

ToiletPaper #114

Using the browser's full potential

Author: Santo Pfingsten / Software Engineer / Office Leipzig

✘ Problem

Everyone knows how to use a browser to perform certain tasks. And yet (or maybe because of that) there are problems many people might work around, because they don't know the browser's full potential. Possible problems:

- You suspect that the browser cache, local storage or cookies are responsible for a problem and delete them to test this theory.
- You write debugging code to help analyzing the status (change) of the application.
- You have to log in to a web service (Gmail, Office365, ...) with different user IDs and therefore use a different browser for each user account.

✔ Solution

- Instead of clearing the browser cache, etc., simply use the browser's private/incognito mode. It strictly separates cache, local storage, cookies, etc. from the non-private tabs.
- So-called web extensions (formerly called browser add-ons) allow you to add functionality to the browser. Ad blockers are the best known, but there are also some special extensions for developers. These allow, for example, to monitor the Redux State.
- In Firefox, Mozilla has taken the encapsulation of private tabs one step further:
 - So-called container tabs (contextual identities) allow you to create new tabs, which run separately from the normal tabs – similar to the private mode. Unlike in private mode, cache, cookies, etc. are not forgotten when closing the tab.
 - That way, you can create separate containers – for the company, the customer and also for the private account. Simply press the plus in the tab bar a bit longer or search using the menu "File->New Container Tab".

➔ Example

Web extensions worth looking at:

- Container tabs: [Firefox Multi-Account Containers, Temporary Containers](#)
- Developer tools: [Augury \(Angular\), React Developer Tools, Redux DevTools, ColorZilla, CORS Everywhere, Nimbus Screenshot & Video Recorder, Page Ruler, Let's get color blind](#)
- Recording and playing back sequences: [iMacros, Selenium IDE](#)
- Inspect and edit cookies: [Cookie Quick Manager, EditThisCookie](#)
- Written by me:
 - [Forget Me Not](#) (Rule-based deletion of cookies, local storage and more)
 - [Dict.cc Translation](#) (In-page translation)

+ Further Aspects

If there is no suitable web extension available, it is relatively easy to write one yourself in JavaScript or TypeScript:

- Popups or sidebars can provide information or controls
- Panels can complement browser developer tools
- HTTP headers can be inspected and modified live. It is also possible to redirect requests to other URLs.
- Code and CSS can be injected into web pages to add missing functionality.
- Reading, saving and deleting cookies is part of the basics.

There are endless possibilities. Most features are available in Firefox, but all major browsers (Firefox, Chrome, Edge, Opera) support web extensions at least in part. Polyfills for cross-browser development can be found on NPM: JavaScript ([webextension-polyfill](#)) und TypeScript ([webextension-polyfill-ts](#))

API documentation: <https://developer.mozilla.org/docs/Mozilla/Add-ons/WebExtensions>