



**FAKULTA
INFORMAČNÍCH
TECHNologiÍ
ČVUT V PRAZE**

Bakalářská práce

Webová aplikace pro online web scraping

Jakub Drahoš

Obor Webové a softwarové inženýrství (BI-WSI), zaměření Softwarové inženýrství

Katedra softwarového inženýrství

Vedoucí práce: Martin Podloucký

14. dubna 2019

Úvod

Klíčová slova web scraping, extrakce dat, aplikace, JavaScript, rozšíření do Chromu, právní rozbor

Keywords web scraping, data extraction, application, JavaScript, Chrome extension, legal analysis

Cíle práce

Hlavním cílem této práce je návrh a tvorba webové aplikace, která bude umožňovat uživatelům extrahovat požadovaná data z libovolné stránky v reálném čase bez jakékoli nutné znalosti programování. Při specifikaci požadavků tohoto softwaru se přihlídně k analýze stávajících řešení, jež je vedlejším cílem této práce. Druhým vedlejším cílem je poskytnout čtenáři úvod do právní problematiky web scrapingu a shrnout na jednom místě fakta, která máme k dispozici.

Neméně důležitou součástí práce tvoří dodržení klasického vývojového cyklu softwarového projektu – analýza, design, implementace a testování.

Klíčovým aspektem aplikace je též *přehlednost a jednoduchost uživatelského rozhraní* – důraz bude kladen na intuitivní a rychlé ovládání.

Naopak v rozsahu této práce není tvorba web crawlera ani žádného jiného podobného mechanismu, jenž by systematicky a především *automatizovaně* procházel danou oblast webu.

Motivace

Téma web scrapingu je v dnešní době velice aktuální a čím více dat produkujeme, tím více bude stoupat potřeba tyto informace určitým způsobem získávat a zpracovávat. Téměř kdokoli, kdo pracuje s daty dostupnými z inter-

netu, bude nucen využít nějaký nástroj k vytěžování, aby byl vůbec schopný udržet krok s konkurencí.

Tedy důvod k vytvoření softwaru umožňující extrahovat data z webových stránek je jasný. Ač podobných nástrojů existuje několik, jejich obsluha je poměrně složitá a je nutné strávit určitý čas, než se člověk seznámí s jejich fungováním a může je naplno využít. Právě tento aspekt se snaží aplikace vyvíjená v rámci této bakalářské práce eliminovat – motivací je tak poskytnout uživatelům možnost jednoduše a rychle vytěžit požadovaná data bez zbytečného zdržování a dlouhého času stráveného seznamováním se s nástrojem.

Jak již bylo řečeno, přínos aplikace spočívá především v její jednoduchosti. Z toho mohou těžit hlavně uživatelé, kteří se nezabývají programováním nebo tvorbou webových stránek. Využije ji tak kdokoli, kdo potřebuje jednorázově získat data z libovolné internetové stránky, která obsahuje velké množství dat pohromadě na jedno místě. Z důvodu prozatím chybějícího crawlingu (automatizovaného procházení) je naopak nevhodná k pravidelnému získávání dat (jako je například dlouhodobé sledování cen produktů) či ke zpracování stránek, kdy se jednotlivá data nacházejí rozptýlená po celé doméně.

To, čím je tato práce unikátní, je ale rozbor právního aspektu web scrapingu. V žádném případě se nejedná o hlubokou analýzu, která by byla očekávána od studenta právnické fakulty. Zároveň ale shrnuje podstatné poznatky a fakta z dané oblasti na jednom místě. Při hledání tohoto tématu na internetu totiž uživatel narazí na jeden nešvar – téměř vše začíná odstavcem ve stylu „...nejsem právník a toto je jen můj názor...“, informace jsou velmi kusé a chybí jim nějaká ucelená struktura. Tento neduh se práce snaží napravit a představuje tak vstupní bránu do této rozsáhlé problematiky. Zároveň může pomoci všem vývojářům, kteří tvoří software určený k web scrapingu.

Členění práce

Kapitola 1 je věnována analýze tematiky web scrapingu. První sekce shrnuje obecné informace, následuje pohled z právní strany věci a nakonec analýza stávajících řešení problému. Kapitola 2 se zaměřuje na návrh aplikace – specifikace požadavků, architektura systému, návrh uživatelského rozhraní. Ve 3. kapitole je popsána realizace daného návrhu, výběr použitých technologií a odůvodnění rozhodnutí, která byla učiněna. Poslední kapitola je věnována testování celé aplikace.

Analýza

1.1 Analýza konkurenčních nástrojů

První skupinou, na kterou můžeme při hledání na internetu narazit, jsou společnosti, které nabízejí zákazníkům kompletní péči v rámci extrakce dat. Cílí především na velké korporace, jimž postaví scrapovací nástroj přesně na míru, který poté také hostují a spravují. Zákazník tedy dostane data a o nic víc se již nemusí starat. Jako příklad lze jmenovat třeba ContentGrabber, Mozenda a další.

Pro tuto práci mnohem relevantnější kategorií je konkurenční nabídka nástrojů poskytujících uživatelům rozhraní k web scrapingu. Zaměříme se pouze na takové nástroje, které nevyžadují jakoukoli znalost programování – tedy žádné knihovny, API a nástroje pro budování vlastních scraperů.

Mezi ty největší představitele patří ParseHub, Octoparse, WebScaper, Data Scraper a Dexi.io. Čtyři ze zmíněných nástrojů jsou volně dostupné (které mají však velmi omezenou funkcionalitu a pokročilejší operace se odeknou až s určitým platebním plánem – tzv. freemium model) a jeden poskytuje bezplatně pouze 7denní zkušební verzi.

Předtím, než bude možné jednotlivé nástroje porovnávat, je nutné určit kritéria, podle kterých lze hodnotit kvalitu daného nástroje. Především půjde o jednoduchost používání, celkovou přehlednost a rychlost, se kterou se uživatel dostane k požadovaným datům. Důležitý je také způsob výběru dat, možnosti exportu získaných dat, jak aplikace sama dokáže uživatele seznámit s používáním a také, v jaké formě se nástroj vůbec používá a čím se od ostatních odlišuje (ať už v pozitivním či negativním smyslu).

Pojďme se tedy na některé nástroje podívat blíže:

1.1.1 ParseHub

Výhody:

1. ANALÝZA

- výběr dat jak pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí), tak pomocí XPath, regulárních výrazů nebo CSS selektorů
- aplikace obsahuje interaktivní tutoriál, který na jednoduchých příkladech ukáže, jak s nástrojem zacházet
- možnost získání dat různými formami - přes API, jako CSV/XLS, do GoogleSheets nebo do Tableau
- různé módy kliknutí (výběr, relativní výběr, kliknutí), zooming in/out na HTML elementy – když se uživatel netrefí (nebo ani trefit nemůže) přesně na požadovaný prvek, lze na něj lehce přejít pomocí této funkce
- automatická rotace IP adresy (tedy nedochází k blokování ze strany serveru)

Nevýhody:

- nutnost stažení aplikace (ale je zde podpora pro Windows, Linux i Mac)
- aplikace je celkově těžkopádná, nemá moc přívětivé uživatelské rozhraní, ovládání působí nepřehledně a přehlceně – na uživatele se vyvalí hodně informací a možností najednou

1.1.2 Octoparse

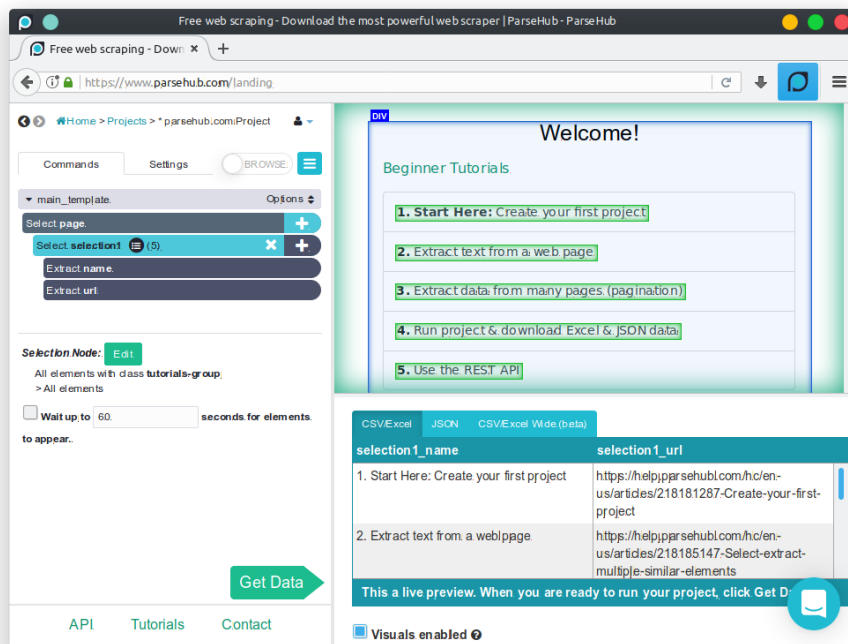
Výhody:

- výběr dat jak pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí), tak pomocí XPath nebo regulárních výrazů
- nástroj obsahuje hotové šablony, které mohou velmi urychlit práci
- pestrá paleta možností (branch judgment, tvoření smyček apod.) – dá se vytvořit téměř jakákoli logika procházení webu a extrakce dat
- lehký způsob, jak scrapování automatizovat
- možnost řídit tasky přes API (a získávat tak data taktéž přes API); data jdou nahrát rovnou i do lokální databáze

Nevýhody:

- nutnost stažení aplikace (která je navíc pouze pro Windows)
- těžkopádné a pomalé ovládání, neintuitivní rozhraní

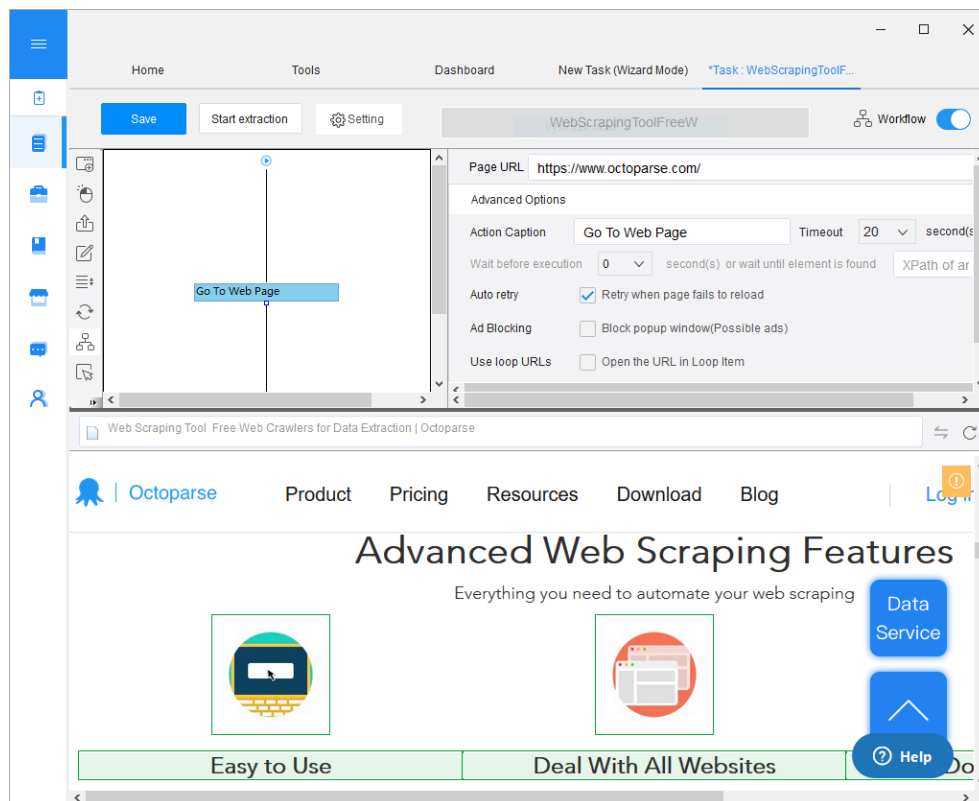
1.1. Analýza konkurenčních nástrojů



Obrázek 1.1: ParseHub[1, snímek pořídil autor]

- tutoriál je v podstatě nic neříkající
- připravených šablon je jenom pár a jsou velmi konkrétní

1. ANALÝZA



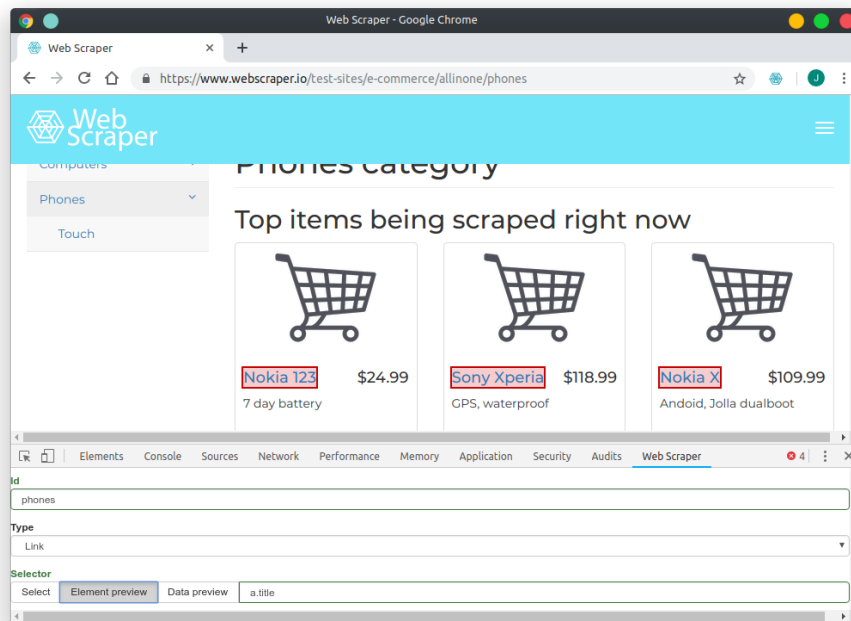
Obrázek 1.2: Octoparse[2, snímek pořídil autor]

1.1.3 WebScaper

Výhody:

- jednoduchá instalace (jedná se pouze o rozšíření do prohlížeče Google Chrome); scrapování probíhá skrze vývojářskou konzoli
- výběr dat pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí)
- tutoriály jsou formou videí – jednoduché, rychlé a naprosto postačující
- různé typy elementů, které vybíráme (text, odkaz, scroll down), takže lze celkem snadno projít celou doménu
- možnost získání dat různými formami – přes API, jako CSV/XLS nebo do Dropboxu
- klávesové zkratky při výběru elementů velmi usnadňují práci
- možnost využít jejich cloud k automatizaci celého procesu

1.1. Analýza konkurenčních nástrojů



Obrázek 1.3: WebScraper[3, snímek pořídil autor]

- oproti konkurenci nabízí přehledné rozhraní, rychlé a jednoduché používání

Nevýhody:

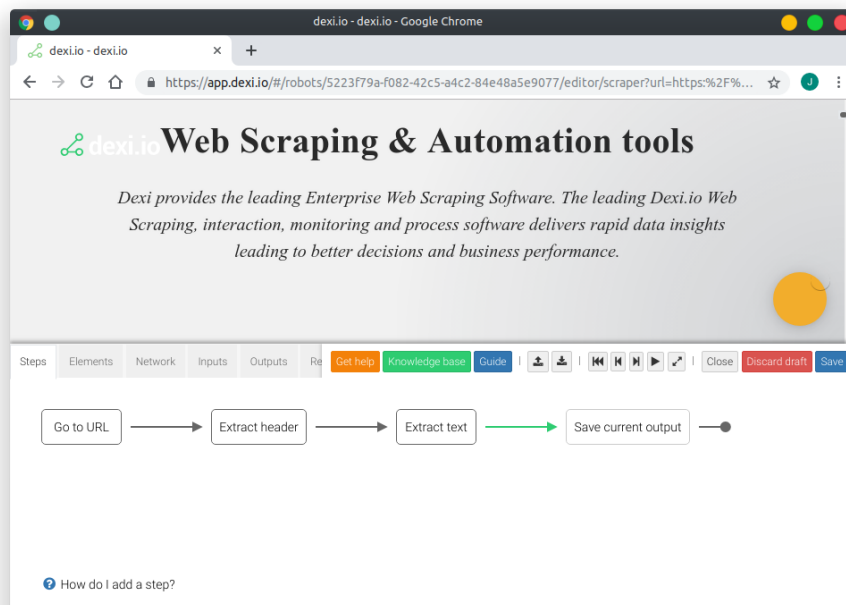
- nutnost používat Google Chrome, což pro některé uživatele může být překážka
- nelze vyhledávat podle klíčových slov ani podle HTML nebo CSS, tudíž všechno se musí manuálně naklikat

1.1.4 Dexi.io

Výhody:

- bez nutnosti stahování aplikace – vše se ovládá přes webové rozhraní
- výběr dat jak pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí), tak pomocí HTML, CSS nebo textové shody
- mnoho návodů dostupných na stránkách, interaktivní rádce přímo při scrapování

1. ANALÝZA



Obrázek 1.4: Dexi.io[4, snímek pořídil autor]

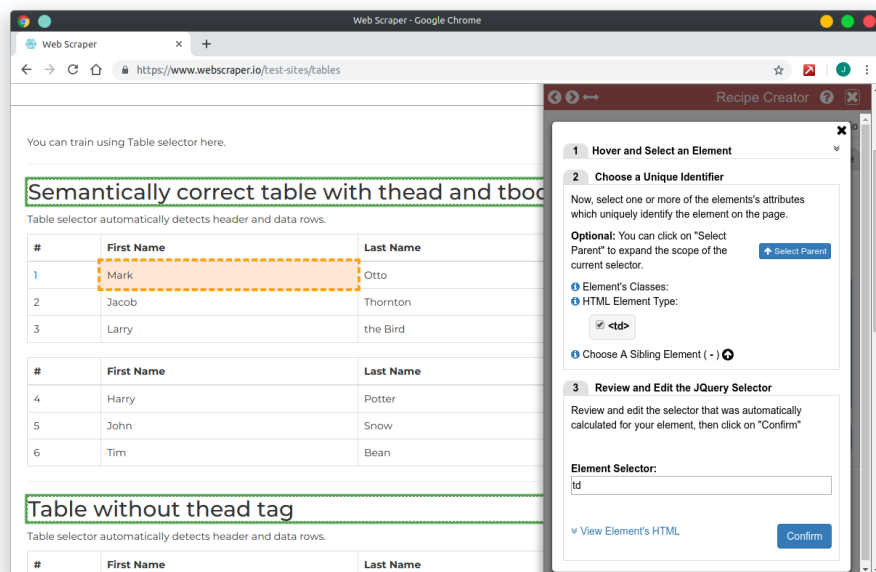
- všechny možné druhy kliknutí, takže lze lehce projít celou doménu
- možnost exportovat data do CSV, JSON, XLS, získat přes API, poslat do Google Drive, Google Sheets nebo Amazon S3
- různé módy aplikace – scraping, crawler, pipes (skládání menších scrape botů) a autobot (extrahování z více stránek najednou se stejným rozložením); možnost takto automatizovat celý proces.
- nápomocné jsou různé addony (např. na obcházení Captchy)

Nevýhody:

- široká nabídka možností, a tak chvíli trvá, než se člověk zorientuje
- placený nástroj, zadarmo je dostupná pouze týdenní zkušební verze
- úvodní tutoriál je velmi strohý a žádné velké seznámení s nástrojem se nekoná

1.1.5 Data Scraper

Výhody:



Obrázek 1.5: Data Scraper[5, snímek pořídil autor]

- jednoduchá instalace (jedná se pouze o rozšíření do prohlížeče Google Chrome).
- velmi jednoduché ovládání a přehledné rozhraní
- výběr dat probíhá pomocí klikání
- klikáním se utváří JQuery selektor, který si uživatel může podle svého upravit a doladit tak drobné detaily, jež by jinak nutně zahltily uživatelské rozhraní (tedy je možné vyhledávat i podle HTML tagů, id, CSS selektorů – zkrátka vše, co umí klasické JQuery)
- různé druhy kliknutí
- možnost spustit na stránce libovolný JavaScriptový kód v rámci scrapování

Nevýhody:

- nutnost používat Google Chrome, což pro některé uživatele může být překážka
- oproti ostatním nástrojům se může zdát velmi chudý na různé funkce

1.1.6 Shrnutí

Jak jsme viděli, největšími neduhy, které se prolínají napříč valnou většinou aplikací, jsou *těžkopádné uživatelské rozhraní*, *neintuitivní ovládání* a *rychlost* (nebo spíš pomalost), se kterou se uživatel dostane k požadovaným datům. Také jsme se přesvědčili, že nejpříjemnější cestou je celou aplikaci ovládat přes webové rozhraní *bez nutnosti stahování a instalace*.

Na druhou stranu se lze u konkurence i inspirovat. Za vyzdvižení stojí určitě *různé druhy výběru dat – klikání* přímo na stránce spolu s inteligentním hledáním podobných prvků jistě tvoří mocný mechanismus. Avšak je potřeba zajistit i ostatní způsoby výběru (jako je např. *textová shoda*, *HTML tagy*, *CSS selektory*) pro případ, kdy je pouhé klikání zdlouhavé či nevyhovující. Rovněž široký výběr způsobů exportu dat, intuitivní klávesové zkratky a zooming in/out na prvky může uživatelům zpříjemnit práci s nástrojem.

Realizace

2.1 Diskuze možných řešení

Původní záměr byl vytvořit webovou aplikaci. Důvod je jednoduchý, nikdo v dnešní době nechce cokoli stahovat a instalovat. Vzhledem k důrazu vyvíjeného nástroje kladeného na jednoduchost a rychlost používání je tak webová aplikace jasnou volbou, jelikož se jedná o nejpřímočařejší řešení a pro uživatele určitě nejpohodlnější.

Návrh tedy předpokládal jednoduchou webovou aplikaci s hlavní obrazovkou, kde by byla zobrazená uživatelem zadaná stránka a postranním panelem, který by obsahoval veškeré ovládací prvky. Jasným řešením tak byl HTML `iframe`, který reprezentuje vnořený kontext procházení (kontext procházení si můžeme představit jako jedno okno/záložku prohlížeče) – tedy umožňuje zobrazit HTML dokument uvnitř jiného HTML dokumentu.

Ač se toto zprvu zdálo jako ideální řešení, hned v úvodu jsem narazil na stěžejní implementační problém – z bezpečnostních důvodů existuje HTTP hlavička *X-Frame-Options*, která určuje, kdo může danou stránku vložit do `iframe` tagu a zobrazit ji tak v rámci své vlastní stránky (viz MDN web docs). Spousta webových stránek nastavuje tuto hlavičku tak, aby nebylo možné jejich stránky vkládat do `iframů`, čímž se brání tzv. *clickjacking* útokům (viz wikipedia). Jenže to představuje v podstatě neřešitelný problém, neboť HTTP hlavičky se v prohlížeči nedají nijak obejít a dá se předpokládat, že toto blokování bude provádět mnoho stránek.

Možnou alternativou by bylo stáhnout veškerý obsah ze zadané domény (HTML, CSS, JavaScript, všechny assety jako obrázky apod.), ten sestavit dohromady a poskytovat z vlastního serveru pouze danému uživateli. V podstatě by tak došlo k vyscrapování všech dat z cílové domény a uživatel by již pouze odfiltroval informace, o které nemá zájem. To je ale nevhodné hned z několika důvodů – jednak by byla uživatelům prezentována stránka, která ve skutečnosti není tou, za kterou se vydává; mohlo by docházet k porušení copyrightu a autorských práv a v neposlední řadě by složitost takového řešení

naprosto neodpovídala poměrně přímočarému úkolu výsledné aplikace.

Na problém se však lze dívat i opačně a základní myšlenku invertovat, což nás dovede k použitému řešení. Pokud není možné vložit stránku do webové aplikace, je nutné vložit aplikaci do požadované stránky. To lze elegantně vyřešit pomocí rozšíření do internetového prohlížeče Google Chrome – tzv. *Chrome-extension*.

2.2 Použité technologie

Celé řešení zadaného problému je implementováno jako rozšíření do internetového prohlížeče Google Chrome (jehož popisu se věnuji níže). Z tohoto důvodu je zvoleným programovacím jazykem čistý JavaScript, resp. ECMAScript 2018 verze 9. Dále je použit značkovací jazyk HTML k definici struktury celého ovládacího panelu a zbylých kontrolních prvků spolu s CSS jazykem určujícím styl zobrazení jednotlivých elementů. K testování aplikace byl použit JavaScriptový testovací framework Jest.

2.2.1 Rozšíření do prohlížeče Google Chrome

Chrome-extensions se skládají z několika různých komponent:

Background script je základní komponenta (můžeme si ji představit jako takový backend celého rozšíření), která se vykoná při každém spuštění prohlížeče. Zde je možné registrovat obsluhy různých událostí (např. když se otevře nová záložka v prohlížeči nebo když je dané rozšíření poprvé nainstalováno).

Browser action představuje tlačítko rozšíření umístěné v hlavním panelu nástrojů prohlížeče Google Chrome (vedle pole pro zadávání adresy). Po kliknutí na něj je vypuštěna událost mířící do background scriptu, kde se odehrává následné zpracování.

Content script je tou nejdůležitější částí, jenž nás bude zajímat. Jedná se o kód, který se vloží do požadované stránky a spustí se v rámci jejího kontextu. To znamená, že každý takto vložený skript má kompletní přístup k DOMu dané stránky i všem ostatním skriptům. Stane se validní součástí stránky, může vytvářet, mazat nebo měnit prvky, a sice do doby, než je stránka obnovena a znovu načtena.

2.3 Zvolené řešení

Jak již bylo řečeno v úvodu této kapitoly, použité řešení spočívá

Závěr

Literatura

- [1] PARSEHUB. *ParseHub. Version 54.0.1* [software]. 2015 [cit. 13. 4. 2019]. Dostupné z: <https://www.parsehub.com/quickstart>.
- [2] OCTOPUS DATA INC. *Octoparse. Version 7.1.2* [software]. 2018 [cit. 13. 4. 2019]. Dostupné z: <https://www.octoparse.com/download>.
- [3] WEBSCRAPER. *WebScraper. Version 0.3.8.9* [software]. 2016 [cit. 13. 4. 2019]. Dostupné z: <https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehlklipmbmhn>.
- [4] DEXI APS. *Dexi.io* [software] [cit. 13. 4. 2019]. Dostupné z: <https://app.dexi.io/>.
- [5] SOFTWARE INNOVATION LAB LLC. *Data scraper. Version 3.299.84* [software]. 2015 [cit. 13. 4. 2019]. Dostupné z: <https://chrome.google.com/webstore/detail/data-scraper-easy-web-scr/nndknepjnlbdbbepjfgmncbggmopgden>.