RegEx Search & Replace Extension for Chrome and Firefox

Dalimil Hajek

BSc Hons Project Report
Computer Science
School of Informatics
University of Edinburgh

2018

Abstract

The aim of this project was to build a browser extension to allow users to search and replace text with regular expressions in editable text input fields of web pages.

After evaluating existing extensions that were unsuccessfully attempting to implement this functionality, the new extension has been carefully designed, developed, and finally successfully released for Chrome and Firefox browsers.

In addition to the future-rich search and replace function, this plugin also adds the ability to save favorite patterns, store search history, or predefine text templates that can be inserted into the editable area of a page.

The software followed an iterative development process, where user feedback was collected via several means, including Google Analytics, which was used to track user interaction, and a support website used to collect user feedback comments.

After the initial release, about twenty updates have been subsequently released over the span of a few months. This iteration was further supported by automated tests of several kinds.

The extension has received excellent reviews and at the time of writing has over 3000 weekly users (users from both browsers combined).

Acknowledgements

Thanks to

- Boris Grot - for supervising the whole project, and making important feature suggestions leading up to the first official releases of the extension on the Chrome and Firefox web stores
- Michael O'Boyle - for making suggestions, especially regarding Google Analytics
- Christoph Metze - for finding several important bugs that subsequently led to releases `1.3.2`, `1.3.3`, and `1.3.4`
- Daniel Tomberlin - for pointing out a use case when trying to search across multiple single-line inputs, and for updating his web store rating and review after I implemented it in `1.3.6`
- GitHub user MarkRH - for finding a bug that was later fixed in `1.1.3`
- StackOverflow user wOxxOm - for suggesting `Document.execCommand` API that I used to fix issues with templates in `1.2.0`

And also thanks to all those people who submitted user feedback or reviews.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Dalimil Hajek*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Search and replace functionality can be extremely useful when composing long emails, writing posts on social media, online forums, or blogging platforms, as well as in any email web clients.

The most frequent use cases are:

- Fixing a typographical error – A word or a phrase may have been used several times with the wrong spelling. This can reoccur several times in a forum post or an email, and it would be convenient to replace everything at once.

  This also includes fixing unreadable characters in blog entries such as $\hat{A}\in\mathring{?}$, due to a change in encoding or some unintended text handling.

- Normalizing incorrectly formatted text – Regular expressions can be used to detect formatting errors such as multiple spaces before a period, missing uppercase letter, various metric unit formatting errors, and similar. Search and replace extension supporting regular expressions can quickly find and fix these.

- Renaming a phrase – Often a word or a phrase that occurs several times throughout a text needs to be corrected or substituted (perhaps using a synonym or a wording that sounds better)

Without having a browser extension for search and replace, one could imagine a solution where all text is copied and pasted into an advanced text editor, fixed using the built-in search and replace function, and copied back into the web page input field.

In addition to being a lengthy and time consuming process, this method would in many cases lose all text formatting, because more advanced editable text elements on the web may contain images, emojis, and text containing many formatting tags, which would not be preserved during the copy-pasting.

Additional motivation behind the development of this project was to add search-and-replace related features that are missing even from the more advanced text editors.

One of them is storing the search history, and also being able to save favorite search patterns, that can later be quickly accessed. Both of these would save time and increase user productivity.

Chapter 2

Background

Chapter 3

Design

Chapter 4

Implementation

Chapter 5

Evaluation

Chapter 6

Conclusions