



Demo apps utilizing Service Workers.

☆ 466 stars🔗 64 forks👁 51 watching🔗 Branches📈 Activity📁 Custom properties📁 Tags











🌐 Public repository

🔗 1 Branch📁 0 Tags🔗📁

🔍 Go to filetGo to file+Add file

<> Code⋮

46M

 Marcos Cáceres Update README.md98ad524 · 8 years ago🕒
 gameUpdate README.md10 years ago
 magazines-appUpdate README.md10 years ago
 mapsUpdate README.md10 years ago
 podcast-appUpdate README.md10 years ago
 todo-appUpdate README.md10 years ago
 twitter-clientAdded note that this is mov...10 years ago
 wikiUpdate README.md10 years ago
 CNAMECreate CNAME10 years ago
 README.mdUpdate README.md8 years ago

📖 README✎☰

THIS REPO IS NO LONGER MAINTAINED

Thanks for stopping by! When we started this work, Service Workers were just becoming "a thing" - so this repo was intended to bootstrap the ecosystem. Since then countless amazing resources and videos on using Service Workers became available. As such, we are no longer able to keep up. Below links are still relevant, but we are not longer adding new ones.

Historical...

We hope to build a set of reference applications that make full use of [service workers](#). However, currently there are no working demos in this repo.

For ServiceWorker demos try out:

Service Worker recipes

[Basic registration](#) - a bare-bones sample that simply performs service worker registration, with placeholders for various event handlers.

[Detailed registration](#) - a sample that provides detailed information about the service worker registration and the state changes that a service worker undergoes.

[Prefetching resources during installation](#) - a sample demonstrating how to prefetch and cache a list of URLs during the service worker's installation, ensuring that they're available offline.

[Selective caching](#) - a sample of how a service worker can cache resources "on the fly", assuming the resources meet certain criteria (MIME type, domain, etc.).

[Read-through caching](#) - a sample of caching *all* resources that are requested "on the fly", unconditionally.

[Offline Google Analytics](#) - extends the [read-through caching](#) example to add in support for "replaying" failed Google Analytics pings, allowing pages to submit Google Analytics data associated with offline/cached page views.

[Fallback responses](#) - a sample illustrating how you can return alternative "fallback" content if an initial fetch request fails.

[Mock responses](#) - a sample illustrating how you can return content created on the fly in response to a page's requests.

[Using `postMessage`](#) - a sample illustrating the use of `postMessage()` to send commands from a controlled page to its service worker, giving the page control over the cache.

Basic Demos

[Wiki engine demo](#)

[isserviceworkerready spec tests](#)

[Cache All](#) - The bare minimum Service Worker to cache all requests from the app.

Substantial demos

[Offline News](#)

[Trained to Thrill](#)

[Tweetdeck offline](#)

This repo is just getting started

All the code in this repo is in the earliest stage of development. We are looking for people to help us create these apps!

Our goal is to show how you can perform tasks that are common in native applications using this new set of APIs. We want to keep the apps simple, so they can be easily bisected and copied by anyone.

Please note that the spec and implementations are in a very early stage! You may find issues with implementations in that they may often change or produce strange results at first. :

To see what actually works in browsers today, see Jake Archibald's [IsServiceWorkerReady](#) page.

How can you help?

The process below is a guideline to how to get started.

- Fork this repo!

- Choose one of the apps from the directories above.

- Use the app's behavior draft to implement the application.

- Keep it really simple! See our list of resources below to learn about service workers.

How to run these things?

We will be hosting them on a website soon, so you can see them as we build them.

Right now, you will need:

- [Chrome](#).

- [Opera](#).

- [Firefox Nightly](#).

Wanna build something different?

If you have a more interesting app you want to build, please add them to this repo. Just add a directory to the root directory with a README.md file explaining what the app does and send us a pull request!

##Resources

- [The specification](#).

Articles/Videos

- [ServiceWorkers and Firefox](#) - Introduction to ServiceWorkers on Firefox!

- [Service Worker - first draft published](#) - Introduction and tutorial including code snippets!

[Video: The ServiceWorker: The network layer is yours to own](#) - video introducing ServiceWorker including HTTP caching, request, and showing SW in forms, the basis for push messaging, alarms, geofencing and background sync.

Debugging

ServiceWorkerInternals (chrome canary only - <chrome://serviceworker-internals/>) - page for debugging ServiceWorkers in chrome, including stopping, unregistering and starting workers.

Examples

[Jake Archibald's Trained to Thrill](#) - Jake made a great demo app which pulls in images from Flickr, uses fetch, uses cache (not implemented yet) and worker registrations.

[Simple ServiceWorker](#) - A simple Service Worker example by Matt Andrews.

Related APIs

[Promises HTML5Rocks](#) - Understanding Promises (used frequently in ServiceWorkers especially in fetch, cache).

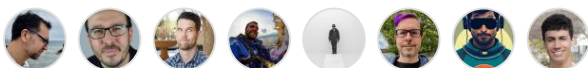
Releases

No releases published

Packages

No packages published

Contributors 8



Deployments 1

✓ **github-pages** 8 years ago

Languages

