Lab: Exploring Local Storage

Estimated time needed: 15 minutes

Welcome to this lab, where you will learn how to access and manipulate local storage in web browsers, with a focus on Google Chrome. Local storage is a handy feature for saving key-value pairs without expiring with the session, making it useful for preserving user preferences, settings, and other data across sessions.

Objectives

After completing this lab, you will be able to:

- Access local storage in different web browsers, focusing primarily on Google Chrome
- Inspect and understand the features available in local storage using developer tools
- Add, modify, and delete data in local storage
- Interact with local storage using the JavaScript console

Exercise 1: Accessing local storage

This exercise will teach you how to access local storage in Google Chrome and other popular web browsers.

1. **Open Developer Tools in Chrome**

   - Right-click anywhere on a web page, select `Inspect`, or use the shortcut `Ctrl+Shift+I` (Windows/Linux) or `Cmd+Option+I` (Mac)

   - Navigate to the `Application` tab to access the local storage section under the `Storage` panel on the left sidebar
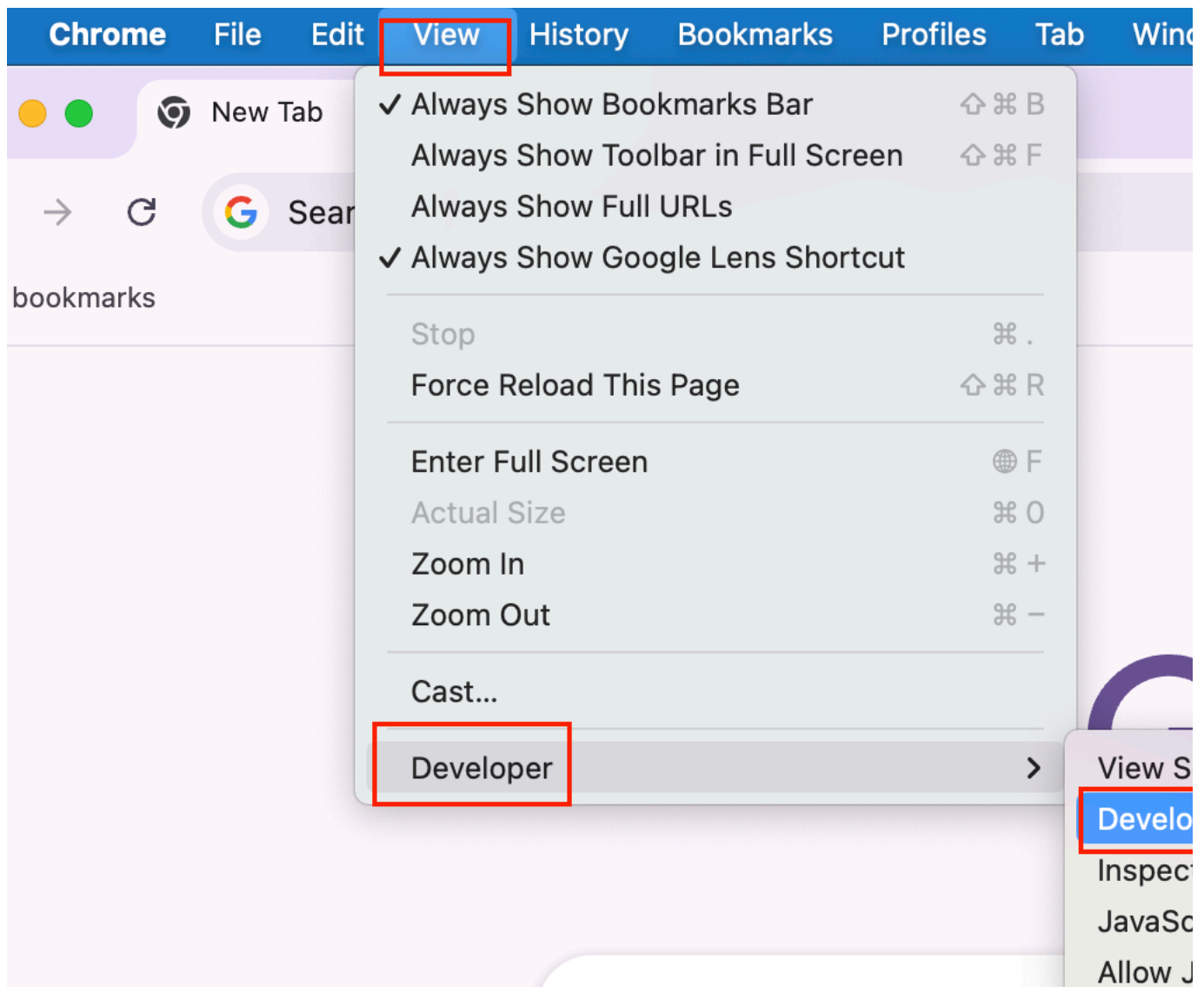
2. **Access local storage in other browsers**

   - **Firefox**: Open Menu > Web Developer > Storage Inspector or use the shortcut `Ctrl+Shift+I` and select the `Storage` tab.

   - **Safari**: Enable the Develop menu via Safari's Preferences, then select `Develop` > `Show Web Inspector` and go to the `Storage` tab.

⊹⊡ ⌂o | Elements Console Sources Network **Application** »

Application

- 📄 Manifest
- ⚙ Service workers
- 🗄 Storage

Storage

- ▼ ⊞ Local storage
  - ⊞ chrome-untrusted://new-tab-page
  - ⊞ chrome://new-tab-page
  - ⊞ https://ogs.google.com
- ▶ ⊞ Session storage
- 🗄 IndexedDB
- ▶ 🍪 Cookies
- 🗄 Private state tokens
- 🗄 Interest groups
- ▶ 🗄 Shared storage
- 🗄 Cache storage
- 🗄 Storage buckets

Background services

- 🗄 Back/forward cache
- ↑↓ Background fetch
- ↺ Background sync
- 🗄 Bounce tracking mitigations
- 🔔 Notifications
- ▭ Payment handler
- 🕐 Periodic background sync
- ▶ ↑↓ Speculative loads
- ☁ Push messaging
- 📄 Reporting API

Frames

- ▶ ▭ top

Local stor

Learn mor

Exercise 2: Exploring local storage

In this exercise, you will explore local storage by inspecting Wikipedia's homepage.

1. **Visit Wikipedia**

   - Go to https://www.wikipedia.org/ and open Developer Tools
     - Navigate to the `Application` tab and select `Local Storage` on the left-hand menu. You should see a list of key-value pairs currently stored by the browser for Wikipedia.

2. **Inspect local storage values**

   - Observe the entries in local storage. You might see keys like `portal_session_id` and `translationHash`, among others.

WIKIPEDIA
The Free Encyclopedia

English
6,873,000+ articles

日本語
1,427,000+ 記事

Русский
1 996 000+ статей

Español
1.974.000+ artículos

Deutsch
2.937.000+ Artikel

Français
2 631 000+ article

Italiano
1.879.000+ voci

中文
1,438,000+ 条目 / 條目

Português
1.132.000+ artigos

فارسی
۱٬۰۱۱٬۰۰۰+ مقاله

EN ⌄ 🔍

文A Read Wikipedia in your language ⌄

Exercise 3: Manipulating local storage

In this exercise, you will learn how to add, modify, and delete data in local storage using Developer Tools.

1. **Add new key-value pairs**

- In the `Local Storage` section of the `Application` tab, double-click an empty row in the table to add a new key-value pair.

- Enter a key name, for example, `testKey` and a value `testValue`.

- Press `Enter` to save the new key-value pair.

2. **Modify existing key-value pairs**

- Double-click the value of an existing entry. Change the value and press `Enter` to update.

3. **Delete key-value pairs**

- To delete an entry, right-click a key and select `Delete`.
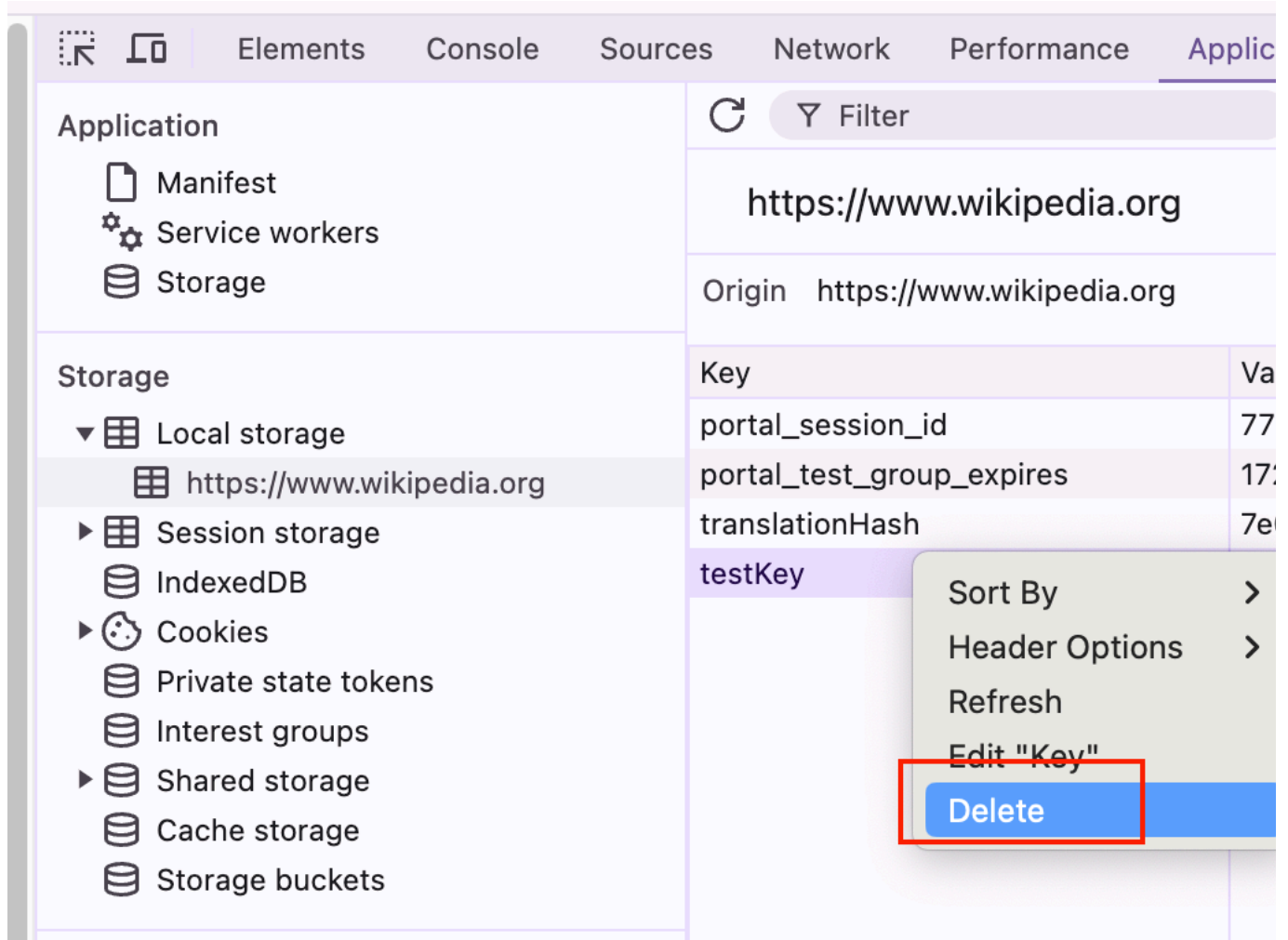
Exercise 4: Using the Console to interact with local storage

This exercise demonstrates how to use the JavaScript Console to interact with local storage programmatically.

1. **Open the Console tab:**

   - Switch to the `Console` tab in Developer Tools.

2. **Add a value:**

   - Type `localStorage.setItem('newKey', 'newValue');` and press `Enter`.
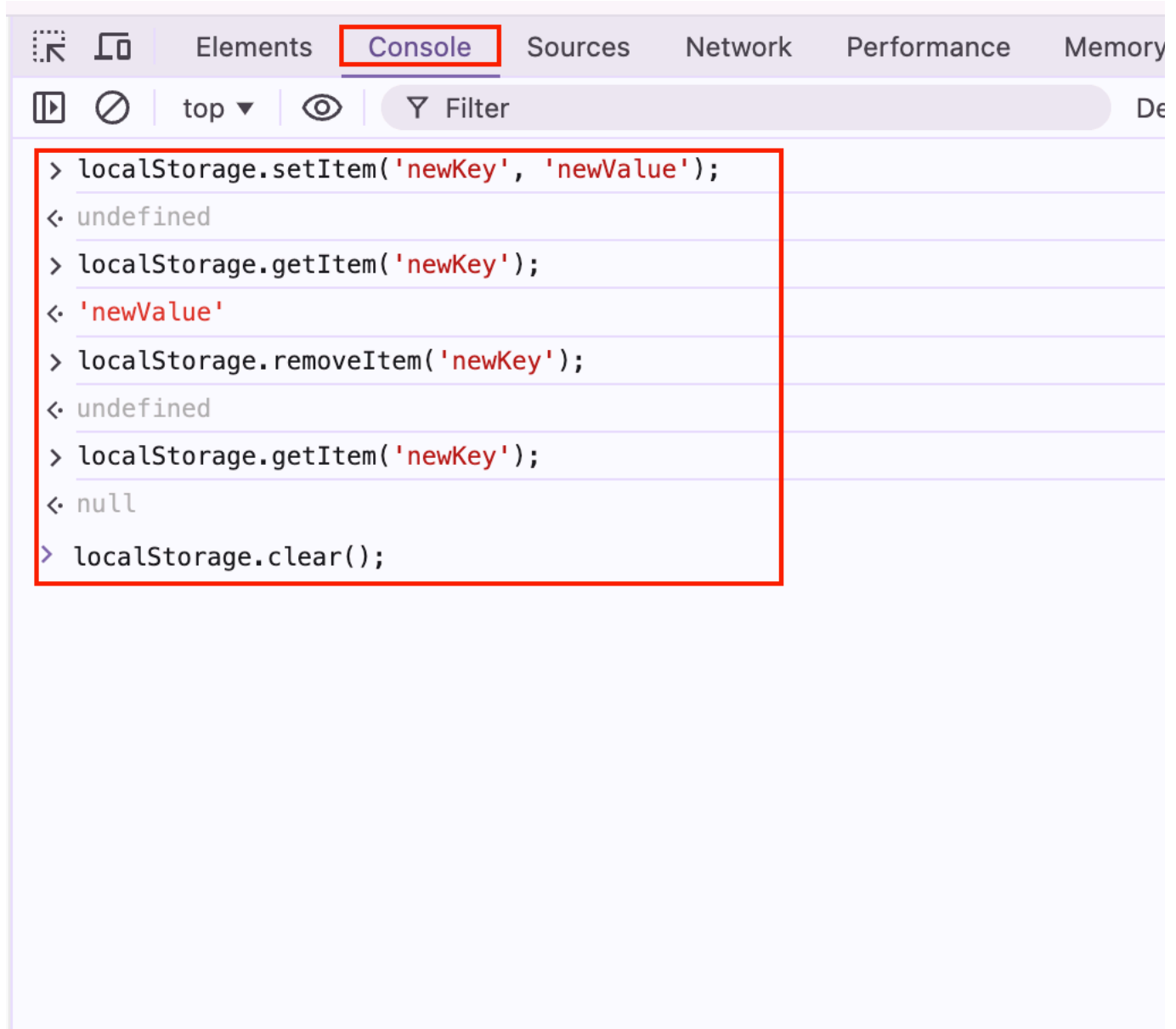
3. **Retrieve a value:**

   - To retrieve the value you just set, type `localStorage.getItem('newKey');` and press `Enter`.

4. **Remove a value:**

   - To remove the key you added, type `localStorage.removeItem('newKey');` and press `Enter`.

5. **Clear all local storage:**

   - If you want to clear all local storage entries for the site, type `localStorage.clear();` and press `Enter`.

Conclusion

In this lab, you have learned how to access and manipulate local storage in various web browsers, focusing on Google Chrome. You've explored how to manually add, retrieve, modify, and delete local storage data through the Developer Tools and programmatically using the Console.

By understanding local storage, you can enhance the user experience on your websites by persistently storing user preferences and other data across browser sessions.

Author(s)

Skills Network