

# ToiletPaper #114

## Das ganze Potential des Browsers nutzen

Autor: Santo Pfingsten / Software Engineer / Standort Leipzig

### ✘ Problem

Jeder weiß, wie man einen Browser bedient, um bestimmte Aufgaben zu erledigen. Und dennoch (oder vielleicht genau deswegen) gibt es Dinge, bei denen viele Menschen um ein Problem drumherum arbeiten, da sie den vollen Umfang des Browser nicht kennen. Mögliche Probleme:

- Man hat den Verdacht, dass der Browser-Cache, Local-Storage oder Cookies für ein Problem verantwortlich sind und löscht diese deshalb, um diese Theorie zu testen.
- Man schreibt Debugging-Code, der helfen soll, den Status(-wechsel) der Applikation nachzuvollziehen.
- Man muss sich in einem Web-Service (Google Mail, Office365, ...) mit unterschiedlichen Nutzerkennungen einloggen und nutzt deshalb je Benutzeraccount einen anderen Browser.

### ✓ Lösung

- Anstatt den Browser-Cache, etc. zu löschen, einfach den privaten/inkognito Modus des Browsers nutzen. Dieser trennt strikt Cache, Local-Storage, Cookies, etc. von den nicht privaten Tabs.
- So genannte Web-Extensions (Früher: Browser-Erweiterung) erlauben es, Funktionalität zum Browser hinzuzufügen. Am bekanntesten sind Ad-Blocker, aber es gibt auch einige spezielle Erweiterungen für Entwickler. Diese ermöglichen es z.B., den Redux State zu beobachten.
- Die Abkapselung von privaten Fenstern hat Mozilla im Firefox eine Stufe weitergetrieben:
  - So genannte Container Tabs (Kontextuelle Identitäten) erlauben es, neue Tabs zu erstellen, die, ähnlich wie der private Modus, getrennt von normalen Tabs sind. Im Gegensatz zum privaten Modus werden Cache, Cookies, etc. beim Schließen nicht vergessen.
  - Man kann sich also jeweils einen Container für die Firma, den Kunden und auch für den privaten Account erstellen. Dazu einfach länger auf das Plus in der Tableiste drücken oder über das Menü "Datei->Neuen Container Tab".

### ➔ Beispiel

Web-Extensions, die man sich mal anschauen kann:

- 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](#)
- Abläufe aufzeichnen und abspielen: [iMacros, Selenium IDE](#)
- Cookies inspizieren und bearbeiten: [Cookie Quick Manager, EditThisCookie](#)
- Von mir geschrieben:
  - [Vergiss Mein Nicht](#) (Cookies, Local-Storage und mehr regelbasiert löschen)
  - [Dict.cc Translation](#) (In-Page Translation)

### + Weiterführende Aspekte

Falls es mal keine passende Web-Extension gibt, kann man sich relativ leicht eine Web-Extension in JavaScript oder TypeScript schreiben:

- Popups oder Sidebars können Informationen oder Bedienelemente bereitstellen
- Panels können die Browser Entwickler-Tools ergänzen
- HTTP Header können live inspiziert und verändert werden. Anfragen auf andere URLs umleiten ist auch möglich.
- Code und CSS kann in Webseiten injiziert werden, um fehlende Funktionalität zu ergänzen.
- Cookies lesen, speichern und löschen gehört zu den Basics.

Die Möglichkeiten sind endlos. Die meisten Features sind im Firefox verfügbar, aber alle großen Browser (Firefox, Chrome, Edge, Opera) unterstützen Web-Extensions zumindest in Teilen. Polyfills für Cross-Browser Entwicklung sind auf NPM zu finden: JavaScript ([webextension-polyfill](#)) und TypeScript ([webextension-polyfill-ts](#))

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