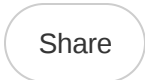
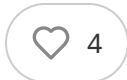


Simple lasts longer: making a map stateful using scotch tape and Web Storage API

🌐 Too many invaders to keep track of; I really don't want to run another database; Web Storage to the rescue



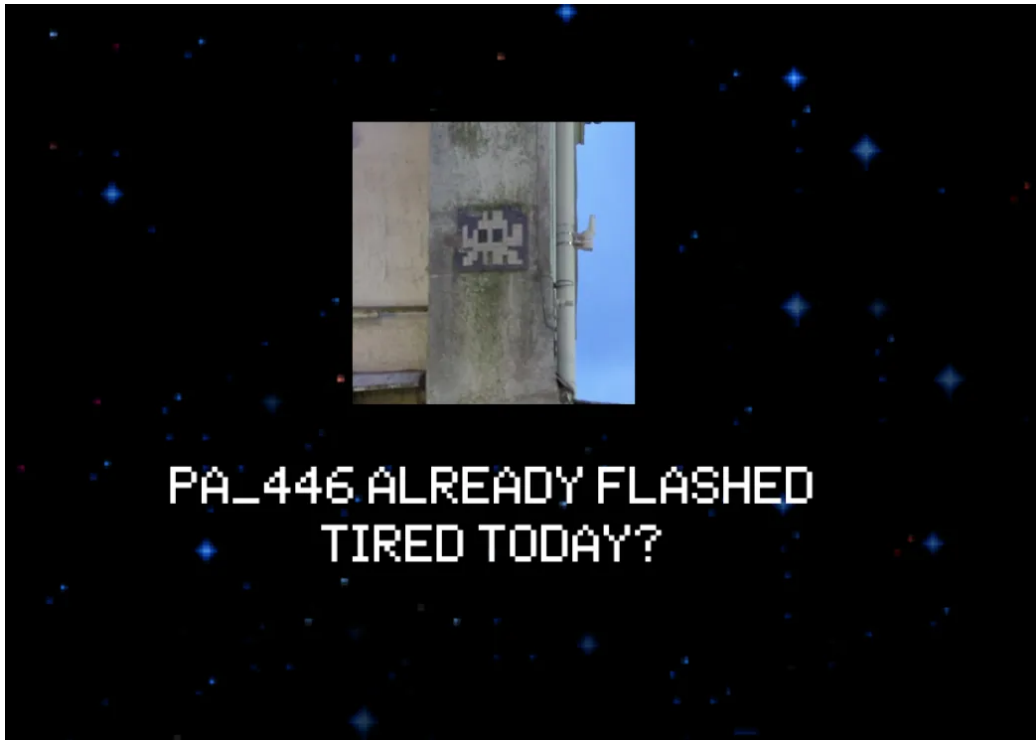
PRZEMEK
JAN 7, 2024



This is less useful than I thought.

I [made a map](#) of the collectible [Space Invader](#) mosaics and [put it online](#). I thought it would be helpful for planning walks (and runs 🚀) with some Space Invader hunting along the way.

But I missed one problem. Can you spot the issue?



My map couldn't track what's already "collected". It was perfectly simple and perfectly "stateless".

Need to put it *somewhere*

Stateless websites are easy to manage. You just need to publish a few static files on the Internet. The website looks the same to every user and it never changes (unless you publish an update).

To keep track of the mosaics flashed by each user, I need to teach the website to store the list of what's already collected ... somewhere. But where?

- The traditional answer is to run a **database** and store the collection of each user there.
- A more trendy, newer option is to use a 3rd party **service** that acts like a database, but runs on someone else's server. [Firebase](#) is a well-known example.

The second approach is less work than the first one (the website can directly talk to Firebase, I don't need to set up any servers myself), but both require some work. Worse

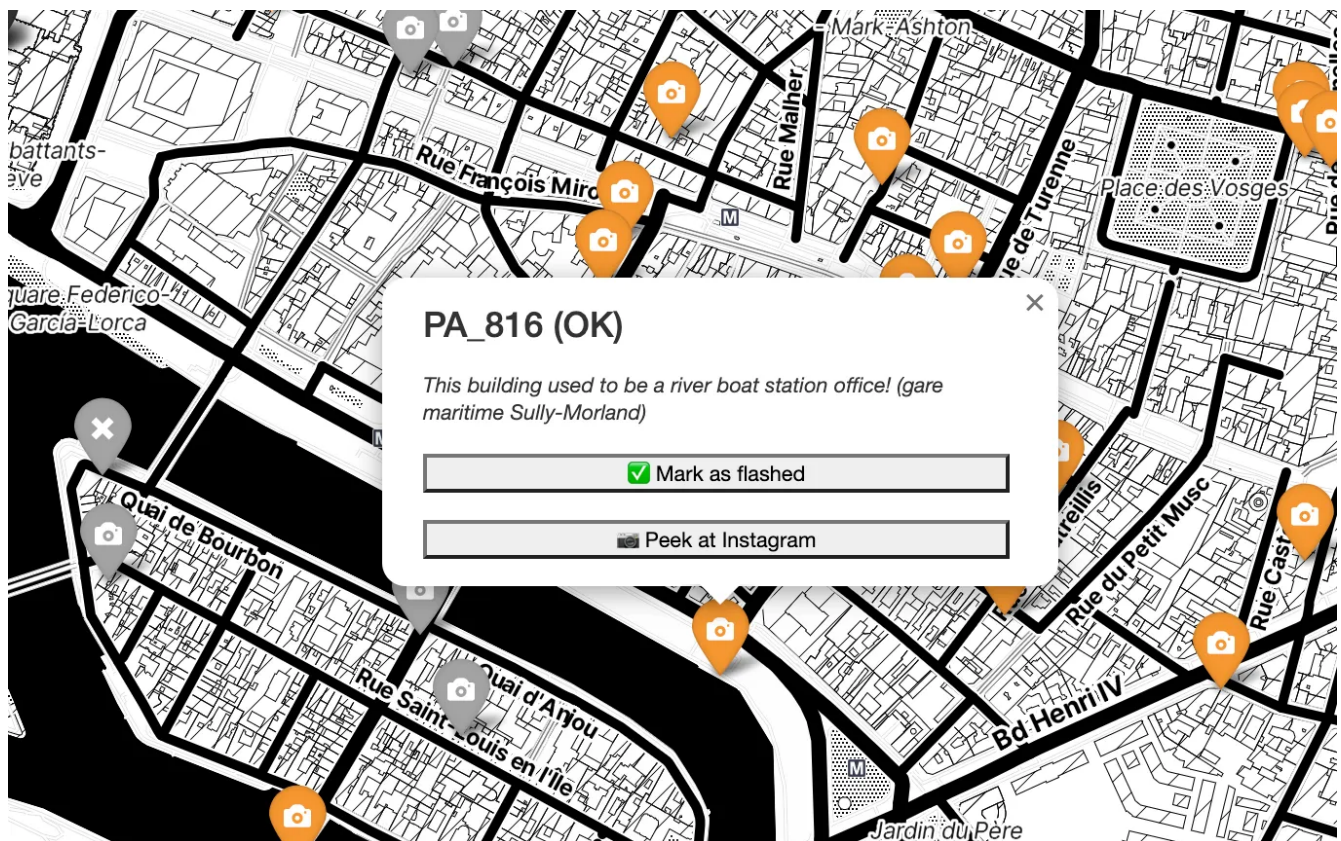
yet, they increase the complexity of the project, adding a dependency on another service.

Complexity kills projects 🗡️ (especially the fragile hobby projects of distracted minds 🌟), so let's do something simpler. Could we store the list of collected invaders directly in the browser of each user?

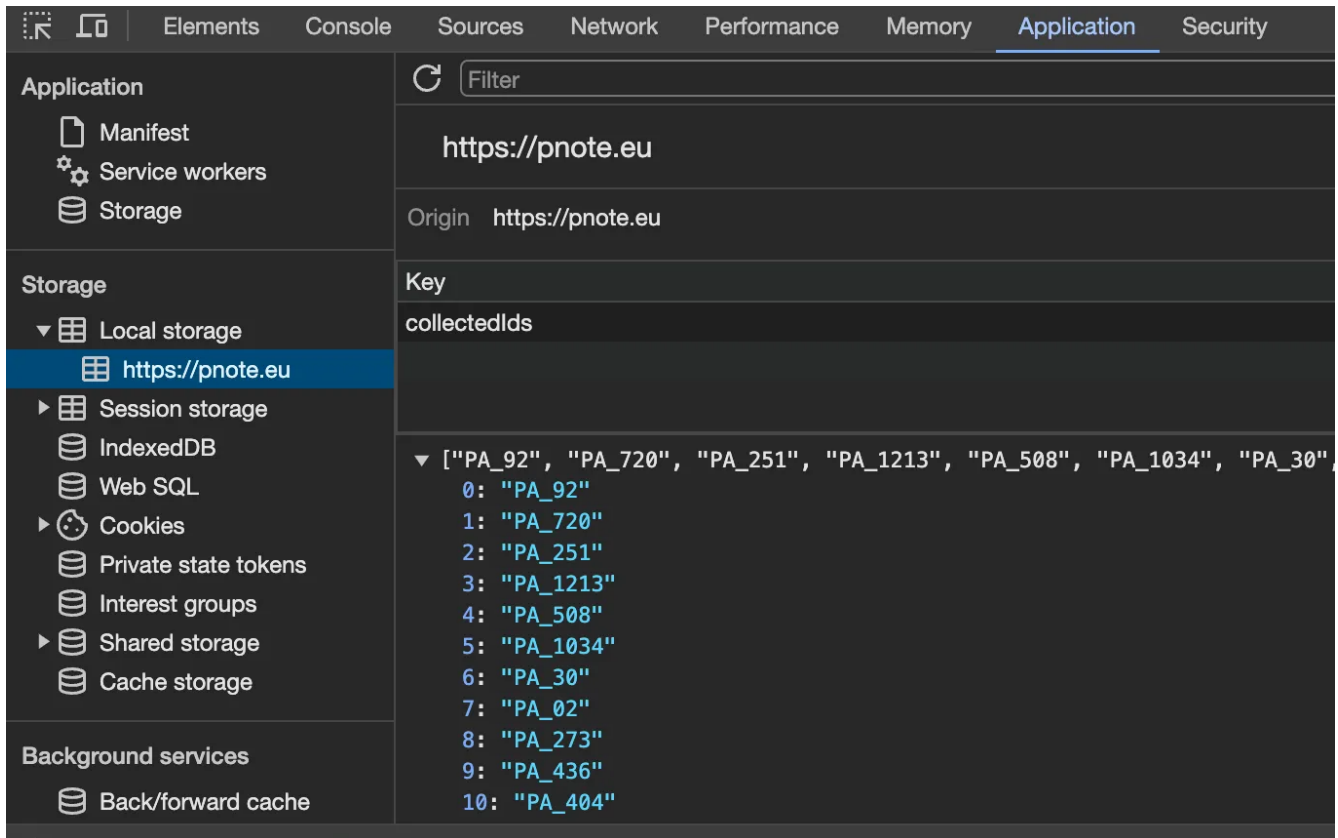
Store your own state

Thanks to the [Web Storage API](#), any website can store a small amount of data on your computer/phone, and it will be persisted through reboots. That's promising!

I modified the map website to include buttons for marking each invader as flashed, and made it persist the state in the local Web Storage.



This works by default, without any permission prompts. Using Chrome developer tools, we can inspect the state the map stores in the browser. It's just the list of invader IDs:



This is very simple and it works! You can see the published updated map [here](#).

No state on the server needed. No user accounts, no authentication, no “I forgot my password” problem. Also, when we don’t store any data on the server, we cannot lose or leak it. ([Local-first software](#).)

The pesky problem of sync and backup

If this solution is so nice, why all the major websites insist on creating an account and storing our state server-side?

For many reasons, one of which is that when the data is stored locally in the browser, it’s not synced across devices and not backed up. If the user re-installs their browser, they lose all state.

Since we’re into simplicity, maybe we can allow users to take care of backup themselves?

It only took 50 lines of code to add a little popup with a feature to import/export the list of collected mosaics. And ChatGPT was happy to practically write all of it:

We ended up with a solution that tries to be the best of all worlds. The website remains simple and it doesn't have to manage user state. But the users can still mark them as collected, and even export and backup their collection if they want to.

Simple lasts longer

Hobby projects are fragile. They can only last if they're simple to maintain, otherwise keeping them working becomes tedious and the author eventually gives up.

I'm still maintaining a database-based hobby project I started 15 years ago. It ended up growing a small community that relies on it. I'm happy about it, but I feel the ongoing maintenance cost of keeping the app running and up to date.




Wiser from this experience, these days I try to build all hobby projects in a way that minimizes dependencies and general fragility. If needed, the app should keep running with zero modifications for the decade to come 🙌.

Simple is not only faster to launch, it also lasts longer.

Question

Are you thinking of starting any hobby project in 2024? How simple can you make it for the initial version?

Cool hobby projects out there

-  [Enso](#), a text editor where you can only write, but not edit. Great for breaking through writer's block: just let the thoughts flow
-  [muted.io](#), a toolkit of music theory tools, running in the browser
-  [Turing Complete](#), a computer science game where you build a computer from increasingly complex components, starting with logic gates

Postcard from Paris

A wave of cold weather is expected in Paris anytime now. Great time for reading and writing with a cup of tea. Or maybe looking up that mulled wine recipe.

Stay warm! 🌟

– Przemek

Comments



Write a comment...

© 2024 Przemek · [Privacy](#) · [Terms](#) · [Collection notice](#)

[Substack](#) is the home for great writing