Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2 , 842 16 Bratislava

Extrakcia dát z webu

[WebExtraction]

Modul Browser extension

Tím: č. 16, WebX
Vedúci tímu: Ivan Srba
Členovia tímu: Ján Brechtl, Tomáš Juhaniak, Martin Kalužník, Rastislav Krchňavý, Michal Kren, Martin Lacek, Andrej Vaculčiak
Akademický rok: 2016/2017
Autor: Tomáš Juhaniak, Martin Kalužník
Verzia číslo: 2.2
Dátum poslednej zmeny: 15.05.2017

1 Úvod

Obsahom tohto dokumentu je opis funkcionality, návrh a implementácia rozšírenia pre Google Chrome v podobe plug-inu alebo inak povedané extensionu.

Ide o samostatnú aplikáciu, komunikujúcu s webovou časťou systému, ktorej poskytuje funkcionalitu spomenutú v podkapitole Analýza.

2 Browser extension

2.1 Analýza

Na výber konkrétneho skriptu a anotáciu dát je potrebné rozšírenie do prehliadača. Toto rozšírenie musí poskytovať overenie totožnosti, resp. prihlásenie používateľa, výber konkrétneho projektu a skriptu a možnosť jednoducho zadefinovať skript.

2.2 Návrh

Rozhodli sme sa vytvoriť rozšírenie pre prehliadač Google Chrome. Okrem klasického popup okna vyvolaného z ikony vpravo od adresového panelu, sme sa rozhodli väčšinu logiky implementovať takzvaným DevTools rozšírením. Používateľské rozhranie takéhoto rozšírenia sa nachádza v okne, ktoré sa zobrazí po stlačení F12. Jeho hlavnou výhodou je, že neprekrýva obsah webovej stránky, vďaka čomu sa môžme v budúcnosti vyhnúť problémom s prekrývaním obsahu stránky.

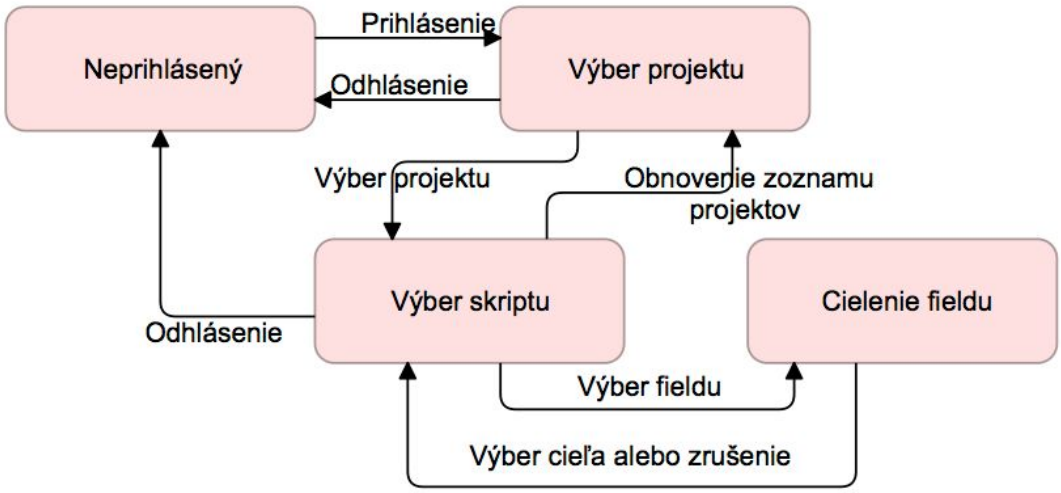
Na získanie informácii o používateľovi sme sa namiesto prihlasovania rozhodli použiť autorizáciu, v budúcnosti to uľahčí tvorbu rozšírení do iných prehliadačov a hlavne to umožní tretím stranám tvorbu aplikácií, ktoré dokážu spolupracovať s naším systémom bez toho, aby sme do neho museli znova zasahovať.

2.3 Implementácia

Na komunikáciu medzi aplikáciou rozšírením sme vytvorili REST API pomocou gemu ‘grape’. Autentifikáciu používateľa realizujeme pomocou OAuth2 protokolu, metódou implicitného toku, kedy musí používateľ autorizovať našu browser extension na vykonávanie požiadaviek v jeho mene.

Rozšírenie je postavené na použití javascriptového rámca AngularJS. Vďaka nemu je rozšírenie možné implementovať ako SPA. Zobrazenie jednotlivých súčastí funguje na princípe stavov. Rozlišujeme stav prihlásený / neprihlásený. V stave prihlásený rozšírenie rozlišuje stavy podľa naplnenia modelov projektov, skriptov a data schém. Po zvolení nového projektu sa modely skriptov a data schém znovu načítajú a predtým zvolené modely sa

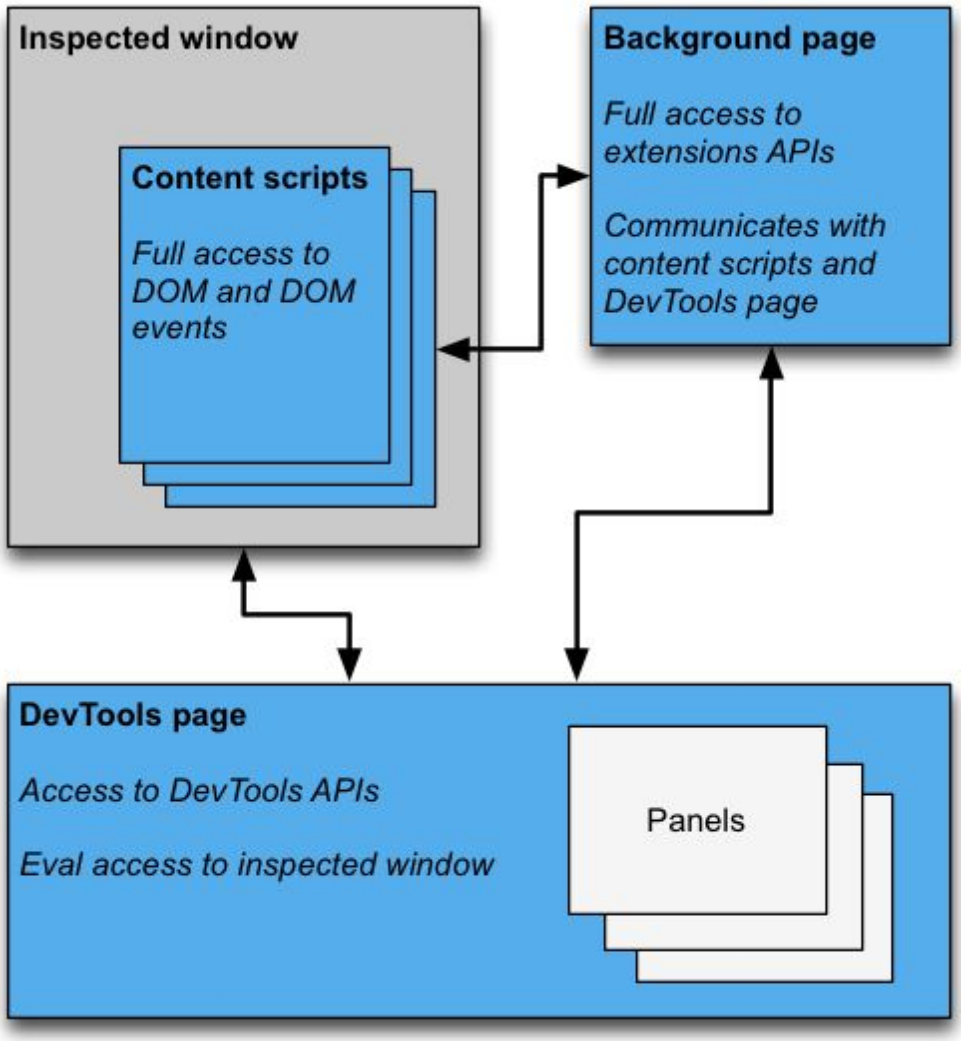
resetujú.



Obr. 1 - Stavový diagram súčastí rozšírenia do prehliadača

Na to, aby používateľ mohol interaktívne definovať, ktoré údaje sa majú extrahovať a ako majú byť spracované, potrebujeme, aby naše rozšírenie dokázalo komunikovať a upravovať obsah aktuálne zobrazenej webovej stránky. Toto je realizované pomocou takzvaných "content skriptov"[1] (javascript súborov), ktoré rozšírenie dokáže vložiť a nechať vykonať v kontexte zobrazenej stránky. Vďaka obmedzeniam kladeným na DevTools stránku rozšírenia, tá nedokáže "content skripty" vkladať priamo. Preto to musí byť sprostredkovávané pomocou "background page"[2] stránky. Ako prebieha takáto komunikácia medzi komponentmi rozšírenia možno vidieť na obr. 2.

[1] https://developer.chrome.com/extensions/content_scripts
[2] https://developer.chrome.com/extensions/background_pages



Obr. 2 -Diagram komunikácie medzi súčasťami rozšírenia

2.4 Testovanie

Testovanie tohto modulu zatiaľ nebolo vyžadované (výnimka v DoD) a teda nebolo riešené. Zatiaľ je v stave prototypu a v nasledujúcom semestri ho čakajú veľké zmeny.

2.5 Technická dokumentácia

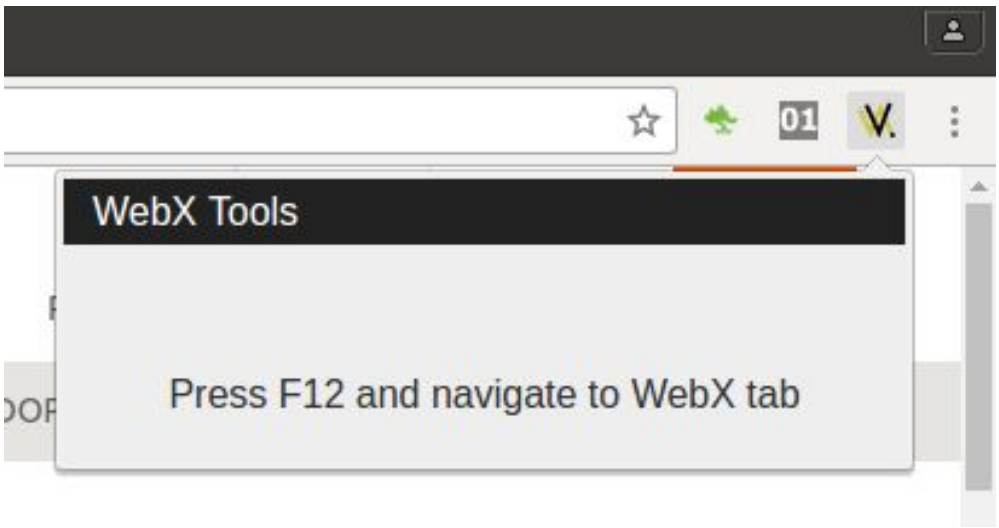
Skript prijatý zo serveru extension rozparsuje do vlastnej dátovej štruktúry, ktorej logika je zoskupená v súbore "js/script_builder.js". Ten má na starosť aj tvorbu výsledného skriptu v JSON formáte.

Jednotlivé možnosti postprocessingu sú implementované v samostatných triedach tak, aby bolo možné jednoducho pridávať nové typy postprocessingov. Teda bez toho, aby programátor musel rozumieť tomu ako rozšírenie spracuje skript prijatý zo servera alebo ako ho znova dostane do podoby na poslanie serveru. Inštrukcie na pridanie nového postprocessingu sa nachádzajú v kapitole 2.6.

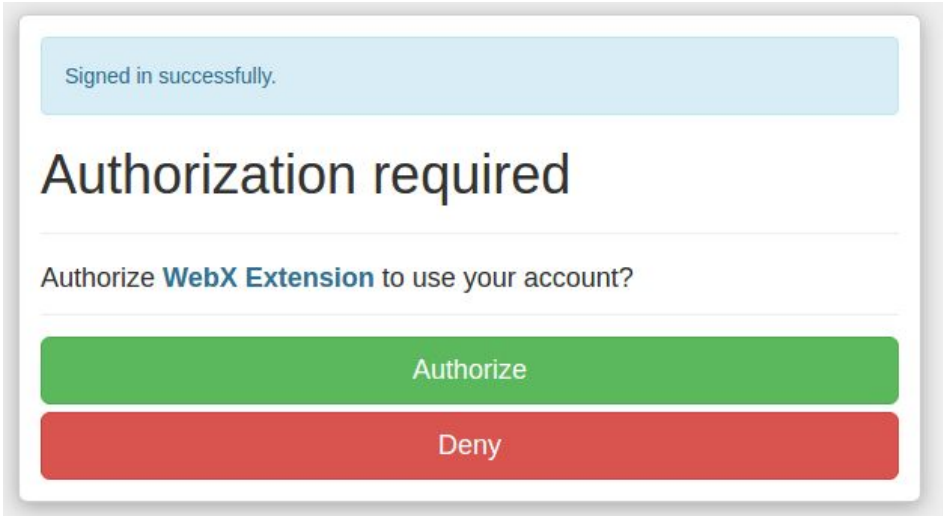
2.6 Pridávanie nového postprocesingu

Pre každý typ postprocessingu existuje súbor v “js/postprocessing” priečinku s triedou reprezentujúcou konkrétny postprocessing. Pri vytváraní nového postprocessingu odporúčame ako šablónu použiť ukážkový kód nachádzajúci sa v súbore “js/postprocessing/postprocessing.js”. Podľa komentárov v šablóne programátor upraví potrebné názvy, pridá atribúty potrebné pre nový postprocessing, prípadne naprogramuje udalosti, ktoré sa majú vykonať pri zobrazení a skrytí postprocessingu. Okrem pridania tohto jedného súboru je ešte potrebné upraviť súbor “devtools.html”, zahrnúť novo-vytvorený javascript súbor, pridať html elementy a namapovať ich pomocou rámca AngularJS aby zobrazovali atribúty vytvoreného postprocessingu. Bližšie informácie aj s ukážkami kódu sa nachádzajú pri šablóne postprocessingu v “js/postprocessing/postprocessing.js”.

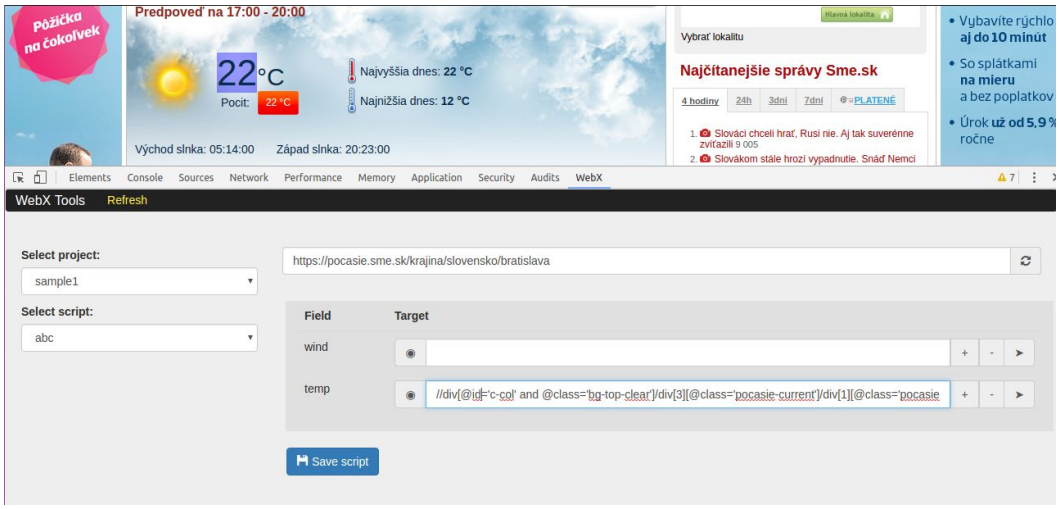
3 Ukážka modulu browser extension



Obr. 3 - Pop-up okno s výzvou na otovrenie DevTools



Obr. 4 - Výzva na autorizáciu extrension



Obr. 5 - Príklad výberu elementu na stránke

Obr. 6 - Príklad výberu a určovania postprocessingov