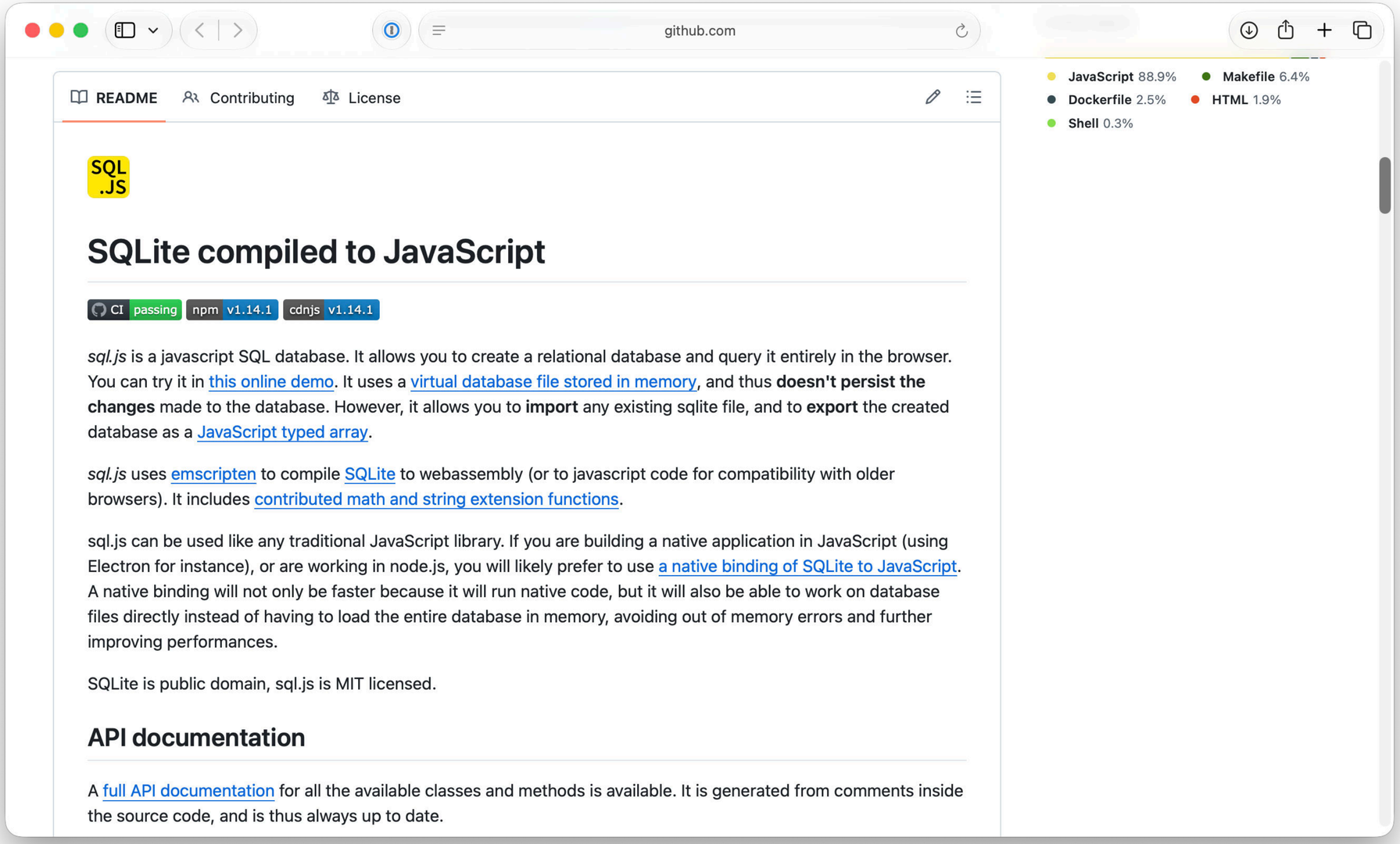
▲ znpy 14 days ago | root | parent | prev | next [–]

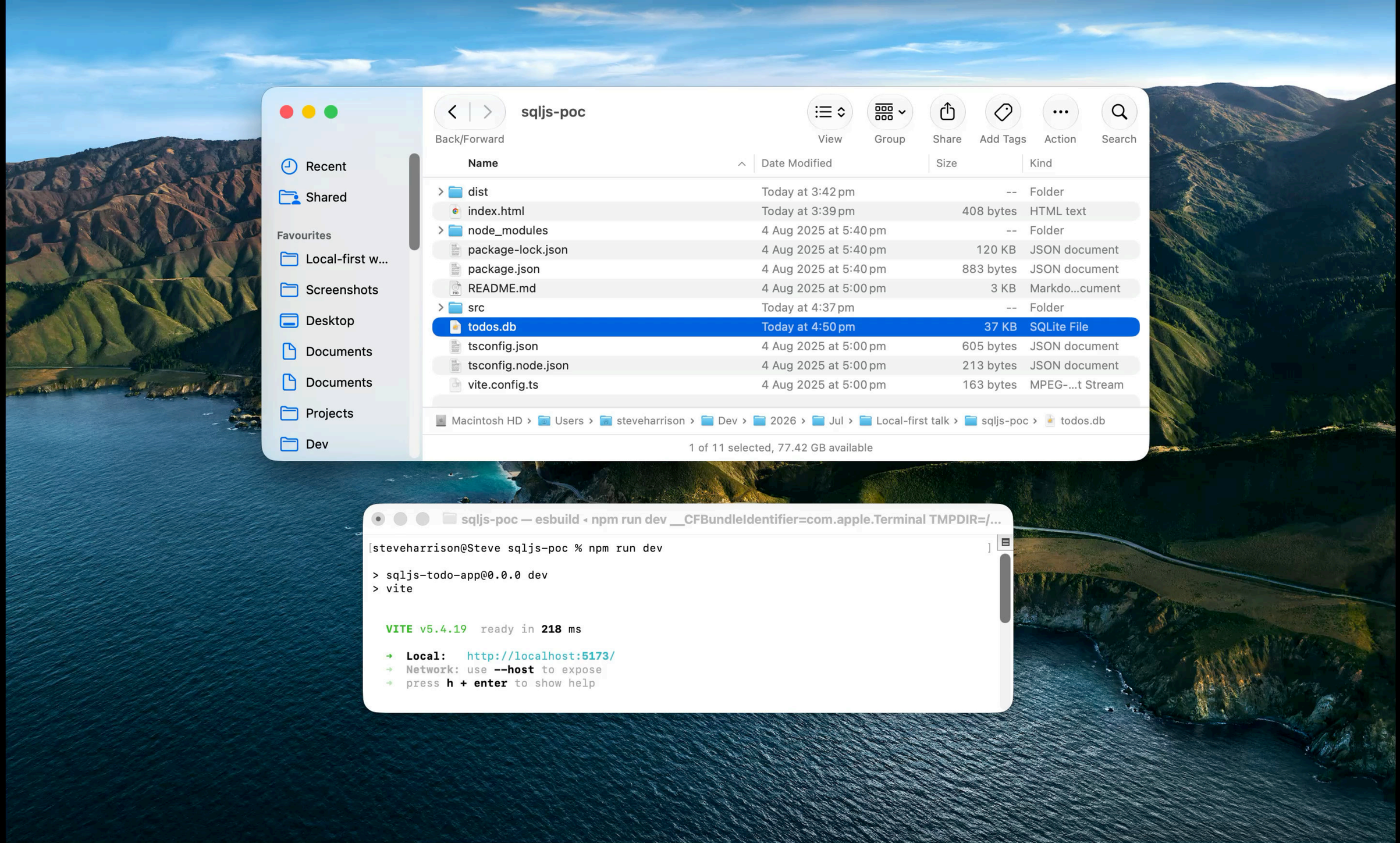
i give zero F about whatever is in my /usr, /var/lib and /opt folders.

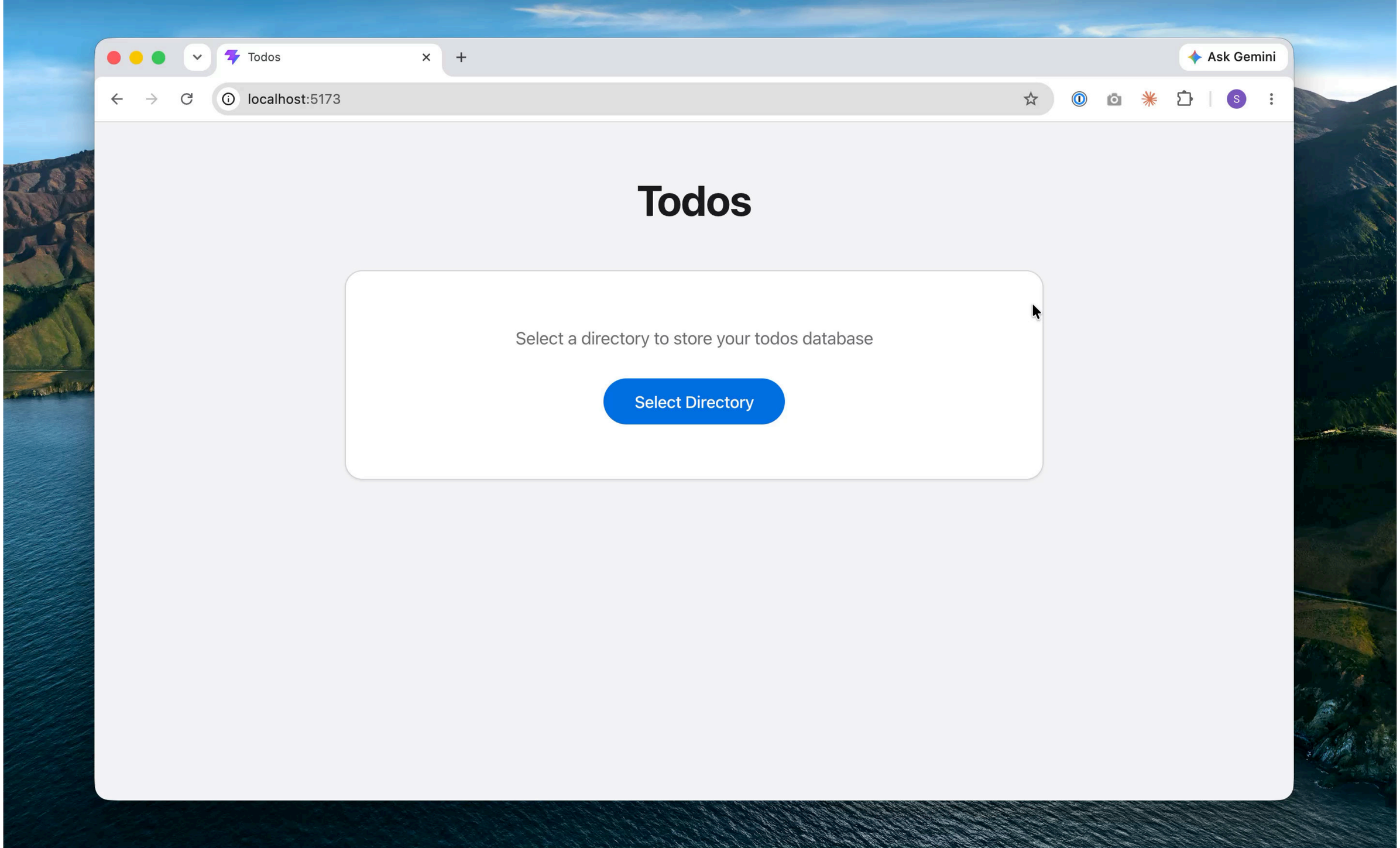
what are websites gonna steal, debian binaries and libraries?

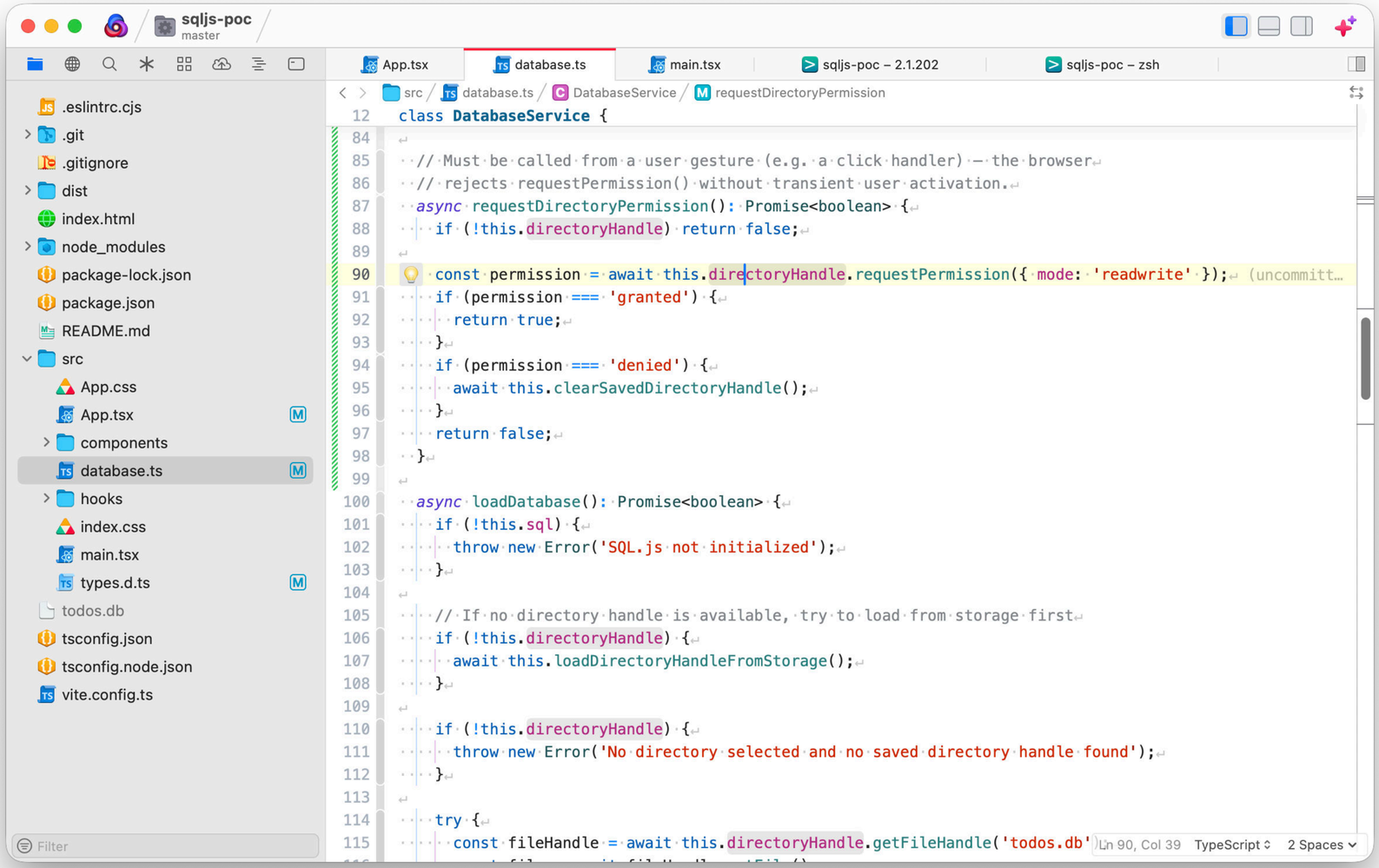
all my important stuff are in my home directory, which is owned (read+write) by me, the same user running the browser.

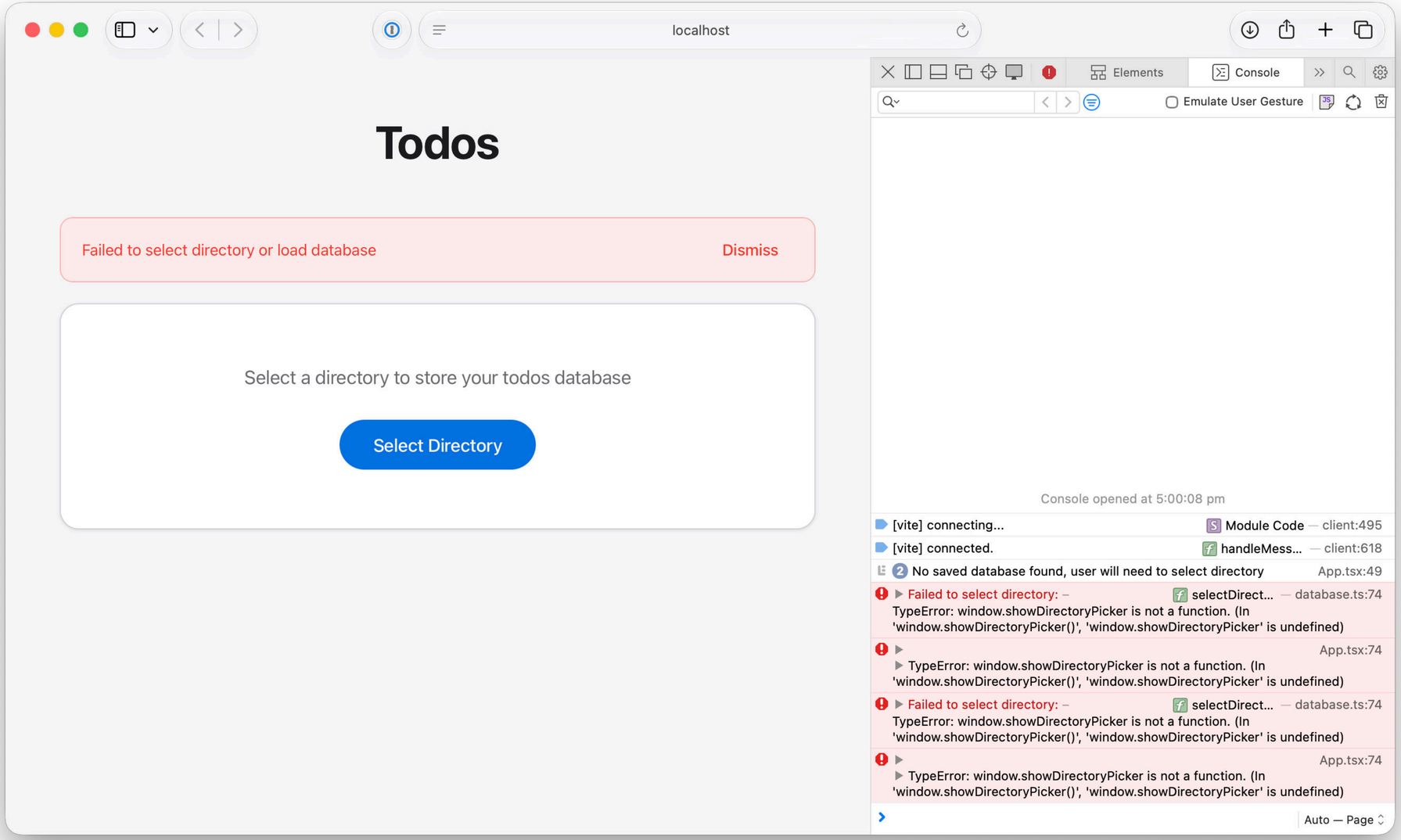
https://news.ycombinator.com/item?id=48629387


github.com
README
Contributing
License
JavaScript 88.9%
Makefile 6.4%
Dockerfile 2.5%
HTML 1.9%
Shell 0.3%
SQL .JS
SQLite compiled to JavaScript
CI passing
npm v1.14.1
cdnjs v1.14.1
sql.js is a javascript SQL database. It allows you to create a relational database and query it entirely in the browser. You can try it in this online demo. It uses a virtual database file stored in memory, and thus doesn't persist the changes made to the database. However, it allows you to import any existing sqlite file, and to export the created database as a JavaScript typed array.
sql.js uses emscripten to compile SQLite to webassembly (or to javascript code for compatibility with older browsers). It includes contributed math and string extension functions.
sql.js can be used like any traditional JavaScript library. If you are building a native application in JavaScript (using Electron for instance), or are working in node.js, you will likely prefer to use a native binding of SQLite to JavaScript. A native binding will not only be faster because it will run native code, but it will also be able to work on database files directly instead of having to load the entire database in memory, avoiding out of memory errors and further improving performances.
SQLite is public domain, sql.js is MIT licensed.
API documentation
A full API documentation for all the available classes and methods is available. It is generated from comments inside the source code, and is thus always up to date.

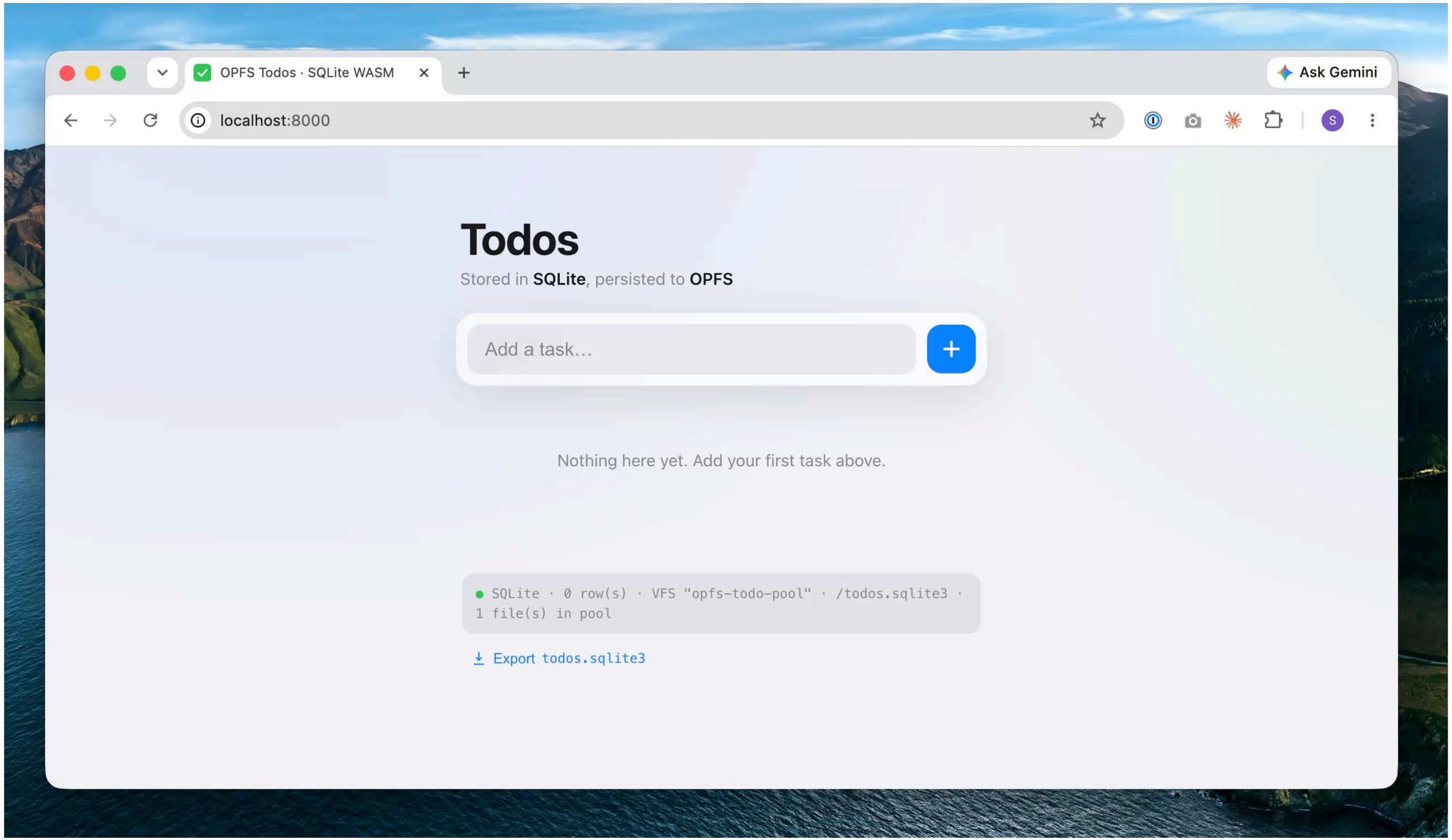

sqljs-poc
Back/Forward
View
Group
Share
Add Tags
Action
Search
Recent
Shared
Favourites
Local-first w...
Screenshots
Desktop
Documents
Documents
Projects
Dev
Name
Date Modified
Size
Kind
dist
Today at 3:42 pm
--
Folder
index.html
Today at 3:39 pm
408 bytes
HTML text
node_modules
4 Aug 2025 at 5:40 pm
--
Folder
package-lock.json
4 Aug 2025 at 5:40 pm
120 KB
JSON document
package.json
4 Aug 2025 at 5:40 pm
883 bytes
JSON document
README.md
4 Aug 2025 at 5:00 pm
3 KB
Markdo...cument
src
Today at 4:37 pm
--
Folder
todos.db
Today at 4:50 pm
37 KB
SQLite File
tsconfig.json
4 Aug 2025 at 5:00 pm
605 bytes
JSON document
tsconfig.node.json
4 Aug 2025 at 5:00 pm
213 bytes
JSON document
vite.config.ts
4 Aug 2025 at 5:00 pm
163 bytes
MPEG-...t Stream
Macintosh HD › Users › steveharrison › Dev › 2026 › Jul › Local-first talk › sqljs-poc › todos.db
1 of 11 selected, 77.42 GB available
sqljs-poc — esbuild ◂ npm run dev __CFBundleIdentifier=com.apple.Terminal TMPDIR=/...
steveharrison@Steve sqljs-poc % npm run dev
> sqljs-todo-app@0.0.0 dev
> vite
VITE v5.4.19 ready in 218 ms
➜ Local: http://localhost:5173/
➜ Network: use --host to expose
➜ press h + enter to show help

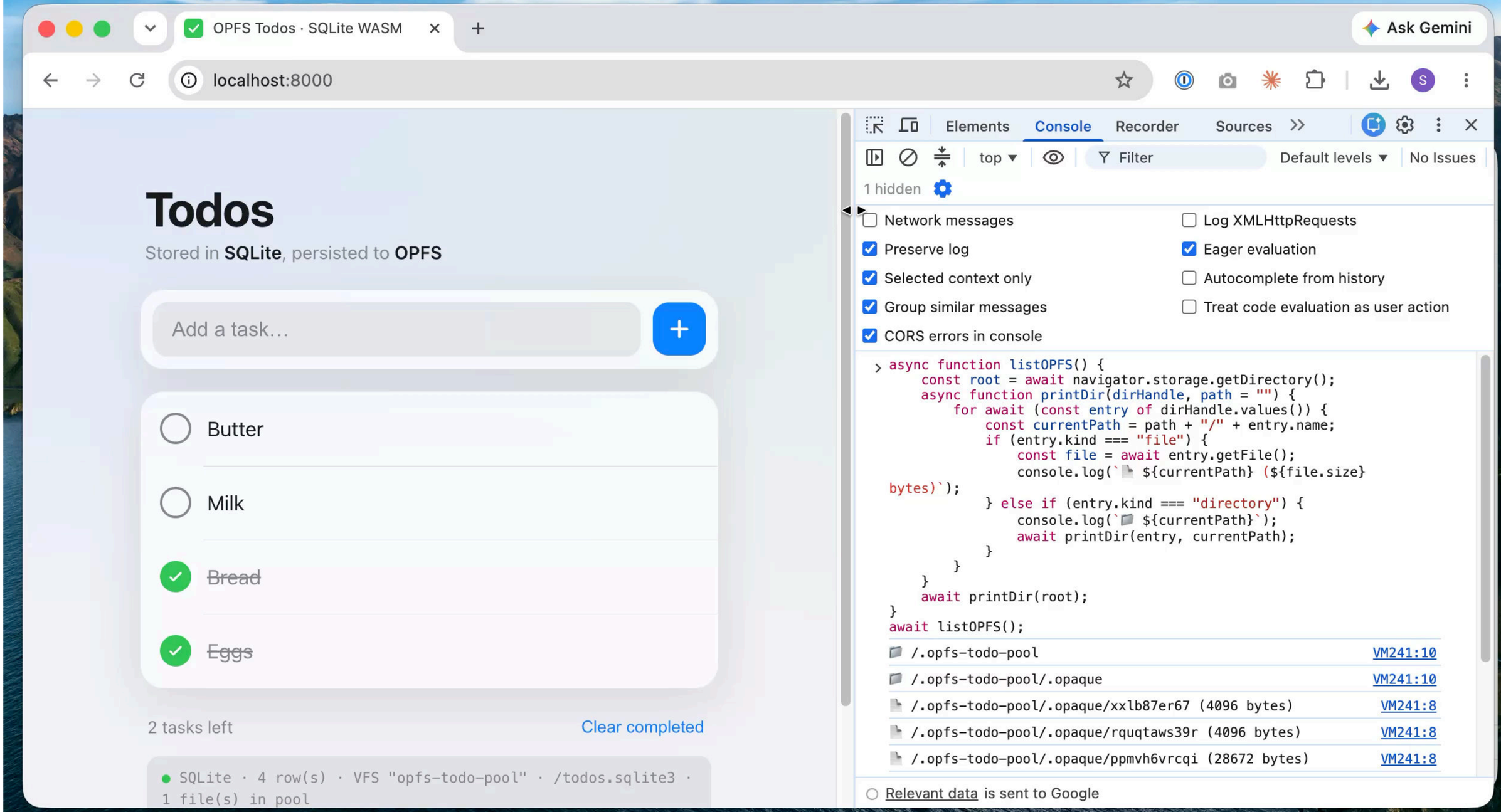

Todos
Ask Gemini
localhost:5173
Todos
Select a directory to store your todos database
Select Directory


sqljs-poc
master
App.tsx
database.ts
main.tsx
sqljs-poc – 2.1.202
sqljs-poc – zsh
src / database.ts / DatabaseService / requestDirectoryPermission
.eslintrc.cjs
.git
.gitignore
dist
index.html
node_modules
package-lock.json
package.json
README.md
src
App.css
App.tsx
components
database.ts
hooks
index.css
main.tsx
types.d.ts
todos.db
tsconfig.json
tsconfig.node.json
vite.config.ts
Filter
class DatabaseService {
// Must be called from a user gesture (e.g. a click handler) — the browser
// rejects requestPermission() without transient user activation.
async requestDirectoryPermission(): Promise<boolean> {
if (!this.directoryHandle) return false;
const permission = await this.directoryHandle.requestPermission({ mode: 'readwrite' });
if (permission === 'granted') {
return true;
}
if (permission === 'denied') {
await this.clearSavedDirectoryHandle();
}
return false;
}
async loadDatabase(): Promise<boolean> {
if (!this.sql) {
throw new Error('SQL.js not initialized');
}
// If no directory handle is available, try to load from storage first
if (!this.directoryHandle) {
await this.loadDirectoryHandleFromStorage();
}
if (!this.directoryHandle) {
throw new Error('No directory selected and no saved directory handle found');
}
try {
const fileHandle = await this.directoryHandle.getFileHandle('todos.db')
Ln 90, Col 39
TypeScript
2 Spaces


localhost
Todos
Failed to select directory or load database
Dismiss
Select a directory to store your todos database
Select Directory
Elements
Console
Emulate User Gesture
Console opened at 5:00:08 pm
[vite] connecting...
Module Code — client:495
[vite] connected.
handleMess... — client:618
No saved database found, user will need to select directory
App.tsx:49
Failed to select directory: –
selectDirect... — database.ts:74
TypeError: window.showDirectoryPicker is not a function. (In 'window.showDirectoryPicker()', 'window.showDirectoryPicker' is undefined)
App.tsx:74
TypeError: window.showDirectoryPicker is not a function. (In 'window.showDirectoryPicker()', 'window.showDirectoryPicker' is undefined)
Failed to select directory: –
selectDirect... — database.ts:74
TypeError: window.showDirectoryPicker is not a function. (In 'window.showDirectoryPicker()', 'window.showDirectoryPicker' is undefined)
App.tsx:74
TypeError: window.showDirectoryPicker is not a function. (In 'window.showDirectoryPicker()', 'window.showDirectoryPicker' is undefined)
Auto — Page

Using OPFS


OPFS Todos · SQLite WASM
Ask Gemini
localhost:8000
Todos
Stored in SQLite, persisted to OPFS
Add a task…
Nothing here yet. Add your first task above.
SQLite · 0 row(s) · VFS "opfs-todo-pool" · /todos.sqlite3 · 1 file(s) in pool
Export todos.sqlite3

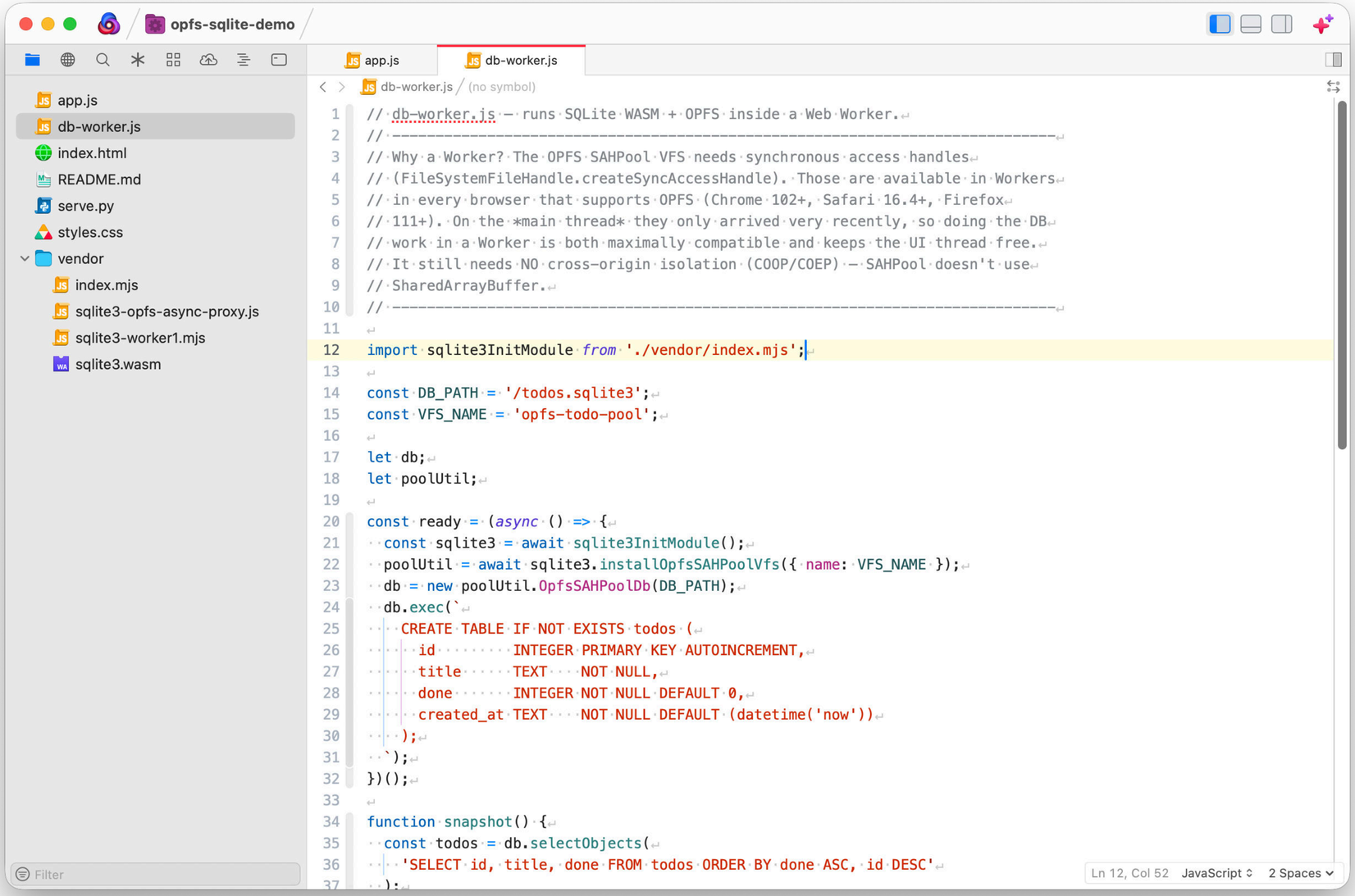

OPFS Todos · SQLite WASM
Ask Gemini
localhost:8000
Todos
Stored in SQLite, persisted to OPFS
Add a task…
Butter
Milk
Bread
Eggs
2 tasks left
Clear completed
SQLite · 4 row(s) · VFS "opfs-todo-pool" · /todos.sqlite3 · 1 file(s) in pool
Elements
Console
Recorder
Sources
top
Filter
Default levels
No Issues
1 hidden
Network messages
Log XMLHttpRequests
Preserve log
Eager evaluation
Selected context only
Autocomplete from history
Group similar messages
Treat code evaluation as user action
CORS errors in console
async function listOPFS() {
const root = await navigator.storage.getDirectory();
async function printDir(dirHandle, path = "") {
for await (const entry of dirHandle.values()) {
const currentPath = path + "/" + entry.name;
if (entry.kind === "file") {
const file = await entry.getFile();
console.log(`📄 ${currentPath} (${file.size} bytes)`);
} else if (entry.kind === "directory") {
console.log(`📁 ${currentPath}`);
await printDir(entry, currentPath);
}
}
}
await printDir(root);
}
await listOPFS();
/.opfs-todo-pool VM241:10
/.opfs-todo-pool/.opaque VM241:10
/.opfs-todo-pool/.opaque/xxlb87er67 (4096 bytes) VM241:8
/.opfs-todo-pool/.opaque/rquqtaws39r (4096 bytes) VM241:8
/.opfs-todo-pool/.opaque/ppmvh6vrcqi (28672 bytes) VM241:8
Relevant data is sent to Google

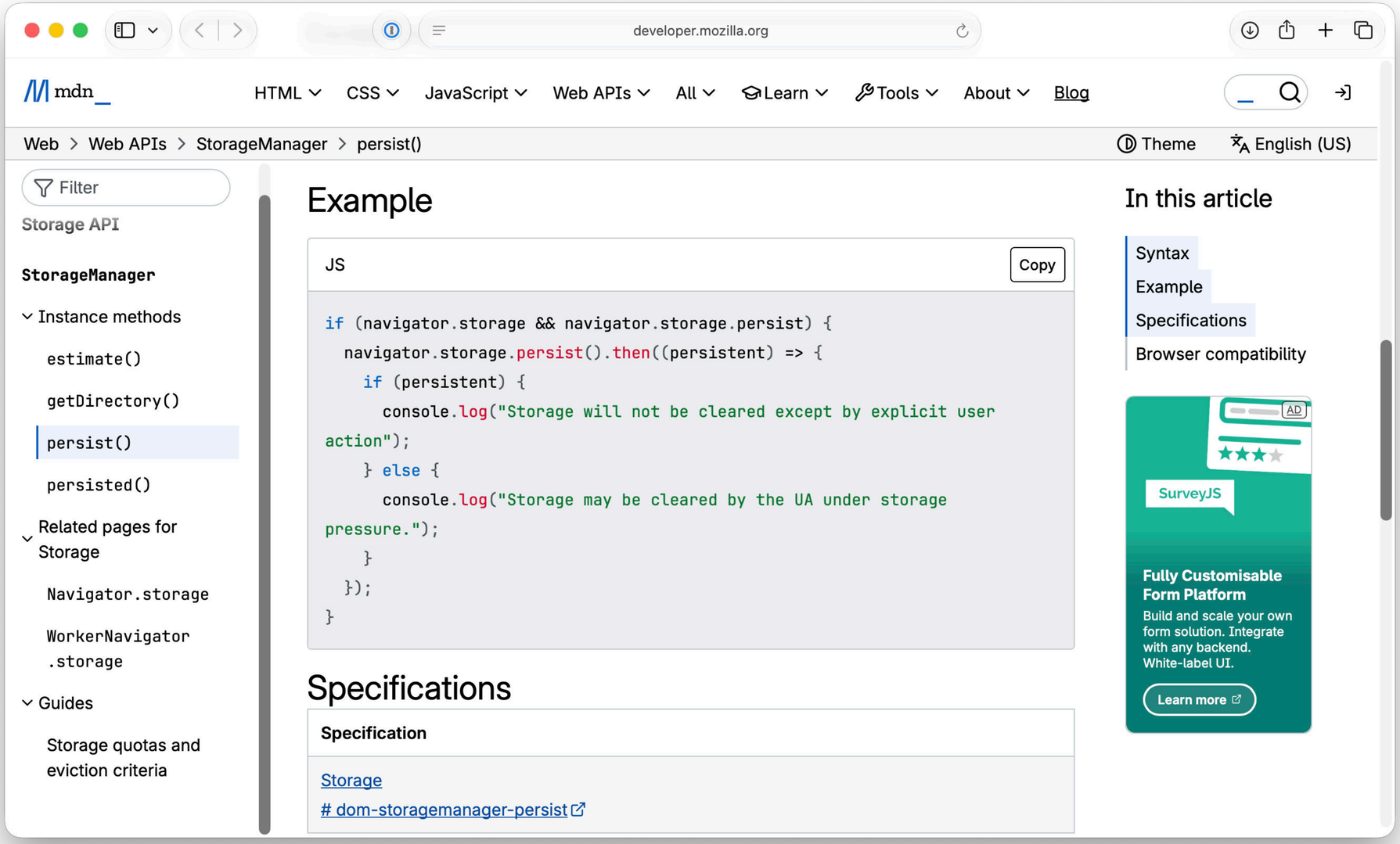
https://web.dev/articles/origin-private-file-system

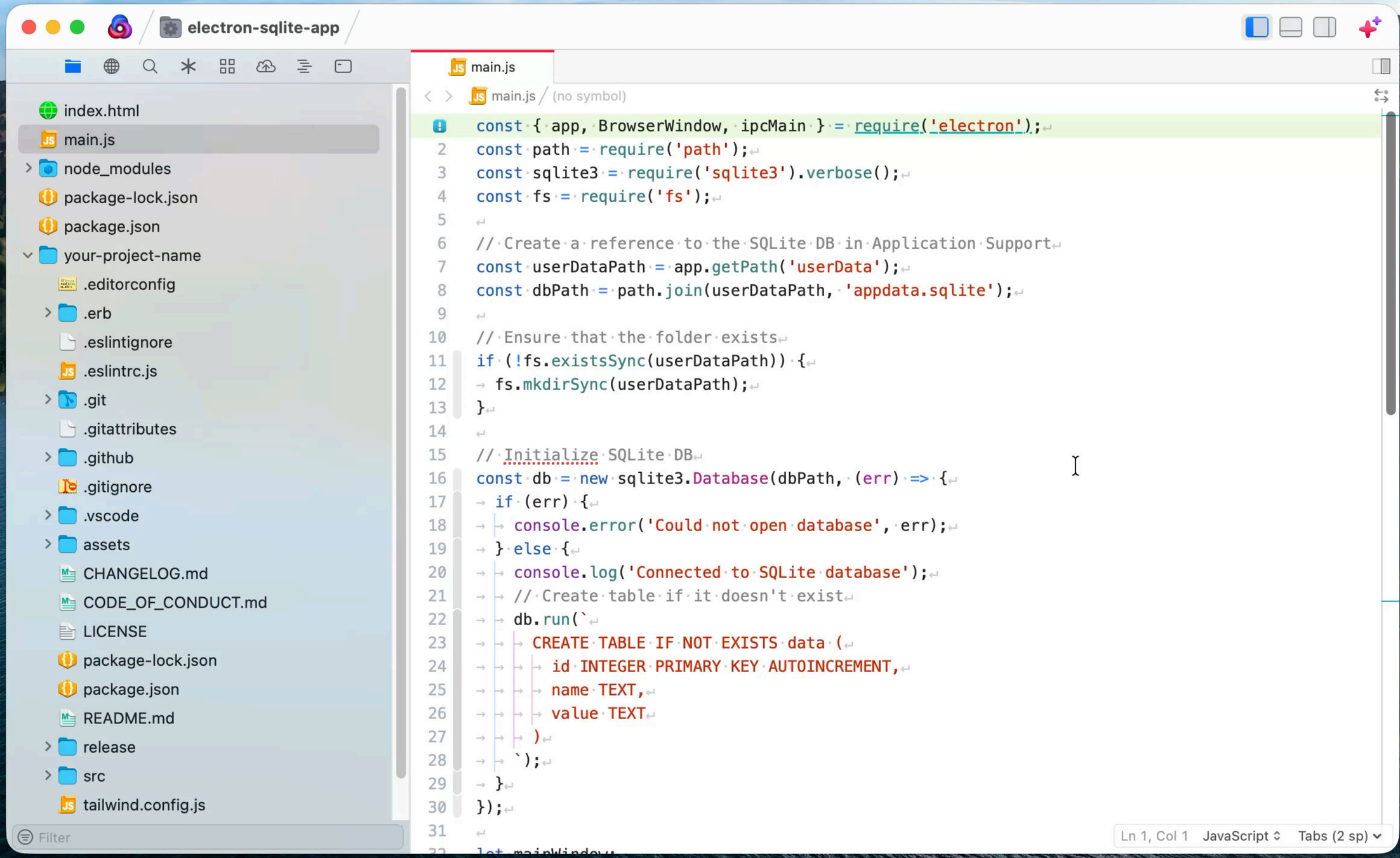
Copy a file from the origin private file system to the user-visible file system

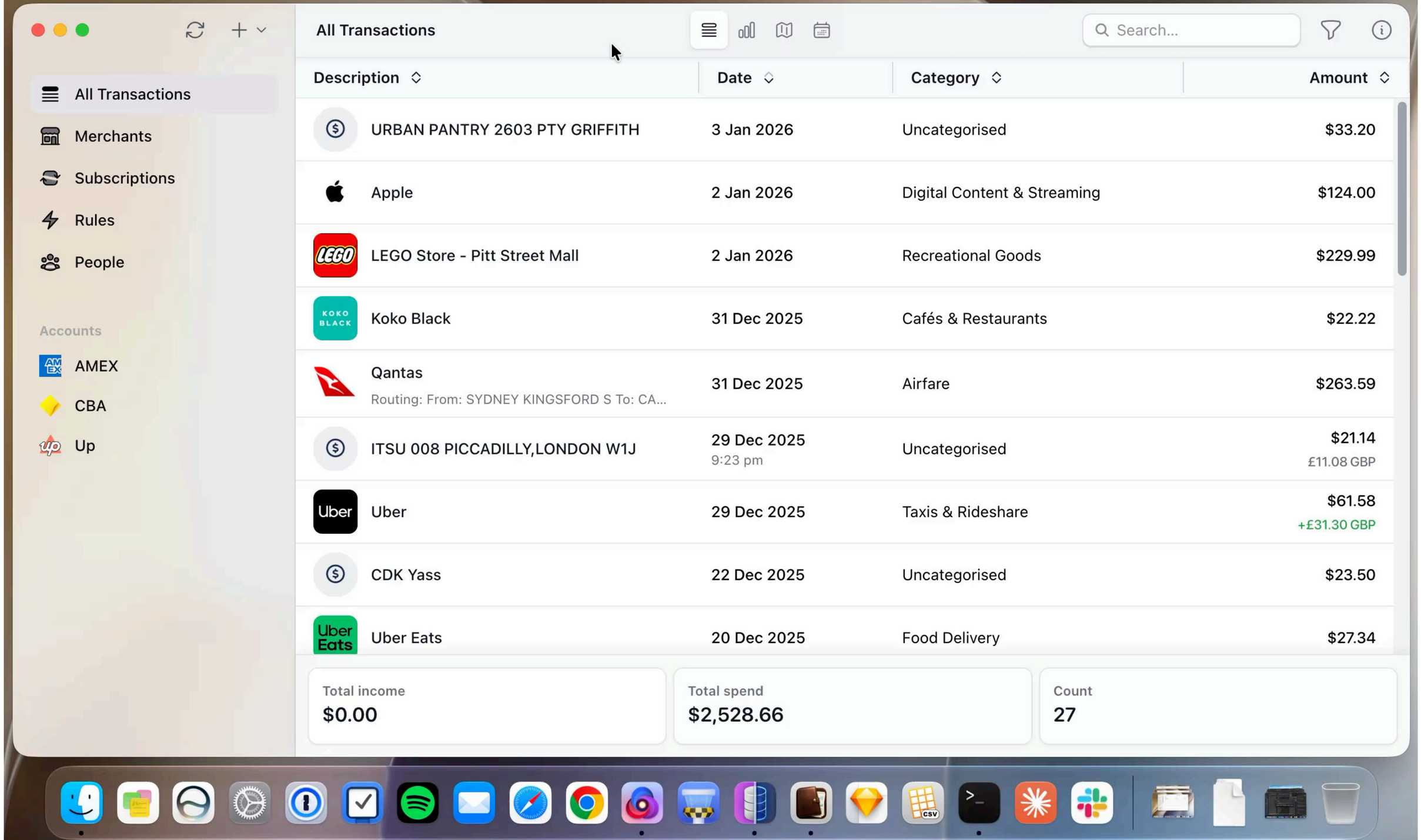
As mentioned above, moving files from the origin private file system to the user-visible file system isn't possible, but you can copy files. Since `showSaveFilePicker()` is only exposed on the main thread, but not in the Worker thread, be sure to run the code there.

```
// On the main thread, not in the Worker. This assumes
// `fileHandle` is the `FileSystemFileHandle` you obtained
// the `FileSystemSyncAccessHandle` from in the Worker
// thread. Be sure to close the file in the Worker thread first.
const fileHandle = await opfsRoot.getFileHandle('fast');
try {
  // Obtain a file handle to a new file in the user-visible file system
  // with the same name as the file in the origin private file system.
  const saveHandle = await showSaveFilePicker({
    suggestedName: fileHandle.name || ''
  });
  const writable = await saveHandle.createWritable();
  await writable.write(await fileHandle.getFile());
  await writable.close();
} catch (err) {
  console.error(err.name, err.message);
}
```


opfs-sqlite-demo
app.js
db-worker.js
index.html
README.md
serve.py
styles.css
vendor
index.mjs
sqlite3-opfs-async-proxy.js
sqlite3-worker1.mjs
sqlite3.wasm
Filter
app.js
db-worker.js
db-worker.js / (no symbol)
// db-worker.js — runs SQLite WASM + OPFS inside a Web Worker.
// --
// Why a Worker? The OPFS SAHPool VFS needs synchronous access handles
// (FileSystemFileHandle.createSyncAccessHandle). Those are available in Workers
// in every browser that supports OPFS (Chrome 102+, Safari 16.4+, Firefox
// 111+). On the *main thread* they only arrived very recently, so doing the DB
// work in a Worker is both maximally compatible and keeps the UI thread free.
// It still needs NO cross-origin isolation (COOP/COEP) — SAHPool doesn't use
// SharedArrayBuffer.
// --
import sqlite3InitModule from './vendor/index.mjs';
const DB_PATH = '/todos.sqlite3';
const VFS_NAME = 'opfs-todo-pool';
let db;
let poolUtil;
const ready = (async () => {
const sqlite3 = await sqlite3InitModule();
poolUtil = await sqlite3.installOpfsSAHPoolVfs({ name: VFS_NAME });
db = new poolUtil.OpfsSAHPoolDb(DB_PATH);
db.exec(`
CREATE TABLE IF NOT EXISTS todos (
id INTEGER PRIMARY KEY AUTOINCREMENT,
title TEXT NOT NULL,
done INTEGER NOT NULL DEFAULT 0,
created_at TEXT NOT NULL DEFAULT (datetime('now'))
);
`);
})();
function snapshot() {
const todos = db.selectObjects(
'SELECT id, title, done FROM todos ORDER BY done ASC, id DESC'
);
Ln 12, Col 52
JavaScript
2 Spaces


developer.mozilla.org
mdn
HTML
CSS
JavaScript
Web APIs
All
Learn
Tools
About
Blog
Web > Web APIs > StorageManager > persist()
Theme
English (US)
Filter
Storage API
StorageManager
Instance methods
estimate()
getDirectory()
persist()
persisted()
Related pages for Storage
Navigator.storage
WorkerNavigator.storage
Guides
Storage quotas and eviction criteria
Example
JS
Copy
if (navigator.storage && navigator.storage.persist) {
navigator.storage.persist().then((persistent) => {
if (persistent) {
console.log("Storage will not be cleared except by explicit user action");
} else {
console.log("Storage may be cleared by the UA under storage pressure.");
}
});
}
Specifications
Specification
Storage
dom-storagemanager-persist
In this article
Syntax
Example
Specifications
Browser compatibility
AD
SurveyJS
Fully Customisable Form Platform
Build and scale your own form solution. Integrate with any backend. White-label UI.
Learn more


electron-sqlite-app
index.html
main.js
node_modules
package-lock.json
package.json
your-project-name
.editorconfig
.erb
.eslintignore
.eslintrc.js
.git
.gitattributes
.github
.gitignore
.vscode
assets
CHANGELOG.md
CODE_OF_CONDUCT.md
LICENSE
package-lock.json
package.json
README.md
release
src
tailwind.config.js
Filter
main.js
main.js / (no symbol)
const { app, BrowserWindow, ipcMain } = require('electron');
const path = require('path');
const sqlite3 = require('sqlite3').verbose();
const fs = require('fs');
// Create a reference to the SQLite DB in Application Support
const userDataPath = app.getPath('userData');
const dbPath = path.join(userDataPath, 'appdata.sqlite');
// Ensure that the folder exists
if (!fs.existsSync(userDataPath)) {
fs.mkdirSync(userDataPath);
}
// Initialize SQLite DB
const db = new sqlite3.Database(dbPath, (err) => {
if (err) {
console.error('Could not open database', err);
} else {
console.log('Connected to SQLite database');
// Create table if it doesn't exist
db.run(`
CREATE TABLE IF NOT EXISTS data (
id INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT,
value TEXT
)
`);
}
});
Ln 1, Col 1
JavaScript
Tabs (2 sp)


All Transactions
Search...
All Transactions
Merchants
Subscriptions
Rules
People
Accounts
AMEX
CBA
Up
Description
Date
Category
Amount
URBAN PANTRY 2603 PTY GRIFFITH
3 Jan 2026
Uncategorised
$33.20
Apple
2 Jan 2026
Digital Content & Streaming
$124.00
LEGO Store - Pitt Street Mall
2 Jan 2026
Recreational Goods
$229.99
Koko Black
31 Dec 2025
Cafés & Restaurants
$22.22
Qantas
Routing: From: SYDNEY KINGSFORD S To: CA...
31 Dec 2025
Airfare
$263.59
ITSU 008 PICCADILLY,LONDON W1J
29 Dec 2025
9:23 pm
Uncategorised
$21.14
£11.08 GBP
Uber
29 Dec 2025
Taxis & Rideshare
$61.58
+£31.30 GBP
CDK Yass
22 Dec 2025
Uncategorised
$23.50
Uber Eats
20 Dec 2025
Food Delivery
$27.34
Total income
$0.00
Total spend
$2,528.66
Count
27

My Finances — ✳ Calculate total Uber spending — claude — 80×28

Here's your total Uber spending:

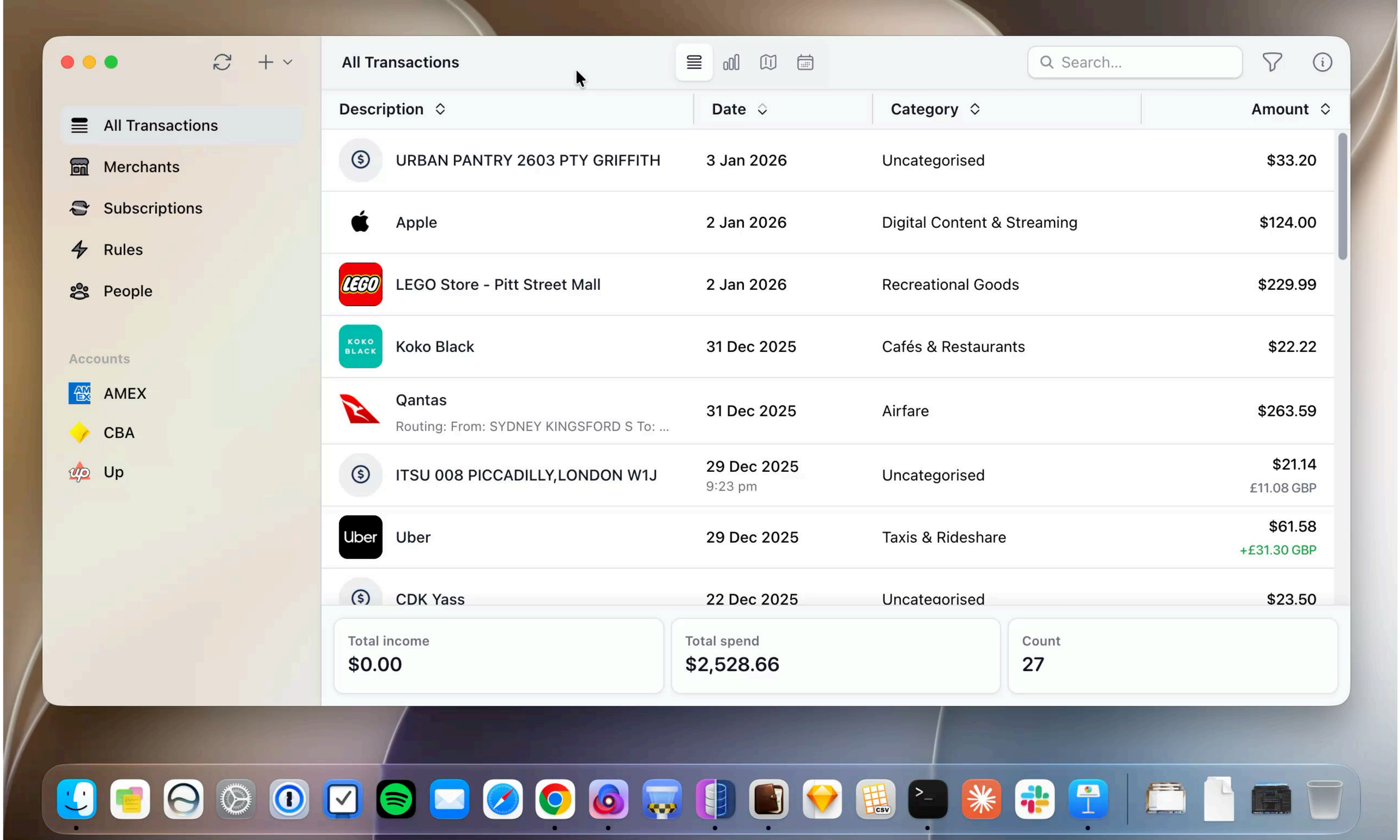
Uber spending: $198.63 across 5 transactions

Category	Transactions	Total
🚗 Uber (rides)	2	$93.25
🍔 Uber Eats	3	$105.38
Total	**5**	**$198.63**

All transactions:
- 2025-09-20 — Uber Eats — $32.75
- 2025-12-17 — Uber Trip — $31.67
- 2025-12-19 — Uber Eats — $45.29
- 2025-12-20 — Uber Eats — $27.34
- 2025-12-29 — Uber Trip — $61.58

This covers everything in your database matching Uber (checked both merchant tags and raw transaction text), spanning Sep–Dec 2025. Want me to break it down by month, or separate rides vs. food delivery in more detail?

✻ Churned for 3m 47s


All Transactions
Search...
All Transactions
Merchants
Subscriptions
Rules
People
Accounts
AMEX
CBA
Up
Description
Date
Category
Amount
URBAN PANTRY 2603 PTY GRIFFITH
3 Jan 2026
Uncategorised
$33.20
Apple
2 Jan 2026
Digital Content & Streaming
$124.00
LEGO Store - Pitt Street Mall
2 Jan 2026
Recreational Goods
$229.99
Koko Black
31 Dec 2025
Cafés & Restaurants
$22.22
Qantas
Routing: From: SYDNEY KINGSFORD S To: ...
31 Dec 2025
Airfare
$263.59
ITSU 008 PICCADILLY,LONDON W1J
29 Dec 2025
9:23 pm
Uncategorised
$21.14
£11.08 GBP
Uber
29 Dec 2025
Taxis & Rideshare
$61.58
+£31.30 GBP
CDK Yass
22 Dec 2025
Uncategorised
$23.50
Total income
$0.00
Total spend
$2,528.66
Count
27

steveharrison.dev