

AGH

Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji
Kierunek Informatyka

Inżynieria Oprogramowania
2019



Memento

Rozszerzenie przeglądarki do przygotowywania
oraz zarządzania zebranymi cytatami.

Mateusz Buta

Kamil Janda

Bartosz Pietrzyk

Paweł Zeller

Dokumentacja deweloperska

Spis treści

Omówienie zakresu realizacji	4
Rozwiązania zastosowane w produkcie	4
Architektura systemu	4
Komunikacja	5
Przechowywanie danych	6
Wbudowana strona ustawień wtyczki	6
Uruchomienie wersji deweloperskiej	6
Problemy i jak je sprytnie rozwiązaliśmy	6
Ulepszenia lub poszerzenia zakresu zastosowań	7
Możliwości rozbudowy, alternatywne sposoby realizacji	8
Repozytorium	9
Autorzy	9

Omówienie zakresu realizacji

Projekt został zrealizowany jako projekt studencki w ramach przedmiotu Inżynieria Oprogramowania w roku 2019 na szóstym semestrze studiów inżynierskich.

W ramach realizacji projektu konieczne było dokładne zapoznanie się z budową oraz działaniem mechanizmu rozszerzeń w przeglądarce Firefox. Budowa rozszerzenia skupiała się głównie na api dostarczonym przez Firefoxa jednak wykorzystane zostały też zewnętrzne biblioteki takie jak jQuery które w znaczący sposób wpłynęły na wizualny odbiór rozszerzenia np. mechanizm ustalania parametrów.

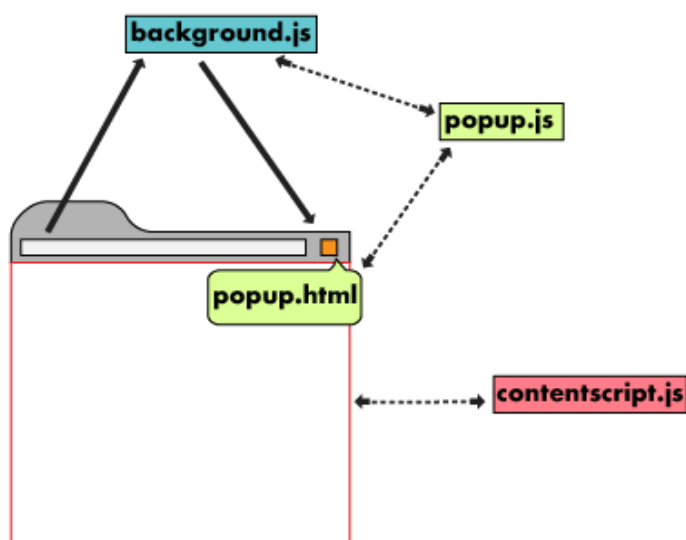
Rozwiązania zastosowane w produkcie

Wtyczka jest kompatybilna z przeglądarką Mozilla Firefox. Rozszerzenia dla Firefox są budowane przy użyciu Web Extension APIs, czyli systemu wspólnego dla przeglądarek do rozwoju rozszerzeń.

Stosowanym językiem jest JavaScript, ponieważ jest to język wspierany przez WebExtension API. Dodatkowo popularność i prostota tego języka, pozwoliła na szybkie wdrożenie zespołu w nową technologię.

Architektura systemu

Nasz system będzie reprezentuje typową dla rozszerzeń przeglądarkowych architekturę.



Komponenty systému:

- ## 1. background.js

Skrypt główny wtyczki odpowiedzialny za tworzenie, edycję i usuwanie notatek oraz ich wstrzykiwanie do wybranego pliku.

- ## 2. popup.js

Skrypt odpowiedzialny za działanie wyskakującej okna wtyczki (popup), odpowiada zwłaszcza za interakcję z użytkownikiem

- ### 3. popup.html

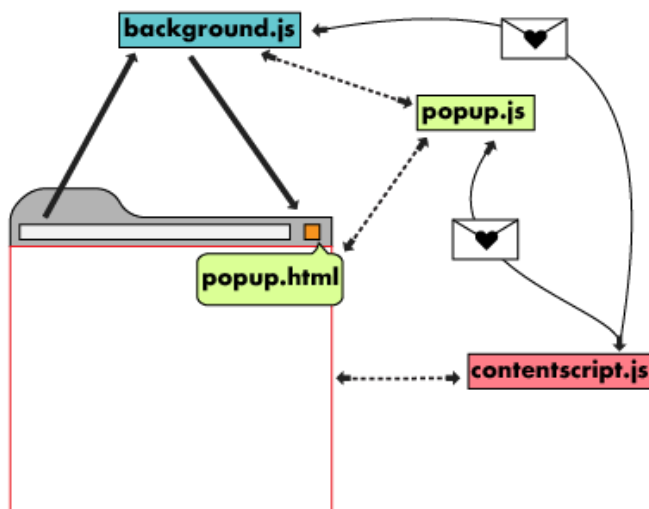
Odpowiada za wizualne walory okna wtyczki. Wzbogacony o arkusze stylów Css.

- #### 4. content-script.js

Istotne w naszej wtyczce skrypty działające w tle każdej strony, zbierające informacje o zaznaczonym tekście.

Komunikacja

Warto zwrócić uwagę na komunikacja między skryptami działającymi w tle, a *popupem*. Została ona zrealizowana przy pomocy mechanizmu wiadomości *Message* który umożliwia synchroniczne lub asynchroniczne wysyłanie i nasłuchiwanie wiadomości przez skrypty działające w ramach rozszerzenia.



Wyskakujący popup, wysyła wiadomość do skryptu w tle, z prośbą o odesłanie zaznaczonego elementu. Skrypt w tle odsyła w odpowiedzi treść zaznaczenia, która jest wstrzykiwana do pola tekstowego popupa.

Przechowywanie danych

Istotny jest również sposób przechowywania danych, w specjalnej bazie danych udostępnianej przez przeglądarkę Firefox, czyli *Storage*. Udostępnia ona odpowiednie API do komunikacji z bazą danych. Baza umożliwia przechowywanie danych tylko w formie słownika.

Wbudowana strona ustawień wtyczki

Przygotowana przez Mozillę wbudowana strona do konfiguracji ustawień wtyczki została przez nas wykorzystana do konfiguracji markera i wprowadzenia możliwości importu i eksportu notatek.

Uruchomienie wersji deweloperskiej

Aby uruchomić wtyczkę w wersji deweloperskiej, należy wejść na stronę debugowania, poprzez wpisanie w pole url w przeglądarce Firefox adresu *about:debugging*. Po naciśnięciu przycisku *Load Temporary Add-on* należy wskazać plik *manifest.json*, który spowoduje tymczasowe załadowanie wtyczki.

Alternatywą jest skorzystanie z narzędzia deweloperskiego [web-ext tool](#). Należy przejść do katalogu z kotem wtyczki i wpisać w terminalu *web-ext run*. Spowoduje to uruchomienie okna przeglądarki i automatyczną instalację wtyczki. Wtyczka będzie aktualizowana po każdorazowym zapisie po edycji kodu.

Problemy i jak je sprytnie rozwiązaliśmy

1. Wstrzykiwanie notatki z popupa

Jest to znany błąd po stronie Mozilli, przy kliknięciu na przycisk dodawania pliku, pojawia się okno ładowania pliku. W tym momencie popup się automatycznie zamyka, razem z oknem dodawania plików. Obeszliśmy to w ten sposób, że po naciśnięciu, zadanie jest kierowane do skryptu działającego w tle strony, który ją modyfikuje i wymusza pojawienie się okna do ładowania pliku.

discourse.mozilla.org

bugzilla.mozilla.org

2. Przechowywanie konfiguracji

Ustawienia konfiguracyjne wtyczki, takie jak marker, czy treść bieżąco dodawanej

notatki są również przechowywane w *Storage*, pod odpowiednimi, zastrzeżonymi kluczami. Warto rozważyć udoskonalenie tego mechanizmu, bo miejsce przechowywania konfiguracji wydaje się właściwe

Ulepszenia lub poszerzenia zakresu zastosowań

1. Dodanie typów kategorii

Potencjalnym ulepszeniem byłby dodanie różnych typów kategorii, które ułatwiały sortowania notatek. Użytkownik przypisywałby notatkom kategorie. Warto również rozważyć mechanizmy dziedziczenia kategorii.

2. Wiele markerów

Wprowadzenie kilku kategorii umożliwiłoby definiowanie markerów dla poszczególnych kategorii, które naturalnie sortowałyby notatki przy wstrzykiwaniu do pliku.

3. Definiowanie formatów przez użytkownika

Oddanie w ręce użytkownika możliwości definiowania własnych formatowań, sprawi że nasze narzędzie będzie elastyczne i da w zasadzie nieograniczone możliwości. Warto pozostawić kilka predefiniowanych formatów. Jest to bardzo istotne w przypadku ekspansji wtyczki na inne kraje.

4. Konfigurowanie ustawień wtyczki

Obecnie odbywa się w dwóch miejscach. Niektóre elementy, takie jak kolejność wstrzykiwanych elementów notatki jest dostępna w menu bocznym w popupie, natomiast ustawienia makera, import i eksport jest dostępny w oknie ustawień wtyczki. Warto rozeznąć wśród użytkowników, czy takie rozwiązanie jest optymalne.

5. Wspieranie popularnych formatów cytatów

Niektóre popularne formatowania takie jak BBCode powszechnie występują na stronach i forach internetowych. Warto rozszerzyć wtyczkę o możliwość wyboru wśród uschematyzowanych formatów cytowań: `[quote=Bjarne]Quoting someone in particular[/quote]`.

6. Podmienianie ukrytych linków

Podmiana niewidocznych linków np link “kliknij mnie” powinien zostać zamieniony na bezpośredni adres url, aby nie utracić informacji.

7. Tłumaczenie wtyczki na różne wersje językowe
Przetłumaczenie przycisków i komunikatów na inne popularnie występujące języki.
8. Lepsza integracja z użytkownikiem
Dodanie wyskakujących komunikatów w kilku miejscach, np zabezpieczenie przycisku usuwającego wszystkie notatki poprzez wymaganie dodatkowego potwierdzenia wykonywanej operacji.
9. Automatyczny backup
Dodanie możliwości automatycznego eksportowania notatek w formacie json, do wskazanego miejsca, na wypadek utraty notatek przechowywanych we wtyczce.
10. Pobieranie stron internetowych lokalnie
Przechowywanie zapisanych stron w celu szybkiej wzrokowej weryfikacji zgromadzonych cytatów z danej strony internetowej. Takie cytaty musiałyby ściśle odpowiadać kopiowanej treści.
11. Obsługa plików
Przysporzyła nieco kłopotów, ponieważ wtyczka przeglądarkowa zasadniczo nie ma dostępu do systemu plików. Warto rozeznąć temat głębiej, czy nie da się tego zrobić lepiej i uniknąć pewnych sztuczek które stosujemy.

Możliwości rozbudowy, alternatywne sposoby realizacji

1. Nowe mechanizmy eksportu notatek
Możliwość eksportu notatek to główna zaleta naszej wtyczki, która może odciągnąć użytkowników od podobnych narzędzi które tylko gromadzą notatki. Warto przemyśleć i poszukać innych możliwości eksportu, zwłaszcza wśród nowych użytkowników wersji 1.0.0.
2. Kompatybilność z przeglądarką Google Chrome
Naturalnym krokiem dalszego rozwoju projektu jest udostępnienie go również użytkownikom Google Chrome. Rozszerzenia dla Firefox są budowane przy użyciu Web Extension APIs, czyli systemu wspólnego dla przeglądarek do rozwoju rozszerzeń. Dodanie kompatybilności z przeglądarką Google Chrome pozwoliłoby zwiększyć grupę potencjalnych użytkowników. Wymagało to dużych nakładów pracy

z uwagi na różnice w api obu przeglądarek, a ograniczony czas realizacji projektu pozwolił nam na zrealizowanie rozwiązania tylko dla Firefoxa.

3. Aplikacja desktopowa

Pozwalająca na zarządzanie notatkami poza przeglądarkami, oraz umożliwiające dużo lepsze zarządzanie plikami do których będą eksportowane notatki.

4. Realizacja w TypeScript

Jednym z możliwych alternatywnych sposobów realizacji byłaby możliwość użycia nadzbioru języka JavaScript, czyli TypeScript. TypeScript likwiduje problemy wynikające z braku statycznego typowania, które znacząco spowalniały tempo prac.

Repozytorium

<https://github.com/agh-ki-io/Memento>

Autorzy

Mateusz Buta

Kamil Janda

Bartosz Pietrzyk

Paweł Zeller