Documentation

Topic: Chrome “Find in webpage” with auto-correction, stemming, and synonym detection

Team members: Xiaohe Cheng (xiaohec3@illinois.edu) - Project coordinator

Overview

The project implements a Chrome extension that provides more powerful in-page search ability. Search within a webpage (“Find”, Ctrl+F) is a very common use case in web browsing. Chrome’s built-in search finds all text occurrences within a page that precisely match user input. This Chrome extension enhances the search usability by allowing users to further search for:

- The auto-corrected version of the input word
- Other word forms of the input word
- Synonyms of the input word

Implementation

The Chrome extension is implemented with HTML, CSS, and JavaScript language. Implementation-wise, it consists of the following components:

- Manifest file: A metadata file for the extension, defining the name, icon, dependency, etc.
- Pop-up window: An HTML page that shows up when users click the extension icon. A CSS file defines the style of the window.
- Search button controlling logic: JavaScript logic that defines the callback when users click the search button:
    - Modify the pop-up window, removing the search button and adding previous button and next button
    - Get user’s query and compute the auto-corrected word, stemming, and synonyms
    - Search in the page to find all matched text and highlight them by modifying the page HTML
- Previous and next logic: JavaScript logic that defines the callback when users click the previous or next button, implementing the “jumping” effect

The computation of auto-corrected word, stemming, and synonyms uses the following npm packages:

- autocorrect: https://www.npmjs.com/package/autocorrect
- porter-stemmer-english: https://www.npmjs.com/package/porter-stemmer-english
- synonyms: https://www.npmjs.com/package/synonyms

One difficulty is that the browser JavaScript environment does not allow importing npm packages. To get around that, I created a fake npm application and used browserify and babel to transpile the functions that use the packages, creating a helper JavaScript file. I mainly followed this guide:

https://hackernoon.com/use-es6-javascript-syntax-require-import-etc-in-your-front-end-project-5eefcef745c2

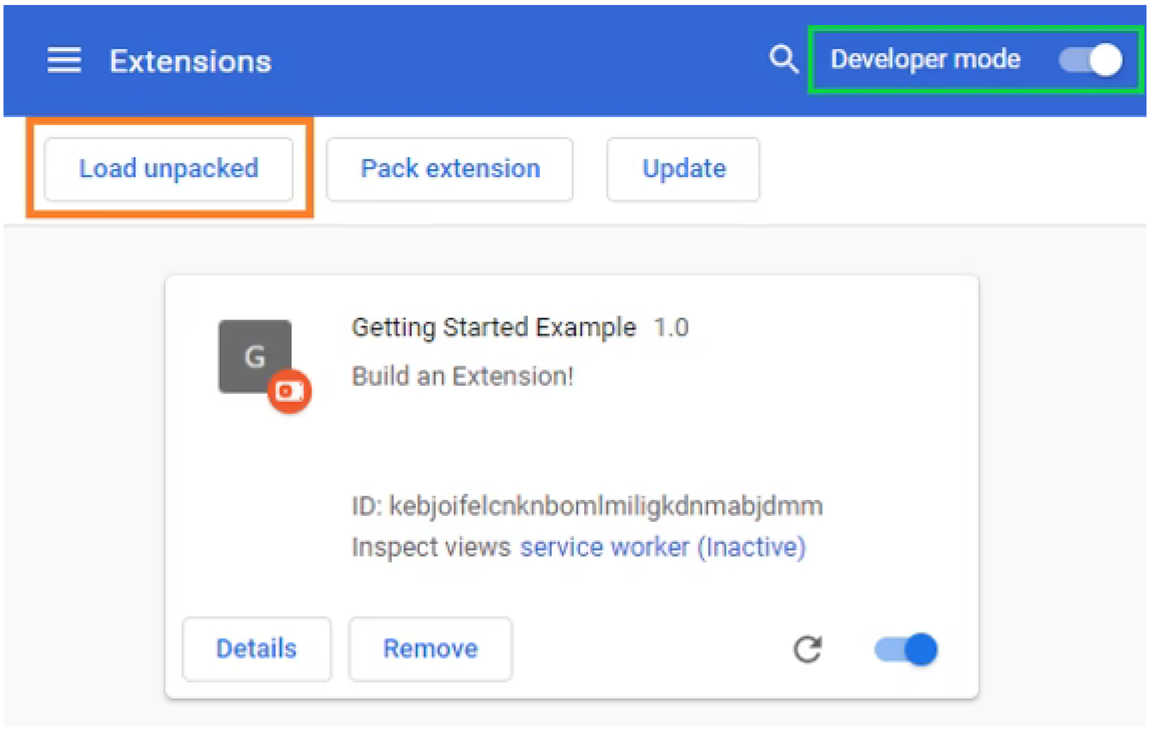
Potential further improvements of the project include:

- Expand the coverage of HTML elements. Currently, the extension fails to search for text in some HTML elements. Writing more sophisticated JavaScript logic will give better coverage.
- Allow users to choose from candidate words. Although the current implementation allows users to turn on or off each of the three features, sometimes users may want more fine-grained control. For example, they may want to remove a synonym from the search list, but keep others.
- Package the extension and distribute it.

Usage

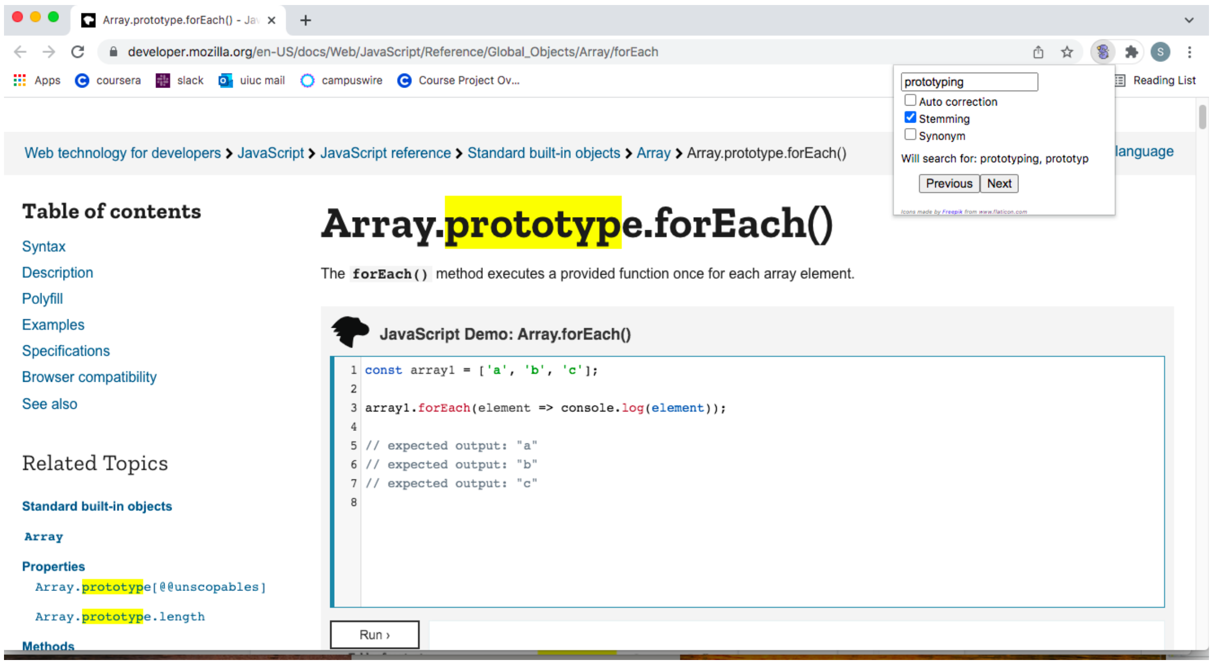
Step 1: Download the source code. Download all code and assets in https://github.com/xihec3/CS410_CourseProject/tree/main/code/bright_search. Put them inside a directory.

Step 2: Load the extension. Open Chrome and go to chrome://extensions. Toggle developer mode on and click "Load Unpacked". Select the directory created in Step 1.


Extensions
Developer mode
Load unpacked
Pack extension
Update
Getting Started Example 1.0
Build an Extension!
ID: kebjoifelcnknbomlmiligkdnmabjdmm
Inspect views service worker (Inactive)
Details
Remove

Step 3: Open the pop-up window. Go to the page you want to search in (example: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach) and click the extension icon (a purple one) in the extension bar.

Step 4: Enter input and start searching. Enter a word in the text box and set options (example: prototyping, with stemming on) , then click “Search in page”. The matched words will be highlighted.



Array.prototype.forEach() - Ja
developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach
Apps
coursera
slack
uiuc mail
campuswire
Course Project Ov...
Reading List
Previous
Next
Web technology for developers > JavaScript > JavaScript reference > Standard built-in objects > Array > Array.prototype.forEach()
language
Table of contents
Syntax
Description
Polyfill
Examples
Specifications
Browser compatibility
See also
Related Topics
Standard built-in objects
Array
Properties
Methods
JavaScript Demo: Array.forEach()
Run ›

prototyping
Auto correction
[✓ Stemming]
Synonym
Will search for: prototyping, prototyp

Array.prototype.forEach()

The `forEach()` method executes a provided function once for each array element.

```
const array1 = ['a', 'b', 'c'];

array1.forEach(element => console.log(element));

// expected output: "a"
// expected output: "b"
// expected output: "c"
```

`Array.prototype[@@unscopables]`

`Array.prototype.length`

Step 5: Navigate the search results. Click “Previous” and “Next” to jump between search results.