

Získávání dat pro realitní agregátor z veřejně dostupných zdrojů

Data acquisition for real estate aggregator from publicly available sources

Bc. Roman Tiefenbach



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Roman Tiefenbach**
Osobní číslo: **A12738**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Získ a zpracování dat pro agregátor nemovitostí
z veřejně dostupných zdrojů**

Zásady pro vypracování:

- 1. Provedte literární rešerši zaměřenou na získávání dat se zaměřením pro agregátor nemovitostí.**
- 2. Analyzujte vhodné softwarové nástroje a přístupy pro realizaci získávání dat z veřejně dostupných zdrojů.**
- 3. Realizujte systém pro získání dat z veřejně dostupných zdrojů.**
- 4. Zjistěte a zrealizujte možnosti čištění takto získaných dat.**
- 5. Provedte diskusi nad uvedeným tématem a jeho výstupy.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MLÝNKOVÁ, Irena, et al. XML technologie: Principy a aplikace v praxi. Praha: Grada, 2008. 272 s. ISBN 978-80-247-2725-7.**
2. **HEROUT, Pavel. XSLT 2.0 a SVG prakticky. 1. vyd. České Budějovice: Kopp, 2010, 293 s. ISBN 978-80-7232-406-4.**
3. **SKONNARD, Aaron a Martin GUDGIN. XML – pohotová referenční příručka: referenční příručka programátora ke XML, XPath, XSLT, XML Schema, SOAP a dalším. 1 vyd. Praha: Grada, 2006, 342 s. ISBN 80-247-0972-4.**
4. **POYNTER, Ray. The handbook of online and social media research: tools and techniques for market researchers. New York: Wiley, 2010. ISBN 978-0-470-71040-1.**
5. **SCHRENK, Michael. Webbots, spiders, and screen scrapers: a guide to developing Internet agents with PHP/CURL. San Francisco: No Starch Press, 2007. ISBN 15-932-7120-4.**

Vedoucí diplomové práce:

doc. Mgr. Roman Jašek, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

21. února 2014

Termín odevzdání diplomové práce:

20. května 2014

Ve Zlíně dne 21. února 2014



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Diplomová práce popisuje obecné možnosti získávání dat se zaměřením na realitní agregátory. Jako hlavní téma rozvádí konkrétní metodu zisku dat z veřejně dostupných zdrojů, tzv. web scraping. V teoretické části se práce snaží objasnit problematiku, nastínit využití, ale i upozornit na možné právní důsledky. Poukazuje na technologie a nástroje používané při web scrapingu. V druhé části je obsažen návrh a realizace vytvořené webové aplikace pro automatické získávání dat z více zdrojů. Součástí je i prezentace dosažených výsledků a porovnání se zástupci ve stejné oblasti. Zmíněny jsou i možné způsoby využití takto získaných dat v praxi.

Klíčová slova: web scraping, data scraping, získávání dat, extrakce dat z HTML, parsovací nástroje, technologie scrapingu, jazyk C#, ASP.NET

ABSTRACT

This master thesis describes general means of data acquisition with a focus on real estate aggregators. The main topic in first part elaborates on specific method of web scraping to obtain data from publicly available sources. The theoretical part of the thesis contains clarification of web scraping issues, outlines applications and highlights the possibility of legal consequences. It also covers the technology and tools used in web scraping. The second part of thesis contains design and implementation of web application which was created for automatic acquisition of data from multiple sources. It also includes a presentation of gathered results and compares application with representatives in the same area. The final part discusses possible uses of data obtained through web scraping in practice.

Keywords: web scraping, data scraping, data discovery, data extraction from HTML, parsing tools, scraping technology, C# language, ASP.NET

Rád bych poděkoval vedoucímu mé diplomové práce doc. Mgr. Romanu Jaškovi, Ph.D. za jeho vedení, cenné rady a konzultace, které mi ochotně poskytoval. Děkuji jednateři a kolegům z firmy, kteří umožnili vznik této práce, poskytli technické zázemí, odbornou pomoc, náměty a připomínky. Děkuji také své manželce za neutuchající podporu a svému synkovi za jeho úsměvy, které dovedou prozářit i zatažený den.

„A pessimist sees difficulty in every opportunity. An optimist sees the opportunity in every difficulty. “

Winston Churchill

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

V Olomouci 28. dubna 2014

.....

podpis diplomanta

OBSAH

ÚVOD.....	9
1.1 TYPY A FUNKCE AGREGÁTORŮ	9
1.2 ZPŮSOBY ZÍSKÁVÁNÍ DAT PRO AGREGÁTORY	10
I TEORETICKÁ ČÁST.....	12
2 CO JE „WEB SCRAPING“	13
2.1 VÝHODY.....	14
2.2 PROBLÉMY A NEVÝHODY	15
3 S ČÍM SE VYPOŘÁDAT A POTŘEBNÉ ZNALOSTI.....	16
3.1 HTML.....	16
3.2 CSS.....	18
3.3 XML A XSLT	19
3.4 JAVASCRIPT, JQUERY, AJAX.....	20
3.5 ACTIVE-X, ADOBE FLASH.....	22
4 NÁSTROJE POTŘEBNÉ PRO ZÍSKÁNÍ DAT	24
4.1 PHP.....	24
4.2 PERL.....	25
4.3 RUBY.....	25
4.4 PYTHON.....	25
4.5 .NET	26
4.6 DALŠÍ MOŽNOSTI.....	26
5 PRÁVNÍ STRÁNKA VĚCI.....	27
5.1 AUTORSKÉ PRÁVO	27
5.2 OCHRANA OSOBNÍCH ÚDAJŮ	27
5.3 INFORMAČNÍ A INFORMATICKÉ ZLOČINY	28
5.4 DOPORUČENÍ.....	28
II PRAKTICKÁ ČÁST	30
6 NÁVRH APLIKACE	31
6.1 OBECNÉ CÍLE APLIKACE	31
6.2 MOŽNÉ PROBLÉMY	31
7 VYBRANÉ SOFTWARE NÁSTROJE.....	34
8 ARCHITEKTURA.....	35
8.1 DB PROJEKT	35
8.2 WEB SCRAPING KOMPONENTA.....	36
8.3 PREZENTAČNÍ ČÁST	36
8.3.1 Veřejný přístup.....	36
8.3.2 Administrativní přístup	38
9 ŠABLONY.....	42
9.1 OBJEKT TYPU „ITEM“	42
9.2 OBJEKT TYPU „PATH“	42
10 FILTROVÁNÍ DAT	44

10.1	DUPPLICITY „PRVNÍHO“ DRUHU	44
10.2	DUPPLICITY „DRUHÉHO“ DRUHU	45
10.3	ŘEŠENÁ FILTRACE V APLIKACI	45
11	VYUŽITÍ DAT	46
12	VÝKON	47
12.1	KOLEKCE.....	47
12.2	VLÁKNA	48
12.3	DATABÁZE	48
12.4	ZPŮSOB ZÍSKÁVÁNÍ DAT	49
13	MOŽNÁ DALŠÍ VYLEPŠENÍ.....	50
13.1	UŽIVATELSKÁ STRÁNKA.....	50
13.2	TECHNICKÁ STRÁNKA	50
14	POROVNÁNÍ S AKTUÁLNÍMI ŘEŠENÍMI	52
14.1	EXITUJÍCÍ REALITNÍ SERVERY NA BÁZI WEB SCRAPINGU	52
14.2	OBECNÉ WEB SCRAPERY	52
14.2.1	WebHarvy	53
14.2.2	Helium.....	55
14.2.3	Rozdíly	56
ZÁVĚR	58	
ZÁVĚR V ANGLIČTINĚ.....	60	
SEZNAM POUŽITÉ LITERATURY.....	62	
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	65	
SEZNAM OBRÁZKŮ	66	
SEZNAM TABULEK.....	67	
SEZNAM PŘÍLOH.....	68	

ÚVOD

Podle některých názorů lze na internetu nalézt úplně vše. Je to trochu nadnesené, ale pravda je, že tam lze najít opravdu hodně. Velký problém je ono hledání. Aby byla šance vůbec něco na internetu najít, existují vyhledávače.

Vyhledávače indexují webové stránky, podle složitých algoritmů. Člení je do skupin, podle relevance, důvěryhodnosti, počtu odkazů a spousty dalších kritérií. Vesměs jsou to fulltextové vyhledávače.

Vyhledávače se uplatňují samozřejmě i na konkrétních stránkách, resp. webech či serverech. Tam je situace, většinou, trochu snazší, jelikož známe strukturu dat, nad kterými vyhledáváme, což obecně neplatí. Pak lze nabídnout i relevantnější výstupy. Tyto vyhledávače, kromě fulltextové části většinou využívají i nějaké atributové členění. Vždy záleží na konkrétních stránkách, či serverech.

Pokud něco hledáme, začínáme s dotazy nad obecnými vyhledávači typu seznam.cz, google.com. Vracené relevantní výsledky jsou konkrétní podle fulltextového hledání, tj. obsahují co nejvíce hledaných slov. Tyto výsledky se zpravidla nacházejí na určitých typech serverů, které se zabývají danou problematikou. Jedná se o servery, které shromažďují (agregují) informace stejného typu nebo zaměření z různých zdrojů. Přitom lze odkazovat na další servery, nebo nabízet konečný obsah.

1.1 Typy a funkce agregátorů

Agregátory mohou fungovat na automatické nebo manuální bázi. Agregátory na manuální bázi dělají „víc“ z dalších zdrojů, ale pod dohledem nějakého správce, který vybírá, co se zobrazí a co už nikoliv. Ať už se jedná o filtrování nasbíraných informací, nebo sběr těchto informací správcem a jejich následné zadání do systému.

Na automatické bázi, můžeme říct, že jeden systém/server získává data od dalších systémů/serverů podle aktuálního nastavení, bez zásahu správce.

Mezi nejznámější typy agregátorů, tak jak jsou chápány širokou veřejností, určitě patří:

- **RSS čtečky** jsou agregátory všemožných krátkých zpráv podle toho, které jsme si přidali.
- **Agregátory zpráv** – zobrazuje zprávy z různých zdrojů (např. <http://news.google.cz>, <http://www.seznamzprav.cz>, <http://pravednes.cz>)

- **Agregátory slev** – sbírá slevy napříč republikou (např. <http://www.slevomat.cz>, <http://www.raketa.cz>, <http://www.bozskeslevy.cz/>, <http://www.mixo.cz>, <http://skrz.cz/>)
- **Agregátory zboží** – sleduje konkrétní typy zboží a jejich ceny (<http://www.heureka.cz>, <http://www.zbozi.cz/>)
- **Agregátory půjček** – zobrazuje všechny možné, převážně nebankovní, půjčky (<http://agregator-pujcek.cz/>)
- **Agregátory nemovitostí** – nabízejí velké množství nemovitostí od různých kanceláří na jednom místě (<http://sreality.cz/>, <http://realcity.cz/>, <http://www.viareality.cz/>)

Slovo agregátor je od slova agregovat, neboli sdružovat. Pokud budeme uživatele popř. lidi považovat za různé zdroje informací, přeneseně pak můžeme agregátorem označit téměř vše, kudy se během brouzdání internetem pohybujeme. Webové servery zpravodajských služeb (každý reportér je určitě jiný zdroj informací), servery zabývající se poskytováním služeb kolem médií (flicker, picasaweb, youtube), různé blogy, servery typu wikipedia, či sociální sítě (facebook, google+, twitter) pak také patří do agregátorů těchto informací.

1.2 Způsoby získávání dat pro agregátory

Pro „agregátory“ jsou nejdůležitější data a uživatelé. Pokud jsou data aktuální, přesná, konkrétní, obsáhlá a relevantní své uživatele si určitě najdou. Samozřejmě zde hrají svou roli i další aspekty, jako uživatelské prostředí, funkčnost, povědomí veřejnosti a na neposledním místě konkurence. Lidé obecně nejsou příliš ochotni měnit návyky, místa a způsob uvažování, pokud k tomu nemají pádné důvody.

Pomineme-li ruční procházení webů a vybírání jednotlivých informací manuálně správcem, pak můžeme data získat například pomocí:

- **Ruční zadávání od uživatelů** – nepatří až tak do automatického nebo do ryze „agregátorského“ získávání dat, ale představuje také určitý podíl na datech. Data jsou vkládána strukturovaně, zpravidla pomocí webových formulářů
- **Sdílení databází** – většinou není možné, protože se jedná o konkurenční firmy nebo prostředí.

- **RSS feedy** – neobsahují plnohodnotné informace (hodí se třeba pro zprávy), ale jen nadpisy, či části textu. Převážně slouží k dalšímu odkazování na zdroj těchto informací.
- **Napojení na API služby** – lze využívat existujícího komunikačního rozhraní konkrétního serveru nebo systému. API většinou není univerzální, ale pokud je na trhu důležitý „hráč“ v určitém okruhu zdrojů, jeho API se může stát neoficiálním „standardem“ a ostatní je též implementují. Většinou potřebuje nějakou formu autorizace, takže není veřejně dostupná.
- **Využití XML sestav** – po domluvě může existovat i výstup do tzv. „sestav“. Ty jsou zpravidla tvořeny 1x denně v kompletní formě a každou hodinu ve změnové formě. Jedná se o soubory XML, obsahující výstupní data v předem domluvené podobě, tak aby se dala dále strojově zpracovávat. Je zde stejný problém jako u API, kdy jednotlivé výstupy jsou různorodé a většinou chráněné nějakou formou autorizace.
- **„Scraping dat“** – vytahování dat přímo ze stránek zdrojového serveru. Cokoli je možné pro uživatele zobrazit, to je možné přečíst a zpracovat i strojově. A pokud jsou tato data dostupná veřejně, ani stroj nepotřebuje žádnou autorizaci.

V dalším textu se zaměřím na strojové získávání a zpracování dat posledním popsáním způsobem – tzv. „scraping dat“.

I. TEORETICKÁ ČÁST

2 CO JE „WEB SCRAPING“

Internet je úložiště dat v podobě textu, medií, nebo jakémkoliv jiném formátu. Každá webová stránka zobrazuje data v nějakém formátu. Přitom tato data mohou být zásadní pro úspěch společnosti v novodobém světě. Naneštěstí jen málo webových stránek umožňuje získání těchto dat.

„Web Scraping (also termed Screen Scraping, Web Data Extraction, Web Harvesting etc) is a technique employed to extract large amounts of data from websites.” [28]

Což volně přeloženo znamená: „Web scraping (taktéž označován jako Screen Scraping, Web Data Extraction, Web Harvesting, atd.) je technika, která vytahuje velký počet dat z webových stránek.“. Zjednodušeně se jedná o program, který vytahuje konkrétní informace z dat zaslaných jiným programem, která jsou primárně určená pro čtení lidským uživatelem.

Web Scraping je technika automatizování procesu získávání rozsáhlých dat z webových serverů, týkající se např. telefonního seznamu, seznamu nemovitostí, sociální sítě nebo různých elektronických obchodů, které by v případě potřeby bylo nutné kopírovat zdlouhavě ručně. Programy zabývající se web scrapingem toto zvládnou za zlomek času. Chovají se stejně jako webový prohlížeč, ale místo zobrazení stránky, uloží požadovaná data do souboru nebo databáze.

Získání dat z různých zdrojů se může například uplatnit pro:

- Analýzu nebo průzkum trhu – oblasti zájmu mohou být servery s nabídkou nemovitostí, bazary aut, stránky s elektronickými zařízeními. Získaná data mohou sloužit pro porovnání cen, nabídek nebo značek. Lze pak jednoduše zjišťovat trendy pro jednotlivá odvětví podnikání, mít přehled co je zrovna v popředí zájmu a co už končí.
- Výzkum – data jsou nedílnou součástí jakéhokoliv výzkumu, ať už se jedná o akademický, marketingový nebo třeba vědecký.
- Marketingové účely – je jednoduché si udělat databázi firem a společností spolu s jejich jmény adresami nebo telefonními čísly. Všechny tyto informace jsou veřejně dostupné.
- Web „crawling“ – je termín označující práci programů, když surfují na internetu. Zkouší hledat cokoli, co mají v popisu práce. Tyto programy jsou nazývány „ants, bots, spiders“. Jsou hlavně doménou vyhledávacích strojů tzv. „search engines“,

kteří si takto udržují přehled o aktuálních a nových stránkách. Odtud si oindexují obsah pro vyhledávání. Taktéž lze tyto programy využít pro údržbu stránek, kde dochází ke kontrole hypertextových odkazů, kontrole HTML kódu, případně chyb pro SEO.

- Vlastní RSS čtečku – „RSS feedy“ dnes nabízí většina stránek s často se měnícím (dynamickým) obsahem. Pokud ne, je možné si je vytvořit.
- „Meshup“ aplikace – jedná se o aplikace, které využívají kombinaci více zdrojů a funkčnosti pro zobrazení žádané informace. Jako příklad může být uvedeno spojení seznamu restaurací se zobrazením na mapě, kde lze zobrazit a vkládat komentáře od návštěvníků.

2.1 Výhody

Jednou z největších výhod web scrapingu je, že cokoliv, co může být zobrazeno uživateli na webové stránce, může být staženo. Každá HTML stránka obsahuje strukturovaná data v podobě HTML značek. Jako další nesporná výhoda je fakt, že vlastníci webových stránek daleko více dbají na údržbu viditelných stránek, které lze navštívit, než nějaký strukturovaný výstup (třeba RSS).

Důvod je většinou prostý. API nebo nějaký výstup se strukturovanými daty se napíše jednou, a pokud na něm nejsou závislé¹ hromady lidí a dalších programů, má tendenci zapadnout a zastarat. Ale pokud se začne webová prezentace zobrazovat nekorektně, je to více o „všechno ostatní zastavte a rychle to opravte“ problému.

Pro veřejné stránky neexistuje nic jako „Rate-Limiting“ (počet přístupů, nebo časová omezení). Přesto že jsou některé stránky, které jsou chráněny „Captcha“² kódem nebo přihlášením, většinou zde není mnoho zábran proti automatickému zpracování. A pokud nepoužíváte paralelní přístup, budete v logu působit jako aktivní uživatel, v případě, že by se někdo díval.

¹ Někdy i přesto – viz API Twitteru nebo Facebooku o jejichž problémech se pravidelně psává – je pravda, že vývojáři velkých firem mají tendence oznamovat změny s předstihem aspoň jednoho měsíce.

² Anglická zkratka pro Completely Automated Public Turing test to tell Computers and Humans Apart – jde o obrázky na kterých je text, většinou hodně špatně čitelný, který musí uživatel zadat, aby prokázal, že se jedná o člověka a byl v prohlížení puštěn dál.

Přístupem přes „HTTP requests“ se posílá jen IP adresa a „cookies“, což zaručuje určitou anonymitu oproti nutnosti si zřídit účet k API a posílat přihlašovací údaje při každém dotazu. O tom, že tyto elementy lze v případě potřeby měnit se ani nezmiňuji.

2.2 Problémy a nevýhody

Ve většině případů je data scraping vnímán jako účelová nebo provizorní a nepříliš elegantní technika používaná jako poslední „záchrana“ v případě, že není žádný jiný mechanismus pro komunikaci, či získání dat.

Nečastější důvod použití web scrapingu je přístup a získání dat s cizích webových stránek, které neposkytují k použití výhodnější API. V tomto případě se cizí servery dívají na web scraping jako nežádoucí. A to nejen z důvodu vyššího zatížení serveru a datového připojení, ale i z důvodu případných ušlých zisků z reklamy nebo ztráty kontroly nad šířícími se daty.

Mezi nevýhody této techniky zpracování dat patří vyšší programovací náročnost, vyšší náročnost na vlastní zpracování, kde se musí zpracovat spousta jiných dat, které nejsou třeba, a v neposlední řadě problémy se změnami vizuální podoby stahovaných stránek. Člověk si s takovou změnou velmi rychle poradí, ale program zpravidla přestane produkovat požadované a správné výsledky a v horším případě přestane fungovat úplně.

Normální přenos dat mezi programy zajišťují speciální formáty a protokoly obvykle pevně dané struktury, které jsou dobře zdokumentované, nechají se jednoduše parsovat s minimální nejednoznačností. Často takto přenášená data nejsou pro člověka čitelná.

Ale data pro programy zabývající se web scrapingem jsou primárně určena pro lidského uživatele, tudíž neexistuje žádná dokumentace nebo pevná struktura pro jednoduché a jednoznačné parsování. Velká část takto získaných dat se zahazuje³. Nicméně část z těchto zahazovaných dat lze dobře využít pro orientaci a navigaci. A tady může nastat problém i při nepatrné změně.

³ Veškeré údaje pro formátování vzhledu dat, skryté komentáře, binární a další nepotřebná data

3 S ČÍM SE VYPOŘÁDAT A POTŘEBNÉ ZNALOSTI

Na internetových stránkách se setkáme s různými prezentacemi dat. Od jednoduchých stránek představujících třeba práci a kontakt řemeslníka. Tato stránka bude stručná, minimalistická, jen se základními údaji.

Následují střední stránky s pokročilejším menu, více druhů stránek od kolekcí obrázků, příspěvků, vlastní tvorby, přes životopisné pojednání. Sem patří představení menších firem nepřímo obchodující na internetu.

A můžeme skončit třeba u zpravodajských serverů, které nás denně zásobují velkým množstvím článků a zpráv všeho druhu. Ty většinou běží na komplexních redakčních systémech umožňující snadné rozčleňování do rubrik, schvalování a kontrolu.

Pro každou prezentaci může být použito několik technologií jako HTML, XHTML, CSS XML+XSLT, javascript, ajax, jquery, active-X, flash, stejně jako programovacích jazyků od čistého HTML, javascript, přes PHP, po Ruby, Python nebo ASP.NET.

Pro web scraping je důležité členění s hlediska zpracování skriptů na straně klienta a na straně serveru. Pokud jsou data zpracována na straně serveru, je výsledek poslán klientovi jako hotový v podobě HTML, zatímco zpracování na straně klienta znamená zapojení klientské strany, která teprve poslaný kód převede na klasické HTML.

Samozřejmě v dnešní době se nepoužívá čistě jen jednoho přístupu, ale jejich kombinací. To znamená, že převážná část se již pošle hotová ze serveru a na místě u klienta se dotáhnou některá specifická data.

Programovací jazyky na straně serveru mají jednotný výstup – HTML stránku obalenou skripty a CSS pro zpracování na straně klienta. Proto se v dalším textu zaměříme na technologie používané při zpracování právě na straně klienta. Jedná se o výsledek, který vidí uživatel a který web scrapingový software dokáže využít.

3.1 HTML

Značkový jazyk HTML (HyperText Markup Language) je hlavním jazykem pro tvorbu webových stránek na internetu, zejména v systému World Wide Web. Pochází z obecnějšího SGML (Standard Generalized Markup Language), ale byl ovlivňován vývojem webových prohlížečů, které jej zpětně definovaly.

V roce 1990 byl navržen jazyk HTML spolu s protokolem HTTP, ke kterým následně Tim Berners-Lee přidal i první webový prohlížeč. Po svém rychlém boomu bylo pro HTML třeba stanovit standardy. Začalo se ve verzi 0,9 v roce 1991, kdy v rychlém sledu následovaly novější verze až do konce roku 1999, kdy se končí na 4,01 a vývoj byl nadále pozastaven ve prospěch XHTML. V současné době se využívá zatím neoficiální (nestandardizovaná) verze HTML 5, zatím ohlášená na konec roku 2014.

Koncepce jazyka HTML je dána skupinou speciálních značek, tzv. tagů. Ty popisují strukturu dokumentu. Od popisu hlavičky (<head></head>), která definuje ikony, jazykovou sadu, autora, klíčová slova, popis stránky a další, přes popis vlastního těla (<body></body>), které tvoří obsah dokumentu, což může být téměř cokoliv od textu, odkazů, obrázků, zvuků či videa až po interaktivní video tvořené třeba ve flashi. Celá tato základní struktura je obalena opět párovou značkou dokumentu (<html></html>)

```

</tr>
<tr>
    <td><b>Druh objektu:</b></td>
    <td>panelov</td>
</tr>
<tr>
    <td><b>Stav objektu:</b></td>
    <td>po rekc</td>
</tr>
<tr>
    <td><b>Vlastnictví:</b></td>
    <td>osobní</td>

```

Obr. 1: Úryvek zdrojového kódu HTML stránky

- člověkem lépe čitelný

Jazyk se skládá z párových tagů, či nepárových, které by měly být uzavřeny speciálním znakem (např.
). Správnost a validnost kódu takto psaných stránek se může testovat přímo na stránkách W3C konsorcia. Nicméně webové prohlížeče podporují i ne zcela validní kód a umějí si poradit i s chybějícími párovými značkami, či překřížením pořadí tagů (<i>Ahoj</i>).

Každý z tagů může obsahovat určité množství atributů, které jej upravují, či rozšiřují. Jedná se zejména o stylování, pojmenování, případně dodatečný výpis informací.

HTML dokument je strukturován vnořováním jednotlivých elementů, přičemž některé slouží jako druh kontejneru pro zobrazení (<div> nebo), jiné jako viditelný konkrétní výstup (<input>).

```

alková cena:</strong> </span> <span class="price"> 15 000 Kč <span cl
<span">+ služby 450,-/os. + elektřina</span> <span class="clear"></span>
vé Město </span> <span class="clear"></span> </p> <p class="row"> <
strong>ID zakázky:</strong></span> <span class="id">56745</span> <sp
class="clear"></span> </p> <p class="row"> <span class="desc"><stro
n>Dobrý</span> <span class="clear"></span> </p> <p class="row"> <sp
u:</strong></span> <span>Centrum obce</span> <span class="clear"></
desc"><strong>Podlaží umístění:</strong></span> <span>3. podlaží</s
</span> </p> <p class="row"> <span class="desc"><strong>Plocha podla
ss="value">MHD, Autobus</span> <span class="clear"></span> </p> <sc
ng>Popis:</strong></span> <p class="description">Pronájem bytu 3+1,
t se pronajímá nezařízený. K dispozici plně zařízená kuchyňská link
orná dopravní dostupnost: metro, tram, bus cca tři minuty chůze. V b
</div> </div> <div class="clear"> <hr /> </div> <div id="re
f="#realityOptions" id="share_twitter" class="share" title="Twitter
are" title="E-mail"></a> <a href="#realityOptions" id="qrCodeLink"
a href="#realityOptions" class="rightLink" id="sendReportLink">Nahl
s" id="sendEmailForm" class="sendEmailForm"> <fieldset> <h4>Poslat
id="myMessage" class="item" rows="5" cols="40"></textarea> <div id=

```

Obr. 2: Úryvek HTML kódu – méně čitelný (pokud by bylo vypnuté zalamování řádků, je pouze na jednom)

Pro web scraping jsou důležité zejména tzv. „CSS háky“ jako název tagu, jméno třídy, či ID elementu.

3.2 CSS

Jazyk pro kaskádové styly (Cascade Style Sheet) popisují vzhled webového dokumentu, jako je velikost, barva a typ písma, styl odrážek, mezery, či odsazení jednotlivých elementů. K tomu využívá v základu „CSS háky“ jako ID, třídu anebo tag konkrétního elementu. Byl navržen standardizační organizací W3C.

Tyto styly se zpravidla do webového dokumentu připojují v podobě odkazu na soubor se styly. Taktéž mohou být vepsána „hromadně“ přímo do HTML dokumentu, stejně jako se nechají stylovat jen konkrétní elementy.

Kaskádové jsou proto, že se aplikují kaskádovitě. Pokud nastavím třeba barvu pozadí nějakému elementu, všechny elementy uvnitř tohoto tagu budou mít stejné pozadí, pokud ji nepřekryjí u konkrétního elementu, pak se použije ta a tak dál. Čím konkrétnější určení cílového elementu v CSS, tím větší váha u překrývání, ale menší dosah v rámci elementů. Největší váhu má atribut <style> přímo v elementu, ale nejmenší dosah – jen tento element.

Tento proces formátování dat (HTML, XML, ...) probíhá až na straně klienta ve webovém prohlížeči. Stává se, že v různých prohlížečích jsou stejně formátovaná data zobrazena různě. Ale existují postupy, jak toto chování sjednotit.

Z pohledu web scrapingu je CSS nepodstatné, přestože využívá jeho záchytných háků pro navigaci v samotném HTML dokumentu.

3.3 XML a XSLT

Jedná se o obecný značkovací jazyk XML (Extensible Markup Language), zjednodušení staršího jazyku SGML. Používá se na serializaci dat, kde si konkuruje například s „JSON“. Zatímco HTML je strukturovaný kontejner na data, nepopisující ani tak data, jako strukturu pro jejich zobrazení, XML je přímo „obal“ popisující strukturu dat. Umožňuje snadné vytváření popisné formy dat pro různé účely a typy.

```
▼<breakfast_menu>
  ▼<food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    ▼<description>
      Two of our famous Belgian Waffles with plenty of real maple
    </description>
    <calories>650</calories>
  </food>
  ▼<food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    ▼<description>
      Light Belgian waffles covered with strawberries and whipped
    </description>
    <calories>900</calories>
  </food>
  ▼<food>
    <name>Berry-Berry Belgian Waffles</name>
```

Obr. 3: Ukázka zdrojového kódu XML

Jelikož XML nemá definované tagy, nebo pojmenované atributy, je třeba toto učinit, tedy v případech, kdy je to potřeba. Pokud takto zaznamenaná data slouží širokému užití, je třeba mít dokumentovaný význam jednotlivých tagů a atributů v tzv. DTD (Document Type Definition), pokud je tento formát použit pro dočasný přenos v aplikacích psaných jedním vývojářem, pravděpodobně se psáním DTD dokumentů obtěžovat nebude. Definičních jazyků je vícero druhů. Také se jím říká XML Schema a jako další zástupce může být XSD. Tyto a další schémata vznikají třeba kvůli schopnosti popisovat datové typy, nebo kvůli složitosti zápisu atd. Díky takto definovaným strukturám dat je možné provádět automatickou kontrolu dat.

Nejnámější využití XML jsou asi RSS zprávy, SOAP (protokol pro komunikaci mezi webovými službami), popřípadě Open Office, nebo další kancelářské aplikace, byť

navenek využívající komprimovaný formát zip. Aktuální verze XML je 1.1 a od 1.0 se liší možnostmi psát jména tagů a atributů v jazycích, které v době vzniku 1.0 nebyly začleněny v unicode sadách písma.

Rodina jazyků XSL (Extensible Stylesheet Language) je určena především k transformaci a úpravě dat popsaných v XML. Je možné data upravovat po vizuální stránce (XSL-Format Object), transformovat do jiného dokumentu (XSLT), jako třeba html, vybírat části dat nebo tvořit osnovu, či obsah k takovým datům. Další z rodiny je dotazovací jazyk XPath, který je sem přidružen přesto, že s XML nemanipuluje, ale používá se v XSLT pro popis XML dat, která se mají zpracovat.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Title</th>
      <th>Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
```

Obr. 4: Ukázka zdrojového kódu pro XSLT

Webové stránky se v čistém XML nevyskytují, přestože by jejich využití bylo pro web scraping mnohem jednodušší. Zpracované XML přes XSLT technologii, pokud je určeno pro web zpravidla končí jako HTML, kde už ztrácí svoji čistotu popisující jen data.

3.4 JavaScript, jQuery, AJAX

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, zejména používaný jako interpretovaný pro tvorbu webových stránek na straně klienta. Autorem je Brendan Eich a vznikl v roce 1995. Jsou jím obvykle ovládány interaktivní prvky GUI (Graphic User Interface – grafické uživatelské rozhraní), například tlačítka, textová pole a jiné.

Dnes je možné použít JavaScript i na straně serveru v podobě open source implementace Rhinola nebo Node.js, ačkoliv první servrová implementace, známá jako LiveWire vznikla již v roce 1996.

Jedná se o slabě typovaný jazyk a nedochází k typové kontrole při překladu, tudíž je možné do proměnné, kde je uložena hodnota celočíselného výrazu, následně pak přiřadit hodnotu řetězce.

```
<script type="text/javascript">
  <!--//--><![CDATA[//><!--
  // (C)2000-2013 Gemius SA - gemiusAudience / centrumcz / xy:
  var pp_gemius_identifier = "nSqV9rOKJZiWgLLF_ZUQ37btDquwlgM
  function gemius_pending(i) { window[i] = window[i] || functi
  gemius_pending('gemius_hit'); gemius_pending('gemius_event')
  (function(d,t) { try { var gt=d.createElement(t),s=d.getEler
    gt.src=document.location.protocol+'//spir.hit.gemius
  //--><![]]>
</script>
```

Obr. 5: Ukázka zdrojového kódu javascriptu

Určitou nadstavbou, nebo JavaScriptí knihovnou je jQuery. Jedná se o open source software, který je velice oblíbený a hojně využíván. Jeho filozofií, stejně jako v případě CSS a HTML, kde CSS odděluje stylování prvků, je oddělení „chování“ od struktury HTML.

Jeho síla spočívá v manipulaci s DOM (Document Object Model) strukturou, vyhledávání prvků s podporou XPath, navazování a odchytávání událostí, manipulace s CSS, animace a efekty, utility doplňující či zjednodušující funkčnost JavaScriptu (třeba funkce each nebo val) a v neposlední řadě využití technologie AJAX.

AJAX (Asynchronous JavaScript and XML) je označením souboru technologií pro vývoj interaktivních webových aplikací, které umí měnit částečný obsah stránek bez nutnosti jejich opětovného kompletního načtení. Využívá asynchronního volání serveru, který žádá o data, která následně zpracuje a zobrazí

Do AJAX technologií řadíme HTML (XHTML) s CSS pro prezentaci, JavaScript pro manipulaci s DOM a dynamické zobrazování změn a XMLHttpRequest, který za pomoci asynchronní výměny dat s webovým serverem, zpravidla za využití XML, ale může i další (HTML, JSON), obdrží nová data aniž by se musela překreslovat celá stránka.

V současnosti je AJAX podporován všemi moderními prohlížeči. Možností nesestavovat pokaždé celou HTML stránku při každém požadavku na server se snižuje jeho zátěž i zátěž přenosových linek. Uživatel má příjemnější pocit z interakce, třeba při hlasování v anketě, kdy se mu po hlasování překreslí pouze oblast ankety s jeho novým hlasem, a nikoliv celá stránka, a pokud měl „odscrollováno“ do půli dlouhé stránky nedochází k efektu skoku na začátek.

Obráceně může docházet k určité latenci při reakci na požadavek. Pokud stránky nepočítají i s drobným zdržením při získávání odpovědi, může se stát, že si třeba lístek do kina zarezervujete 10x. Stránky by v takovém případě měli znemožnit opakovat stejnou akci a dát uživateli patřičně najevo, že se jejich požadavek vyřizuje.

Technologie popsané v této kapitole jsou pro web scraping asi nejproblematictější. Pomineme-li různá zabezpečení přes přihlašování a ověřování CAPTCHA kódů, samozřejmě. Na druhou stranu využití AJAX služeb, které komunikují se serverem za využití XMLHttpRequestu s nějakými parametry, lze tyto využít jako nedokumentovaný zdroj dat (nedokumentované API).

3.5 Active-X, Adobe Flash

Active-X je technologie společnosti Microsoft originálně pro sdílení informací mezi aplikacemi typu MS Office a IE na počítačích se systémem Windows. Myšlenka byla znovu využít již napsaného kódu napříč aplikacemi - třeba kontrola pravopisu bude stejná ve Wordu, PowerPointu nebo Outlooku. Přestože v současné době jsou tyto části kódu digitálně podepisovány kvůli důvěryhodnosti a je ponecháno na uživateli, zda danému certifikátu věří či nikoliv, byla reputace Active-X pošramocena z důvodu možného zneužití. Tyto části kódu, který spustí prohlížeč, mají přístup k hostitelskému počítači, a to včetně práva zapisovat a číst z disku. Na internetu se obecně nedoporučuje využívání Active-X technologie.

Za pomoci technologie Active-X se vyvinula spousta produktů a platforem jako ASP (Active Server Page), ActiveMovie nyní známé jako DirectShow, nebo ASF (původně ActiveX Streaming Format, pak Advanced Streaming Format a naposledy Advanced Systems Format).

Adobe Flash se používá pro tvorbu převážně internetových interaktivních animací, prezentací a her založené na vektorové grafice. Největší oblibu si získal v podobě bannerů, které vytlačily dřívější formát gif, stejně jako pro přidání streamovaných videí, či audia na webové stránky. Výsledný soubor je relativně malý, díky zachování vektorové grafiky, a spustitelný v Adobe Flash Playeru. Je schopen reagovat na uživatelské vstupy skrz myš, klávesnici, či mikrofon a web kameru.

V současné době začíná být flash pomalu vytlačován HTML5. Ať už flash nebo Active-X jsou technologie nepříliš vhodné pro získávání dat skrze web scraping. Obě technologie používají kompilované výstupy, které se nedají normálně číst.

4 NÁSTROJE POTŘEBNÉ PRO ZÍSKÁNÍ DAT

Nástrojů pro získávání dat metodou web scrapingu je celá řada. Hodně záleží na preferencích, programovacích zkušenostech a oblasti použití.

Jedním ze základních přístupů je metoda "copy and paste". Sice základní, ale na dobře chráněných stránkách proti automatizovanému procházení jediná spolehlivě fungující možnost.

Na internetu lze nalézt poměrně velkou základnu aplikací zaměřujících se na web scraping, které se liší cenou, robustností, způsobu ukládání dat nebo zadávání vzoru, podle kterého se stahují informace. Další kritérium může být, zda se jedná o desktopovou aplikaci, či řešení fungující v prohlížeči.

Většina podporuje nějaký způsob zadání hledacího vzoru, kde si vystačíte s myší, nebo například se základy XPath.

Nebo lze vyrazit vlastní cestou, nechat se inspirovat na existujících projektech, a napsat si vlastní scraper v nějakém programovacím jazyce. Protože obecné scrapery jsou sice fajn, ale pokud potřebujete trochu netypické zadání, většinou už si neporadí. A popravdě pokud si napíšete nástroj, aby dělal jednu konkrétní věc na předem určené a definované množině, zpravidla funguje lépe než obecná řešení, která musí řešit spoustu kompromisů.

Výběr programového prostředí je otázka preference. Pokud se vyhnete potřebě sbírat data, která se mění až na straně klienta, vystačíte si s dobrou knihovnou na parsování HTML, stejně jako knihovnou pro práci s HTTP. V opačném případě je potřeba využít zobrazovací jádro nějakého prohlížeče (browser engine) pro vytažení takovýchto dat. Tento proces je výrazně pomalejší než sparsovat text. Níže uvádím některé oblíbené jazyky a technologie pro web scraping na které jsem narazil při získávání informací.

4.1 PHP

Lidé preferující jazyk PHP se zmiňují o SimpleHTMLDom, Shrubber/Curl a AWK. SimpleHTMLDom [15] je parser napsaný v PHP5+, který slouží k manipulaci s HTML. Poradí si s invalidním HTML a lze nad ním vyhledávat jako nad jQuery. Disponuje jednoduchým hledáním a získáváním obsahu.

Curl (Client URL) [16] umožňuje připojení a komunikaci s mnoha typy serverů přes variaci protokolů (http, https, ftp, gopher, telnet, dict, file, and ldap). Wrapper od pana Shrubera zjednodušuje dotazy nad touto technologií.

AWK [17] je interpretovaný programovací jazyk zaměřený na zpracování textu, typicky pak na jeho extrakci a analýzu. Jedná se o výbavu většiny stávajících distribucí postavených na Unixu. Jazyk Perl se jím nechal inspirovat.

4.2 Perl

Knihovna Mechanize [21] je velice mocný nástroj pro extrakci a parsování stránek. Je určena pro automatizaci interakce s webovými stránkami. Umí pracovat a posílat soubory cookie, vyplňovat získané formuláře a následně je odeslat, automaticky sledovat linky ze stránek. Ukládá historii navštívených stránek pro pozdější práci nad nimi.

Její derivace od různých autorů existují pro Perl, Python, Ruby, a možná pár dalších programovacích jazyků. Existují další doplnění a modifikace rozšiřující funkčnost.

LWP (Library WWW for Perl) [18] je knihovna modulů pro práci s webovými stránkami, některé moduly a derivace Mechanize byly odsud převzaty. Stejně jako u předchozího Mechanize dostanete nástroj pro automatické procházení webů.

4.3 Ruby

Kromě výše zmíněného Mechanize, který má svoji podobu i pro Ruby lze s výhodou použít knihovnu s označením Nokogiri [22]. Je to nástroj pro parsování a vyhledávání nad XML, HTML dokumenty. Využívá CSS3 selektory, stejně jako XPath pro navigaci. Obsahuje XML a HTML builder.

4.4 Python

Asi nejznámější a nejvíce doporučovaný modul v Pythonu pro práci s webovými stránkami je BeautifulSoup [20]. Jedná se o parser nad webovými stránkami, který si stránku převede na strom, nad kterým pak lze provádět selekci dat.

Další knihovna Scrapy Open source framework for web scraping in Python [19] tvoří web scraping a web browsing framework. Může být využit při mnoha příležitostech od data-minigu až po monitorování a automatické testování webových stránek pro uživatele.

Html5lib je knihovna, určena pro parsování a serializaci, podle stávající specifikace pro HTML5. Opět se jedná o projekt, který má své derivace pro Python, Ruby nebo PHP.

4.5 .NET

Pro .NET Framework existuje tzv. HTML Agility Pack [23], který získanou stránku nejprve zpracuje do DOM, který lze číst nebo modifikovat. Jedná se o čistou .NET knihovnu, nezávislou na dalších knihovnách. Získaný a zpracovaný HTML dokument nebo stream lze následně převést třeba do XML. Poradí si i s nevalidními stránkami. Ve zpracovaném dokumentu se lze orientovat pomocí XPath. Kromě XPath podporuje XSLT transformace, nebo Linq to Objects (na bázi Linq to XML). Objektový model je podobný System.XML, ale pro HTML dokument.

4.6 Další možnosti

Watir [24] je cross-platform nástroj pro automatizaci webových prohlížečů. Chová se nad prohlížečem jako člověk, dokáže manipulovat s tlačítky, odkazy, formuláři a dalšími interaktivními prvky stránek. Slouží především pro psaní testů. Samotný Watir je pouze pro IE. Jeho nadstavba Watir-WebDriver [25] už dokáže fungovat nad všemi základními prohlížeči (Chrome, Firefox, Opera, IE), stejně jako běžet v „bezhavičkovém“ módu.

Obdobou Watiru je Selenium [26]. Taktéž slouží pro automatizaci webových prohlížečů. Jeho „výhodou“ je možnost testy neskriptovat, ale nahrávat.

5 PRÁVNÍ STRÁNKA VĚCI

Zákonů a práv, která by se dala aplikovat na web scraping, je asi celá řada. Nejsem právník a žádné oficiální právní stanovisko nemám, ani jsem jej nijak nezjišťoval. Z mého pohledu se užívání web scrapingu nejvíce týká práva autorského a duševního, práva na ochranu osobních údajů, popř. zákonů o informačních a informatických zločinech.

Při prohledávání českého internetu se mi nepovedlo najít konkrétní doporučení, či soudem řešené případy, které by se scrapingem souvisely. Dala by se čerpat jistá analogie z USA, kde těchto případů již bylo řešeno spousty. Bohužel český právní systém je od amerického dost odlišný, stejně jako právo duševní, kterého by se scraping také mohl týkat. Proto je těžko říct, jak by stejné činy byly posuzovány v ČR.

5.1 Autorské právo

Podle [10] je autorské právo zvláštní. Při jeho striktním výkladu dochází k jeho porušování téměř neustále. Pustíte si hudbu z Vašeho CD přehrávače, která je slyšet otevřeným oknem ven – porušujete autorský zákon. Pohvizdujete-li si melodii, třeba od Beatles, při průchodu zalidněným městem, porušujete zákon.

Na internetu je tomu obdobně. Podle [10] až 90% webových stránek tak či onak porušuje něčí autorská práva. Z autorského zákona vyplívající odpovědnost je absolutní odpovědnost objektivní. Pak tedy není rozhodující, zda došlo k porušení z dobré vůle či s cílem poškodit, není rozhodující rozsah porušení či zavinění. Pro vznik objektivní odpovědnosti stačí prokázat vznik protiprávního stavu a příčinné souvislosti mezi tímto stavem a jednáním osoby, která se deliktu měla dopustit.

Přestože k porušení autorských práv může dojít a tudíž i ke vzniku odpovědnosti, není ještě vůbec jisté, že oprávněná osoba bude svá práva vymáhat. Nelze určit přesnou hranici, kdy k tomu dojde, ale ze stávající judikatury lze odvodit určitá pravidla, ze kterých lze vycházet.

5.2 Ochrana osobních údajů

Osobní údaje, mezi které se řadí už jen jméno, jsou chráněny zákonem. Tudíž bez souhlasu té konkrétní osoby, které se údaje týkají, by se neměly tyto informace zobrazovat. Zde opět dochází k paradoxu, či porušení třeba indexujícími a vyhledávacími nástroji.

Z českého prostředí lze za praktické a aktuální téma považovat zveřejňování dlužníků v prostředí internetu.[11] Toto je často postihováno právě pro rozpor se zákonem o ochraně osobních údajů. Ze soukromoprávní povahy pohledávky vyplývá, že s ní je možno dále nakládat, např. ji prodat a inzerovat, což nelze bez zveřejnění základních údajů pohledávky – jako jméno a příjmení dlužníka, výše pohledávky a třeba její splatnost. Což koliduje s úpravou veřejnoprávní, která chrání zveřejňování osobních údajů.

5.3 Informační a informatické zločiny

Co se informačních a informatických zločinů týče se mi ohledně web scrapingu nepodařilo najít nic konkrétního. Článek zaměřující se na elektronickou informační kriminalitu [13], který řeší problémy spojené s „hackery“ a možnostech útoků, přes výčet nepřátelského softwaru a různými statistikami konče. Součástí článku je i pohled na právo, zejména autorské.

5.4 Doporučení

Z autorského zákona a [13] vyplývá zákonná ochrana pro SW, díla hudební a filmová, databází i webových stránek. Ale použití jiného autorského díla kromě softwaru či elektronické databáze pro vlastní potřebu fyzické osoby je však v České Republice legální i bez svolení autora.

Při shánění informací týkajících se web scrapingu jsem na internetu narazil vesměs na dva tábory lidí, co jej využívají. Lidé, co si nedělají starosti s tím co jejich „scraper“ napáchá, a lidé řídící se více či méně nějakým etickým kodexem.

Ten zhruba říká:

- vybírat si cílené zdroje pro svou potřebu – nestahovat pokud to nepotřebuji.
- tyto cíle neomezovat, nezpůsobovat výpadky, či nedostupnost, konkrétně „návštěva“ aplikace pro scraping by se příliš neměla lišit od návštěvy člověka, byť velice aktivního
- na takto získaných datech nevydělávat a nepůsobit finanční škodu cíli scrapingu, protože pokud se peněz týká, je to vždy důvod k žalobě
- použít informace ze souboru robots.txt, který je určen automatickým průchodům stránek, a dodržovat je

Další z pohledů na autorské právo se zaměřením na agregátor zpravodajství v ČR je [12]. V tomto rozhovoru-článku se redaktor a autor tohoto agregátoru zabývají pohledem na výdělečnou službu, která přejímá titulky, perexy⁴ a obrázky z jiných webů. Tam se samozřejmě týká o autorská díla, přesto je lze za určitých podmínek podle tohoto článku šířit dále a dokonce na nich vydělávat.

⁴ V žurnalistice – zpravodajství a publicistice, se jedná o označení pro krátký text, jehož cílem je uvést a nalákat na následující delší text článku

II. PRAKTICKÁ ČÁST

6 NÁVRH APLIKACE

Na internetu je spousta obecných řešení web scraperů, jejichž výstupem je obecně množina dat v podobě CSV, XML, či Excel nebo nějaký DBS formát. Teprve nad těmito daty dochází k vlastnímu zpracování, ale to si už každý klient, používající tato řešení, zajišťuje již sám. Vždy samozřejmě záleží na účelu, pro jaký jsou získaná data použita.

6.1 Obecné cíle aplikace

V praktické části představím své řešení problému získávání veřejně dostupných dat metodou web scrapingu se zaměřením na realitní agregátory. Protože se jedná o realitní agregátory, víme, o jaká data se bude jednat. Většina realitních serverů má obdobnou strukturu pro zobrazení a stejnou informační oblast dat. Nejedná se tedy o obecný web scraping jakéhokoliv webu.

Výstupem aplikace nebude tabulka anonymních posbíraných dat. Důležitým prvkem je i správná interpretace těchto dat, stejně jako uspořádání dat do příslušných tabulek a správných datových typů. Z tohoto pohledu má aplikace tři úkoly:

- zajistit si seznam webových adres na konkrétní detaily (ať už nemovitosti nebo realitní kanceláře),
- přečíst a zpracovat tyto detaily, získat z nich potřebné informace,
- tyto získané informace převést do nachystaných objektů pro snazší manipulaci, vyhledávání, uložení a interpretaci získaných dat.

Aplikace obsahuje jednoduché webové rozhraní pro zobrazení nasbíraných dat. Umožňuje základní vyhledávání nad těmito daty a zobrazení nalezeného detailu nemovitosti.

Pokud se uživatel přihlásí do administrátorské sekce, je mu umožněn pohled na stávající nastavení pro jednotlivé servery, s možností jej měnit. Obsahuje i základní statistiky o dosavadní práci serveru.

6.2 Možné problémy

Při odesílání dotazů na server se posílá i identifikace klienta odkud se žádá o data. Protože se dnes vytvářejí webové stránky pro různé druhy zařízení, které mají mnohdy zcela odlišnou podobu, může server na základě této identifikace vracet zcela odlišná data. Jedná se zejména o rozdíly mezi verzí pro chytré telefony, tablety a další mobilní zařízení. Data

mohou být ořezaná a jinak strukturovaná pro jednodušší čtení na menších displejích. Pro potřeby aplikace nastavení zůstalo defaultní pro desktopy.

Další problém, když už se nám podaří získat ze serveru odpověď, je kódování - určení znakové sady - takto získané stránky. Ve většině případů je to uvedené v hlavičce stránky, kterou se snažíme získat. V ostatních ji zkusíme převést na některé ze známějších typů podporujících češtinu (utf-8, ISO 8859-2, Windows-1250). V aplikaci je defaultně nastavena jazyková sada utf-8, ale v případě problémů ji lze změnit.

Každý agregační server s realitními nabídkami má vizuálně relativně podobnou strukturu, která se samozřejmě liší v popisu stránky. Uvozovací slova, kterých by se podle nějakého slovníku dala chytit, kolikrát chybí nebo se překrývají. Jelikož cílem je získávat data od více agregátorů, bylo nutné přijít se systémem šablon, které do určité míry jsou to samé jako v obecných scrapingových aplikacích nastavení filtru, které si buď naklikáte myší, nebo definujete jiným způsobem.

Až po návrhu aplikace jsem zjistil, že na stejném serveru se pro relativně stejná data používá různých druhů zápisu (třeba podle četnosti). Například na jednom zástupci agregátoru je pro jeden telefonní kontakt použit tag <p>, kdežto když je jich více již jsou uvozeny v tagu . O třídách a id jednotlivých elementů nemluvě. A bohužel se nejednalo o osamocený jev. Byl jsem nucen do svého šablonovitého systému přidat možnost několikanásobného definování podmínek pro získání stejných dat.

Nabízela se možnost využít sitemapy serverů pro snadnější přístup k datům. Bohužel takto velké dynamické množství nabídek je těžko udržovatelné vůči sitemapám. Všechny referenční servery je buď nemají udržované, popřípadě mají jen statické stránky, nebo je opustily úplně. Další možností je využití řazení dat přímo na stránkách. Jelikož hledající lidé zajímají nové nemovitosti, většinou je i řazení správné, nebo se nechá parametrem v url adrese vynutit.

Jeden z dalších problémů bylo, zda je řešení proveditelné a udržovatelné z hlediska časového. Jelikož se bavíme o agregačních serverech, které disponují i přes 160 tisíc detailů nemovitostí, pokud se prochází seznam po 10 nemovitostech, bude potřeba 16 tisíc těchto „index“ stránek, z kterých lze získat adresy detailů. Tím jsme na necelých 180 tisících stránek, které musíme stáhnout, přečíst, zpracovat a uložit. Nemluvě o realitních kancelářích a jejich makléřích, které tvoří dalších zhruba 30 tisíc. A to se týká pouze

jednoho serveru. Jaký hardware bude potřeba? Jaké datové připojení? A i kdyby toto bylo v pořádku, nebudou referenční servery pod příliš velkým náparem?

7 VYBRANÉ SOFTWARE NÁSTROJE

Před vlastním návrhem aplikace bylo třeba zvolit zástupce realitních agregátorů a prozkoumat jejich výstupní html kód s ohledem na potřebná data. Na základě tohoto rozboru vybrat i potřebné technologie na zpracování.

Pracuji v prostředí, kde používáme technologie Společnosti Microsoft. Tato platforma nástrojů je mi nejbližší. I proto jsem si zvolil jako vývojové prostředí MS Visual Studio Professional 2013, ve stávajícím posledním updatu. Používám programovací jazyk C# nad .NET Framework verze 4.5. GUI aplikace je pro webové prohlížeče, psáno v ASP.NET Framework.

Jelikož získaná data nepotřebují dodatečné zpracování na straně klienta, není důvod zajišťovat zpracování přes engine nějakého webového prohlížeče. Pro získávání a převod dat používám HtmlAgilityPack v 1.4.6 [23] popsany v předchozích odstavcích.

Pro uchovávání dat jsem se rozhodl pro Microsoft SQL Server 2012. Pro pouhé uložení dat není potřeba tak robustní nástroj, ale do budoucna se předpokládá nasazení algoritmů na doplňování, sortování, porovnání a výběr nad daty, kde už by se některé pokročilejší techniky poskytované tímto nástrojem mohly hodit.

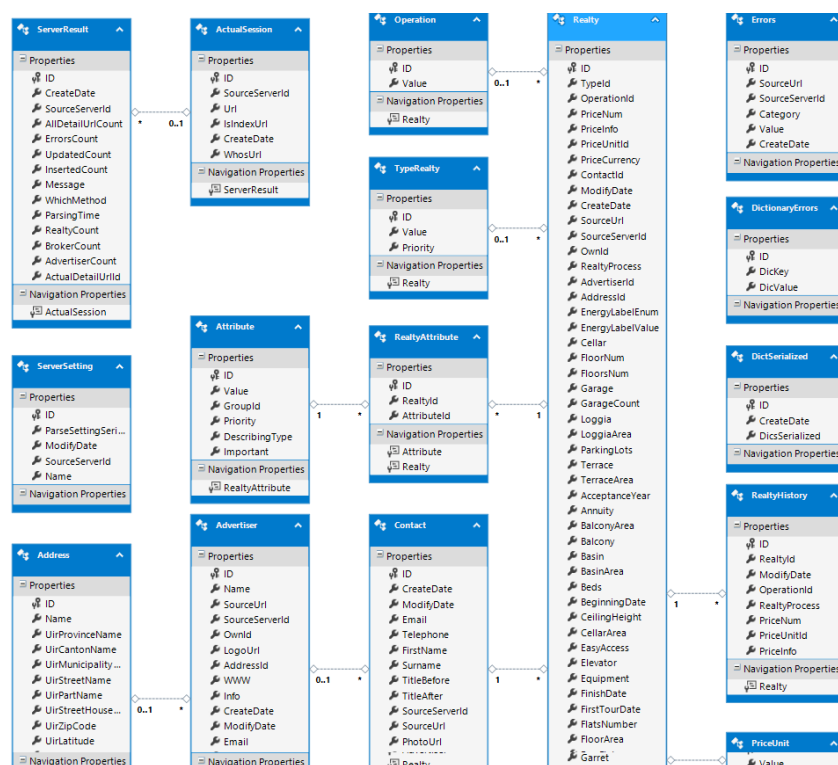
8 ARCHITEKTURA

Aplikace je navržena do třech vrstev. Jedná se o vrstvu datovou (Business logiku), Aplikační vrstvu, kde běží parsovací aplikace, a vrstva prezentační. Tam se prezentují data a očekává se uživatelská interakce. To mi v zásadě dělí aplikaci na tři projekty, kde se jeden stará o komunikaci s databází, pak vlastní parsovací knihovna a webový projekt sloužící jako uživatelské rozhraní.

8.1 DB projekt

Databázový projekt zajišťuje veškerou komunikaci s databází. Je využit Entity Framework verze 6.0. Obsahuje tzv. model, což je „mapování“ do databáze. Dále se v celé aplikaci pracuje již nad objekty, které jsou v rámci kontextu dostupné přes „.“ a lze se „protečkovat“ skrze objekty obsahující další objekty.

Databázový projekt taktéž obsahuje fasády, které zjednodušují vytahování potřebných dat a odstiňují tak aplikační logiku od dotazů do DB. Přesto že napříč aplikací lze s výhodou využít technologie LinqToSql, snažím se jí využívat jen v „business“ vrstvě. Jinak dochází k posunu databázových dotazů do vrstev, kde by již neměly být, a komplikuje se případný přestup na jiného poskytovatele dat.



Obr. 6: Databázový model aplikace

Součástí jsou i rozšíření databázových objektů v tzv. „partial“ třídách o další vlastnosti, či funkčnost, která je s výhodou používána výše v aplikaci.

8.2 Web scraping komponenta

Obsahuje kompletní logiku pro scraping webů. Celá komponenta je přístupná přes jednu řídicí třídu, která deleguje a ovládá činnost webscraperu. K tomu ji slouží zatím 3 metody pro získání realitních kanceláří, nemovitostí a doplnění realitních kanceláří, které byly doplněny během stahování nemovitostí. Při stahování nemovitostí se založí detail realitní kanceláře, pokud již neexistuje, ale s pouze základními údaji pro další identifikaci (její id na tom daném serveru).

Komponenta se vesměs rozděluje na tři funkční celky, z nichž má každý jiný úkol. Při řešení těchto úloh je s výhodou použito dědičnosti (všechny v základu musí stáhnout data a ty zpracovat). Jedna část načítá seznamy odkazů na další, většinou detaily, webové stránky. Další celek takto získaná data - odkazy - nejdříve stáhne a následně začne zpracovávat. Podle druhu detailu pak dochází k jeho převodu na databázový objekt, který je následně uložen. V této třetí části se aplikace liší od obecných scraperů, které by získaná data pouze uložila do nějakého podporovaného formátu.

Kvůli třetímu celku aplikace obsahuje množství různých slovníků, na jejichž základě se získaná data snaží „napasovat“ do známých objektů.

Součástí celé komponenty je ukládání a logování aktuální práce, tak aby na ní při pádu bylo možno navázat, dala se pozastavit v případě velkého vytížení serveru, a to nezávisle pro každou spuštěnou instanci komponenty.

8.3 Prezentační část

Aby aplikaci bylo možno obsluhovat, případně sledovat nějaký průběh (vyjma databáze, nebo logů), obsahuje aplikace také část prezentační. Ta se dělí na dva celky, jeden pro obecnou prezentaci získaných dat a druhý pro administrování nastavení parsovací komponenty.

8.3.1 Veřejný přístup

Veřejná část pro prezentaci dat obsahuje tři stránky, kde jedna je pouze úvod do projektu a stručný popis toho, proč stránky vznikly a co na nich lze dělat.


GetLiving Hello, admin! [Log off](#) [Administrace](#)

[Domů](#) [Nemovitosti](#)

Nemovitosti 10 z 25 395 nabídek


Lokalita.. Byt ☐ Prodej ☐ Min cena Max cena [další filtry](#) [Hledat](#)

[neinovější](#) | [neilevnější](#)




PRODEJ, BYT 2+1, 65 M²
Staroměstská, České Budějovice - České Budějovice 3
1 450 000 Kč za nemovitost včetně provize

Prodej družstevního bytu s balkonem o dispozici 2+1, 63 m² v Českých Budějovicích na ulici Staroměstská. Byt se nachází ve 2. patře zrekonstruovaného panelového domu bez výtahu. V bytě nová plastová okna, jinak je byt v




PRODEJ, BYT 2+1, 78 M²
Sladovnická, Vysoké Mýto - Pražské Předměstí
1 350 000 Kč za nemovitost včetně provize

Prodej bytu 2+1 ve třetím patře panelového domu bez výtahu ve Vysokém Mýtě. Byt je po rekonstrukci - elektřina v mědi, plastová okna, plovoucí podlahy, zděné jádro, nová kuchyňská linka, zděná šatna, koupelna s rohovou vanou,



PRODEJ, BYT 4+1, 82 M²
Na Honech III, Zlín
1 700 000 Kč za nemovitost

Nabízíme k prodeji byt 4+1, na JS, ul. Na Honech, 82m², s lodží. Byt se skládá ze 3 neprůchozích pokojů, velkého obývacího pokoje, kuchyně, chodby a samostatného WC a koupelny. Prošel částečnou rekonstrukcí - plastová okna,



PRODEJ, BYT 3+1, 80 M²
Čeladná (okres Frýdek-Místek)


Obr. 7: Stránka seznamu nemovitostí s filtrem

GetLiving Hello, admin! [Log off](#) [Administrace](#)

[Domů](#) [Nemovitosti](#)

Prodej, byt 3+kk, 87 m² [< zpět](#)

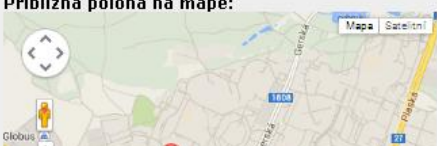
Studentská, Plzeň - Bolevec
2 660 000 Kč za nemovitost včetně provize, včetně právního servisu



Číslo RK: PL14SF167
 Operace: Prodej
 Typ: Byt
 Disposition: 3+kk
 Status: Novostavba
 Konstrukce: Cihla
 Ownership: Osobní
 Equipment: Nezařízeno
 Electricity: 230V
 Heating: Ústřední dálkové
 DriveWay: Asfaltová
 Development: Obytná
 Telecommunication: Telefon

[Spustit slideshow ▶](#)

Přibližná poloha na mapě:



Nabízíme Vám ke koupi byt 3+kk s balkonem a vlastním garážovým parkovacím stáním v komplexu novostaveb ve Studentské ulici v Plzni. Byt situován ve zvýšeném přízemí domu s výtahem. V bytě použity kvalitní prvky jakými jsou dlažby, dveře a zárubně dveří velmi pěkná kuchyňská linka s barem. Vstupní předstíh tvaru L, zde jsou praktické vestavěné zrcadlové skříně, prostorná skladová komora a dále přímé vstupy do pokoje, ložnice, toalety, koupelny a obývacího pokoje s kuchyňským koutem. V

Obr. 8: Detail nemovitosti

Další je již seznam nemovitostí (Obr. 7), který lze filtrovat pomocí přiloženého formuláře. Tam lze nastavit základní parametry jako lokalitu (v současné podobě pouze obce), druh nemovitosti (byt, dům, pozemek, komerce nebo jiné), dále zda se jedná o prodej, nebo pronájem a cena (maximální, či minimální). Lze nastavovat i podrobnější parametry jako dispozice, druh komerčního využití atd.

Při prokliknutí nějaké vybrané nemovitosti se zobrazí její detail (Obr. 8). Ten obsahuje název, cenu adresu, galerii obrázků, interaktivní mapu, seznam vlastností nemovitosti a vlastní popis nemovitosti. Součástí je jednoduchý výpis kontaktu na makléře a realitní kancelář.

Část aplikace pro veřejnou prezentaci dat slouží pouze pro prezentaci její funkčnosti. Proto i její vzhled a funkce si nekladou za cíl suplovat realitní vyhledávací portál. Může obsahovat chyby. Časem bude tato prezentace odstraněna, jelikož pro chod aplikace je zbytečná.

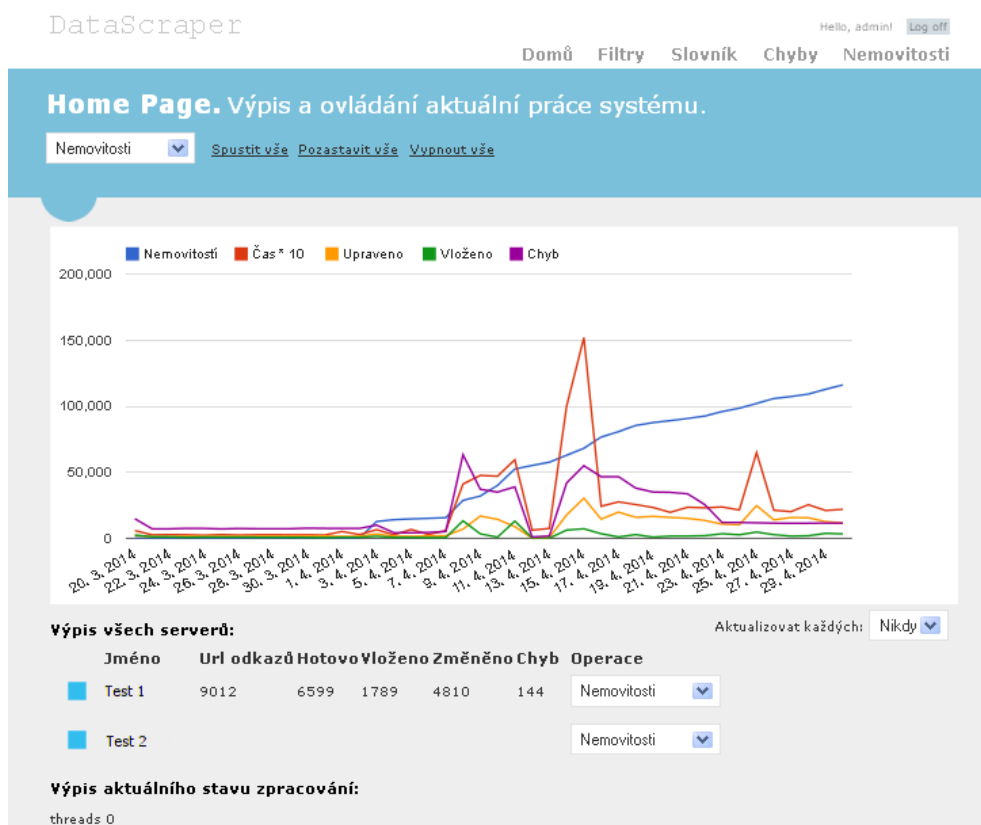
8.3.2 Administrativní přístup

Administrační část je přístupná pouze po přihlášení autorizovanou osobou a není umožněna registrace osob nových. Slouží k ovládání a výpisu informací o parsovací komponentě.

Na úvodní stránce (Obr. 9) je zobrazen seznam aktuálních serverů, nad kterými lze spustit scrapovací komponentu, graf s historickou činností aplikace a jednoduché ovládání pro jednotlivé servery, či pro hromadné spuštění, či zastavení.

Stránka Filtry (Obr. 10) drží nastavení pro jednotlivé servery, lze přidat server nový, nebo editovat stávající. Zobrazuje jednoduché statistiky pro běh konkrétního serveru. Protože jsem během vývoje byl několikrát nucen změnit šablony pro nastavení, rozhodl jsem se při vytváření formuláře použít technologie Reflexe. Tudíž jen vezmu objekt nastavení a pro jeho vnitřní objekty dynamicky vygeneruji patřičné části formuláře. Proto i názvy u formulářů jsou jména objektů či jejich vlastnosti.

Další z odkazů v menu je “Slovník” (Obr. 13) obsahuje veškeré slovníky pro celou aplikaci. Ze slovníků, lze mazat, přidávat, či jinak upravovat. Ale je hlídáno, zdali ve slovníku zůstane aspoň jeden výskyt hodnoty, který se vztahuje k vlastnosti nebo atributům nemovitosti. Těch je konečný počet a ve slovníku se mohou překrývat s jinou klíčovou hodnotou výrazu (kde „ústřední dálkové“ a „ústřední – dálkové“ zastupují stejný atribut).



Obr. 9: Úvodní stránka po přihlášení do administrace

Parsovací filtry. Zde se zakládá nové nebo editují stávající nastavení pro servery.

Nový

Správa filtru

Hlavní nastavení

Realitní kancelář

Makléř

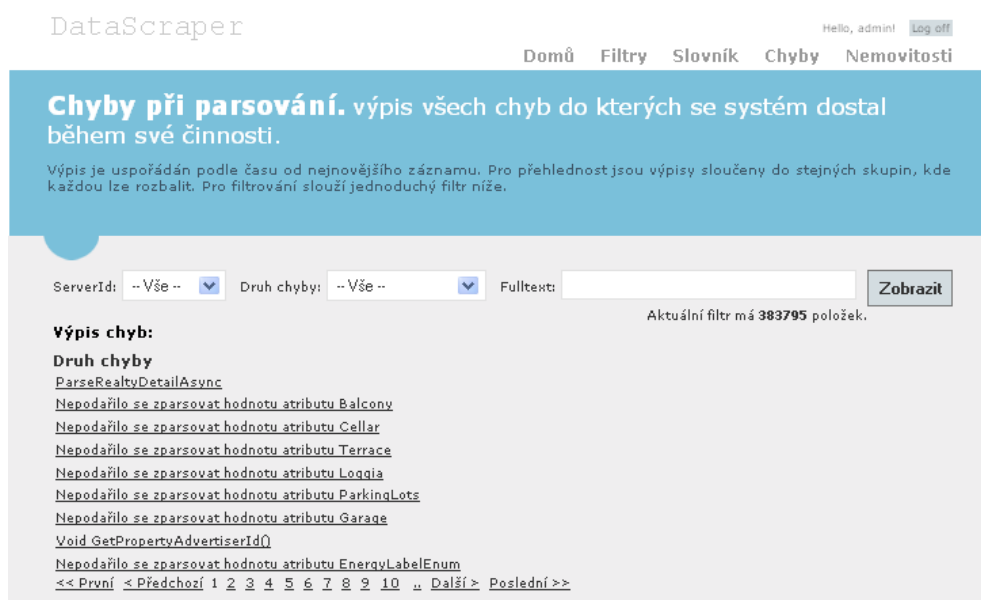
Nemovitost

© 2014 - My DataScaper Application

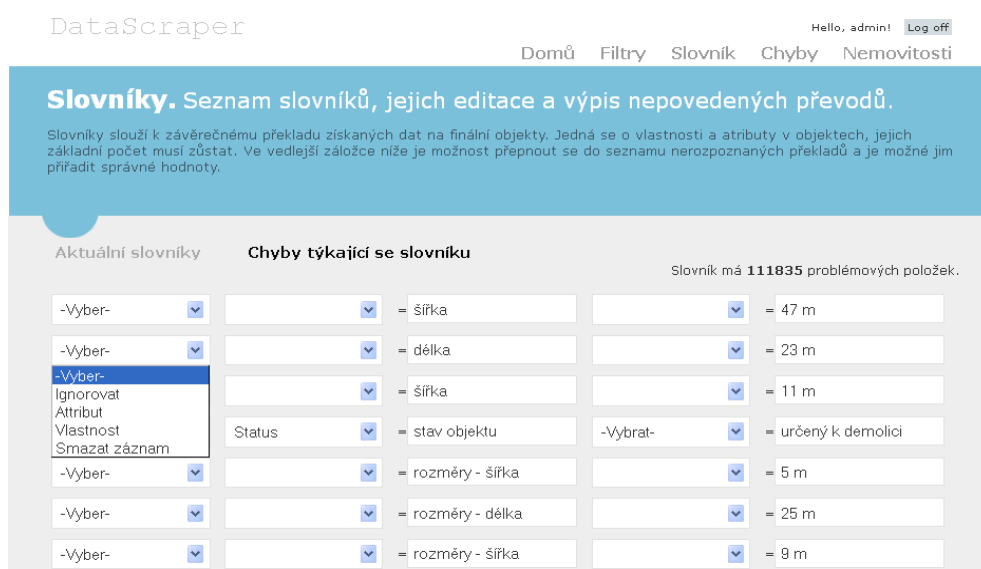
Obr. 10: Stránka s filtry

Sousední záložka (Obr. 12) obsahuje nástroj pro správu chyb ve slovníku. Jedná se o hodnoty, které nebyly nalezeny v překladových slovnících, a nebylo jim možno přiřadit správnou hodnotu v získaném objektu. Jedná se o tzv. atributy nebo vlastnosti nemovitosti. Pro každou nerozpoznanou hodnotu je zde řádek, pokud bylo aspoň určeno, zda se jedná o

vlastnost nebo atribut, jsou tyto vybrány v „DropDownListu“. Stejně tak, pokud byla rozpoznána skupina atributu, je rovněž předvybrána. Pak už se nastavuje jen překlad pro zobrazenou hodnotu⁵. Hodnotě konkrétního řádku lze nastavit, že se má ignorovat (nadále není zaznamenáváno jako chyba), nebo smazat, pak jsou všechny stejné hodnoty z databáze po uložení odstraněny.



Obr. 11: Stránka s výpisem chyb s jednoduchým filtrem

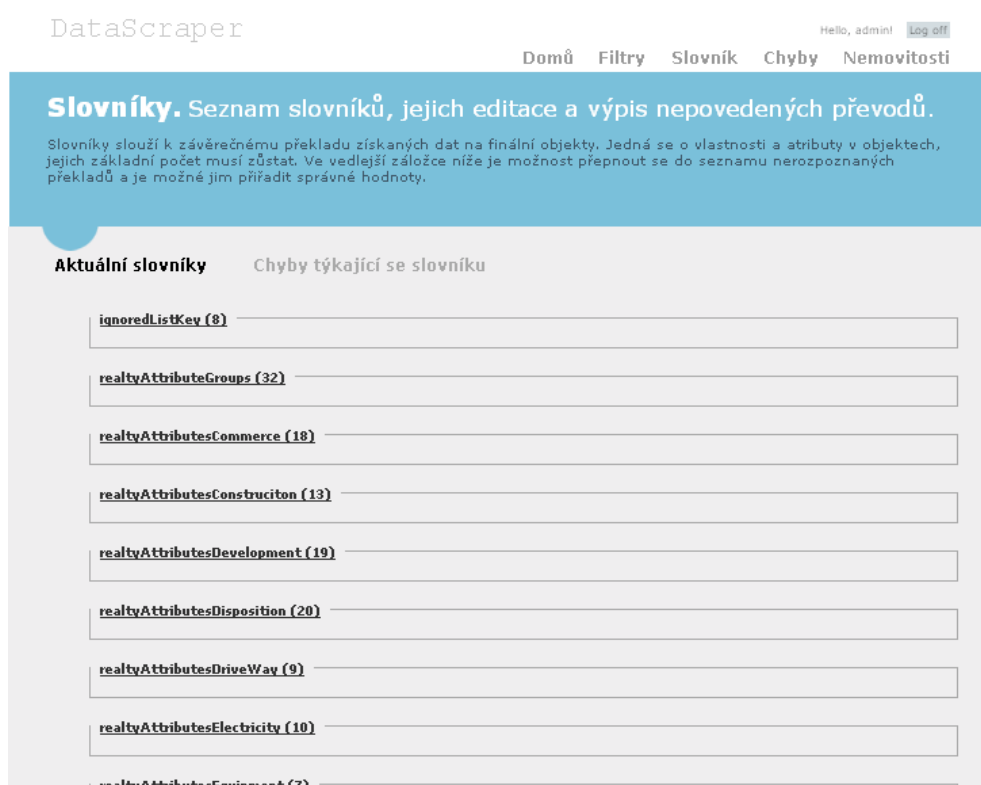


Obr. 12: Stránka s řešením překladových problémů

⁵ Jde o zpětný překlad, kdy známe v tomto případě český výraz a k němu přiřazujeme odpovídající hodnotu atributu nebo vlastnosti.

Předposlední odkaz v menu zobrazí stránku se všemi chybami (Obr. 11). Stránku lze filtrovat podle názvu serveru, druhu chyby a fulltextového vyhledávání nad popisem chyby. Výpis jednotlivých chyb je vždy sloučený podle typu chyby a teprve po rozkliknutí se rozbalí seznam s datem, adresou detailu kde chyba nastala a jejím popisem.

Poslední odkaz v menu vede na stránku nemovitosti, což už je stránka z veřejného rozhraní, kde se dají vyhledávat nemovitosti podle nastavení filtru. V případě přihlášení se lze následně dostat do administrace odkazem v pravém horním rohu.



Obr. 13: Stránka s nastavením slovníků

9 ŠABLONY

Už v textu dříve (kapitola 6.2) jsem se zmínil, že nastavení pro jednotlivé servery, nad kterými lze spustit scrapovací komponentu, je psáno formou šablon ve formátu XML. Stalo se tak díky narůstajícím nárokům na funkčnost, způsoby a reakce na změny na parsovaných serverech při různém zobrazení stejných skupin dat.

9.1 Objekt typu „item“

Po několika úpravách jsem skončil u návrhu třech nejnižších typů objektů – „item“, „path“ a základní datový typ (int, string, bool). „Item“ v sobě drží šest položek.

- XPath – cesta k hledanému HTML elementu
- Metoda – (InnerText, InnerHtml, Attribute, HtmlNode, HtmlNodes) čistý text, vnitřní html, nějaký atribut z elementu, celý element, nebo kolekci elementů
- Jméno atributu – jaký atribut se vytáhne v případě zvolené metody Attribute
- Regex – regulární výraz aplikovaný na získaný text
- Jméno položky v kolekci – při metodě HtmlNodes se jedná cestu k HTML elementu definující jméno
- Hodnota položky v kolekci – při metodě HtmlNodes se jedná cestu k HTML elementu definující hodnotu

Poslední dvě položky jsou trochu speciální. V kombinaci s metodou HtmlNodes lze uplatnit, pokud je na stránce výčet položek, popř. jako je u nemovitostí zvykem tabulka o dvou sloupcích s popisem položky na straně jedné s hodnotou položky na straně druhé.

9.2 Objekt typu „path“

„Path“ označuje položku pro zadání indexovacích stránek. Sestává ze čtyř položek:

- Cesta – url adresa kde má indexovací parser začít – může obsahovat proměnnou, která se následně bere z rozsahu parametrů.
- Rozsah parametrů – jednotlivé proměnné, měnící se v indexovací url adrese.
- XPath – cesta k jednotlivým elementům, které ze stránky potřebuji dostat.
- NextPath – cesta k odkazu na další stránku.

„Item a path“ jsou nadále uloženy v kolekcích. Například budu chtít jméno realitní kanceláře, to se nachází na stránce a vytáhnu ho pomocí nastavení „item1“, ale pokud realitní kancelář nemá logo, předchozí „item1“ již toto jméno nenajde (jiný element s jiným id i třídou), musím definovat další „item2“. Může se stát, že pokud má realitní kancelář více poboček jméno je formátováno opět jinak a „item1 ani item2“ se na jméno nedostanou. Založím si „item3“, ale pořád řeším jen jednu položku – jméno realitní kanceláře. Takovýchto případů lze najít více.

Navzdory veliké popisovací schopnosti objektů typu „item a path“, se tato v některých případech ukázalo jako nedostačující. Proto jádro parsovací komponenty je navrženo pomocí dědičnosti tak, že vše pro konkrétní objekt obstarává jedna základní třída, z které lze dědit a přepsat problémové metody. A to ať už v části, která zpracovává data, nebo i v části, která se je již snaží napasovat na existující objekty.

Celá tato problematika je mnohem komplexnější a individuální v závislosti na typu dat a serverů, z kterých čerpáme.

10 FILTROVÁNÍ DAT

Filtrování, neboli čištění dat, je jedno z důležitých témat při sběru dat z různých zdrojů. Data musejí být co nejvíce přesné a aktuální, tak aby nenabízela již stažené, nebo změněné nemovitosti. Problém duplikování dat je asi největší problém, pokud data již máme.

Najít duplikovaná data je důležité jak z pohledu koncového uživatele, který se musí prodírat hromadou stejných dat, ale i z agregačního serveru. Už při vzniku agregačních serverů tyto řešily, jaká data zobrazovat nejdříve. Většinou se přiklání k technice poslední změny nemovitosti, případně v kombinaci s datem jejího vložení.

Realitní kanceláře neváhaly a přišly s řešením kompletního odstranění zakázky a její nahrání pod jiným číslem znovu, tzv. „prolévání“. Toto chování je samozřejmě nežádoucí a je na každém agregačním serveru jak se s ním vypořádá.

10.1 Duplicity „prvního“ druhu

Výše popsany problém s rozpoznáním stejné zakázky jsem zařadil do duplicit prvního druhu. Týká se jedné nemovitosti, jedné realitní kanceláře, ale nalezené na více scrapovaných serverech, popřípadě znovu vkládané na stejný server samotnou realitní kanceláří (zpravidla se týká vkládání přes API).

Při importech skrze API je toto relativně snadné, jelikož data chodí ve stejném formátu, a pokud není změněno realitní kanceláří, i se stejnými daty, ale jinými identifikačními čísly.

V případě web scrapingu opět můžeme řešit problém v rámci jednoho parsovaného serveru, kdy jedna nemovitost byla odstraněna a nová – stejná přibyla. Ale v případě scrapingu přibývá jiný problém. Pokud beru data z různých serverů, je pravděpodobné, že tyto obsahují z velké části stejná data, přestože jinak zobrazena (stejné realitní kanceláře, se stejnými nemovitostmi). Nalézt takto stejné nemovitosti napříč různými servery už je větší oříšek, nicméně pořád se je čeho chytit. Jedná se o realitní kancelář, makléře a popis nemovitosti.

Jméno, popř. IČO pokud je známo, realitní kanceláře musí být stejné, podobně jméno, telefon a mail realitního makléře by mělo taktéž odpovídat. Popis nemovitosti by v hlavních parametrech měl také odpovídat. Ale kvůli nepřesnostem při převodu scrapovaných dat a kvůli vynechávání některých dat samotnými agregačními servery, nemají tyto množiny 100% překrytí a duplicity nemusejí být odhaleny.

10.2 Duplicity „druhého“ druhu

O dost větší problém, který by výsledný realitní agregátor odlišil od své konkurence, by bylo odstranění duplicit tzv. „druhého“ druhu. Tuto vlastnost by ocenili zejména potencionální klienti některých realitních kanceláří. Umožňovala by výpis jedné nemovitosti s odkazy na všechny realitní subjekty, jež ji nabízejí, spolu s výpisem jejich cen k dané nemovitosti. Díky této vlastnosti by se dala určit exkluzivita prodeje.

Duplicita „druhého“ druhu se týká jedné nemovitosti, na jednom serveru (popř. na několika u web scrapingu), ale inzerována u více realitních kanceláří. Problém je nasnadě. Tím, že nemovitost inzeruje více realitních subjektů, každý si dělá vlastní fotky, vlastní popis a zaškrtná / vyplní vlastní vlastnosti a parametry dané nemovitosti.

V tomto případě již nelze využít informace o realitní kanceláři, popřípadě makléři. Ty se samozřejmě liší. Jediná věc co zůstala na porovnání, se týká samotné nemovitosti, ale díky subjektivní perspektivě priorit se zadání, respektive získaná data, hodně liší. Takže v konečném stavu toho k porovnání moc nezůstane.

Občas se povede najít nemovitost, jejíž obrázky dodá majitel, a když je pak nemovitost inzerována u více kanceláří, má stejné fotky. Ale to se stává jen velmi zřídka, navíc porovnávání obsahu dvou fotek je výpočetně velice náročné.

10.3 Řešená filtrace v aplikaci

Filtrace v aplikaci řeší jen první druh duplicit. U nemovitostí obsahující nějaký identifikátor zakázky realitní kanceláře, je tento využit k porovnávání. U zakázek, které tento identifikátor neobsahují, se začíná od porovnání realitních kanceláří, následně jejich makléřů, a až po té dochází k porovnání nemovitostí za využití předešlých párování.

Tato metoda není a nemůže být 100% úspěšná, protože každý realitní agregátor podle svého uvážení vynechává podružné informace, třeba z důvodu estetického. Ono vypsání třeba 60 vlastností a atributů určitě na pěkném vzhledu stránky nepřidá. Někdy se můžou lišit i texty jednotlivých nabídek. Nadpisy jsou převážně generované, a tudíž rozdílné a pro porovnání nepoužitelné.

11 VYUŽITÍ DAT

Co se získanými daty? Využití je mnohé, stejně jako počet důvodů proč taková data získávat. Pokud vezmeme v potaz majoritu všech dostupných dat z nějakého odvětví, pak nad těmito daty lze dělat různé analýzy. Zkusím nastínit několikero oblastí, kde a jak se dají takováto data využít.

Jedním z důvodů proč se vydat touto cestou web scrapingu může být nástup na cizí trhy. Většina agregátků potažmo realitních softwarů má nějaké exportní API. Ale pokud chcete začít např. v nové zemi, kde ještě nemáte žádné obchodí zastoupení a klientelu začnete teprve oslovovat, přesto je dobré už mít nějakou základnu a mít co nabízet. V úvodu jsem psal, že pro každý realitní agregátor, aby měl úspěch, je stěžejní návštěvnost a data. A tato metoda k datům může pomoci.

Zbytek záleží na kvalitě, přístupu, reklamě a dalších spíše marketingových krocích. Oslovit a napřímo napojit hodně realitních kanceláří je časově nesmírně náročné, nemluvě o problémech napojení přes různé API. Hlavní artikl agregátorů nemovitostí je samozřejmě počet potencionálních klientů, které přivede té konkrétní realitní kanceláři. A pokud máme nějaká čísla a reference, je vstup na nový trh samozřejmě snazší.

Jako další využití takto sesbíraných dat je například sledování migrace realitních makléřů. I když tento úkol není až tak jednoduchý, jak by se mohlo zdát. V českých poměrech je velké procento makléřů pracujících na IČO. Ne zřídka se nechávají najímat více kancelářemi najednou, přebíhají a zkoušejí.

Snadněji získatelným ukazatelem z dat může být velikost realitní kanceláře ať už z pohledu počtu nemovitostí, nebo z pohledu počtu makléřů pro tuto kancelář pracujících.

Lze sledovat průměrné délky inzerce jednotlivých typů nemovitosti, které jdou na odbyt a které jsou tzv. „držáky“. Bohužel data o realizaci prodeje se většinou nedostanou za zdi realitní kanceláře. Stejně tak se těžko poznává exkluzivita smluv (viz kapitola 10), tj. zda se nemovitost prodává výhradně pod hlavičkou jedné konkrétní kanceláře či nikoliv.

Na základě realitních dat lze tvořit orientační cenové mapy. Orientační z důvodu absence informace potvrzující výši uzavřeného obchodu, ale s nějakou procentuální spolehlivostí lze určit průměrné ceny třeba bytu 3+1 o 75m² v různých lokalitách jednoho města, napříč městy, nebo kraji, popř. v mezinárodní úrovni napříč zeměmi.

12 VÝKON

Jelikož aplikace bude stahovat a zpracovávat velké množství dat, je pro ni důležité vybalancovat zatížení systému vůči zatížení sítě. Aplikace pravděpodobně nebude mít celý server sama pro sebe. Tudíž se bude muset chovat ohleduplně i vůči dalším aplikacím běžícím na stejném serveru.

12.1 Kolekce

Vlastnosti jsou převážně číselné údaje určující podlaží, plochu, datum (kolaudace, rekonstrukce, ...). Atributy na straně druhé, v podstatě databázové „enumy“, jsou kolekce přiřazující nemovitosti informace o dispozici, odpadech, přivedené elektřině nebo umístění objektu v rámci obce či zástavbě. Dohromady se jedná o necelých 250 možných nastavovaných hodnot jen pro nemovitost. Ostatní – realitní kanceláře a makléři – již takové množství informací neobsahují.

Během vývoje aplikace jsem, kromě samotného řešení, narazil na několik problémů, které by mohly negativně ovlivňovat zátěž serveru. Jeden z prvních byl, na jakou datovou strukturu převést skupiny atributů, které mají dvou úroňovou hierarchii, tvořenou vždy skupinou a samotnou hodnotou atributu.

Jedná se o necelých 130 hodnot v 21 skupinách, což na vyhledávání není mnoho. Ale protože by se tato operace vyhledávání mohla opakovat milion a půl krát během převodu nemovitostí jednoho serveru, nakonec jsem se rozhodl převést všechny skupiny atributů na jednotlivé slovníky a jejich hodnoty do nich zaznamenal. Atributy všech skupin mohly být v jednom slovníku bez vlivu na rychlost, ale z důvodů další správy jsem zvolil více slovníků.

Další rozhodnutí se týkalo vlastností samotného objektu nemovitosti. Tam se jako výhodné jevílo použít buď pomalejší reflexi a vždy nastavovat jen získané vlastnosti, nebo zkoušet nastavovat vždy všechny vlastnosti z dostupných dat.

Po provedení několika rychlých testů jsem dospěl k názoru, že reflexe je zhruba 10x pomalejší, než přímý přístup. Pro účely aplikace při průměru 10 vlastností na nemovitost a trvání 4ms na nastavení hodnoty přes reflexi, vůči projití 70 vlastností za 0,3ms, jsme na polovině času při přímém přístupu. Samozřejmě tyto časy by se při ostrém nasazení asi lišily, jelikož kolem nastavování vlastností je spousta dalších převodů.

Určitě zajímavý test by byl na ostrých datech na vzorku alespoň 10 000 nemovitostí a ne jen spočítané na základě jedné nastavovací procedury (byť 150 000x volané)

12.2 Vlákna

Jelikož se jedná o časově i výpočetně náročné operace a během získávání dat při čekání na odpověď serveru je prostor pro čekání, rozhodl jsem se pro využití vláken. Ukázalo se, že se celková potřebná doba zkrátila na polovinu, přestože na vzdálené servery neposílám více jak jeden dotaz současně. Abych byl přesný – kromě indexovacích dotazů.

Během testů na rozpracovaném programu jsem našel dvě úzká místa zabírající nejvíce času. Jedná se o stažení dat ze vzdálených serverů a následné vytvoření speciálního XML dokumentu z HTML Agility Pack frameworku a jeho zpracování.

Každá instance spouštěná na vláknech má celkem tři vlákna. První se stará o načítání a zpracovávání index stránek odkud se získávají odkazy k vlastním detailům. Další dvě vlákna mají na starosti časově náročné operace – stažení dat a převedení na speciální XML (spolu s jeho zpracováním a uložením).

Pro společnou práci všech vláken, protože jsou na sobě závislá, jsem pro synchronizaci vyžil dvou Monitorů s Pulse a Wait metodami. Jako strategie je využita Klient – Producer. Metody na konkurenčních vláknech jsou zhruba stejně rychlé a v případě rychlejšího Klienta, dojde k jeho zastavení, ale v případě Producera rychlejšího než Klient se data kupí ve frontě a v paměti. Toto se během ostrého provozu nepotvrdilo; vlákno, které načítá data je minimálně o 1/3 pomalejší než zbytek.

12.3 Databáze

Data v databázi se rychle shromažďují. Pro každý server jsou data, přestože se více či méně duplikují, držena zvlášť. Probíhají nad nimi dotazy kvůli aktualizaci již existujících dat. Během testování i nad relativně malým vzorkem nemovitostí čítající dohromady kolem 40 tisíc nemovitostí již prodlevy ve zpracování databází byly patrné.

Vytěžované tabulky jsem opatřil indexi a sdruženými indexi nad parametry, které jsou používané spolu. Indexi, ať už jednoduché nebo složitější, by se měly postupně doladit v nějakém analytickém nástroji pro optimalizaci dotazů. Tyto si umí říct, které indexi chybí, v kterých chybí parametry, popřípadě, které se nepoužívají.

V tabulce níže je orientačně naznačen vývoj aplikace v čase a v různých prostředích. V popisu je poznačeno, vůči jakému vzorku proběhlo měření a následuje přepočet na jednu nemovitost. V posledním sloupečku je následně zobrazen vypočítaný údaj pro teoretické stažení 100 tis. nemovitostí.

Z tabulky je vidět, že (obzvláště v pozdějším stadiu vývoje) stažení dat je jednou tak pomalejší než jejich zpracování a uložení.

Popis	chyb na nemovitost	čas na nemovitost (s)	stažení nemovitosti (s)	zpracování nemovitosti (s)	čas na získání 100tis. nem. (h)
testování na localhost / 240	-	1,25	0,192	0,788	34,72
localhost / 240	3,6	0,345	0,147	0,183	9,58
localhost - vlákna / 120	3,258	0,208	0,208	0,155	5,78
localhost - vlákna / 1020	3,221	0,173	0,173	0,156	4,81
hosting / 240	3,525	0,146	0,091	0,049	4,06
hosting - vlákna – začátek / 1300	2,99	0,131	0,131	0,101	3,64
hosting - vlákna – konec dubna / 9104	1,78	0,097	0,097	0,049	2,69

Tab. 1: Průměrná doba zpracování nemovitosti v různých prostředích během vývoje

12.4 Způsob získávání dat

Další možností, jak urychlit proces získávání dat scrapováním, je předcházení duplicit. To znamená, že ze všech serverů nebudeme stahovat vše, ale pokud to daný server umožňuje, půjdeme přes seznam realitních kanceláří. Pokud již mám danou realitní kancelář staženou přes předchozí server, z dalšího tyto data již nebudu stahovat.

13 MOŽNÁ DALŠÍ VYLEPŠENÍ

Stávající řešení bych osobně nazval fungujícím prototypem. Určitě obsahuje chyby a zcela určitě obsahuje prostor pro vylepšení, ať už po stránce uživatelské nebo technické.

13.1 Uživatelská stránka

Pominu-li veřejnou prezentaci, která má prostoru pro vylepšení víc než dost, ale v současné verzi slouží pouze k jedinému účelu – prezentaci scrapovaných dat (do budoucna se s ní nepočítá), zbývá část administrátorská.

V současné podobě je zadání a rozběhnutí nového serveru k uspokojivým výsledkům práce spíše pro programátora. V administraci je nástroj pro vytvoření nové šablony, ale obsahuje spoustu parametrů a nastavení, o kterých zadavatel musí vědět, co znamenají. Není testovací nástroj pro otestování nastaveného filtru. A navíc v případě „specialit“ se musí do kódu a dopsat změny ručně.

Uživatelům by pomohlo příjemnější zadávání filtrů pomocí myši přímo nad stránkou, kterou chci získat, tak jak je u některých obecných verzí možné.

V případě potřeby zisku dat z javascriptu doimplementovat vestavěný prohlížeč. Stávající řešení si s ním neporadí. Jedná se spíše o okrajovou záležitost.

Názvy na stránce Filtr a Slovník jsou pouze generované z názvu objektů za pomoci reflexe. Pro uživatele by určitě bylo snazší s českými ekvivalenty (další překladové slovníky) a nejlépe s popisky na co který parametr slouží.

13.2 Technická stránka

Jádro aplikace je v současné době navrženo na „hyper aktivního“ uživatele. To znamená, že se aplikace snaží chovat jako člověk a nevytěžovat cílové servery násobným připojením. Nemá nastavené limity pro stahování, tudíž bere, co stačí dávat / pobírat přenosová zařízení.

Jako vylepšení může být nastavení počtu stahovacích vláken, popřípadě s nastavením nějaké drobné latence. Aplikace by si následně dynamicky volila počet parsovacích vláken podle připravených stránek ve frontě.

Nyní má aplikace defaultně nastaveného klienta, který navazuje spojení se vzdálenými servery a vydává se za uživatele. Tomuto klientovi lze nastavit úplně jiný „podpis“, pod

kterým by se přihlašoval. Ale nevím, zda tím lze dosáhnout lepší produktivity nebo „ochrany“ ze stran cílových serverů.

Může se stát, že se některé servery začnou automaticky bránit proti častým dotazům, nebo množství odchozích dat. V nejhorším případě odhalit, že se nejedná o člověka, ale o stroj, a přidat si takovou adresu na „blacklist“ bez možnosti o navázání dalšího spojení.

Na odborných fórech zabývajících se touto tematikou byla řešení využívající několikero prohlížečů, které následně posílaly data na server, který je zpracovával. Další možnost je využití Proxy serverů a různě nastavených socketů. Ale s touto problematikou jsem neměl možnost se blíže seznámit.

14 POROVNÁNÍ S AKTUÁLNÍMI ŘEŠENÍMI

V českém prostředí jsou desítky realitních agregátorů. Drtivá většina získává data pro další šíření přes API, ale existují i výjimky využívající web scraping. Samozřejmě netuším, kolik robotů se potuluje třeba po realitních serverech získávající informace pro firemní účely různého druhu, které nejsou opět publikovány veřejnosti. Níže pak zmíním i porovnání vůči obecným placeným scrapovacím programům.

14.1 Existující realitní servery na bázi web scrapingu

Existující aplikace, využívající technologii web scrapingu jako získání dat k dalšímu zpracování a prezentaci, znám dvě. Jedná se o server GoHome, který parsuje většinu realitního trhu, dle mého názoru nejen, v České Republice. Další projekt má název Realingo, který se vyjímá zpracováním a zobrazováním dat. Pokud mohu soudit, projekt GoHome je více „rovnější“ před zákonem, co se týče prezentace dat než projekt druhý. Jak by skutečně obstáli v případě soudní pře, se lze jen dohadovat.

Jejich jádro aplikace, ani hardware na kterém běží neznám. Stejně jako způsob obnovy a údržby dat, která zobrazují.

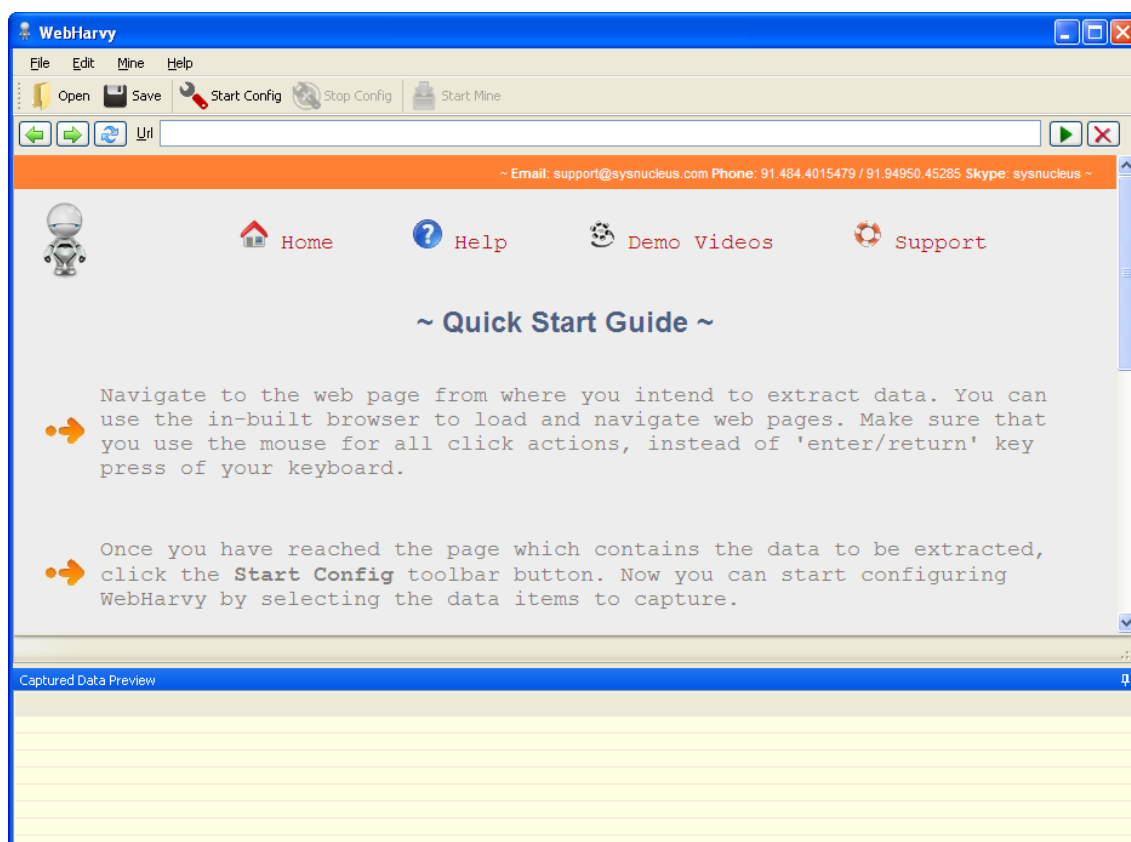
Projekt GoHome je poměrně známý a lidmi využíváný. Za své služby si nechává platit, tedy od stran realitních agentur, nebo agregátorů. Jedná se o jednoduché uživatelské rozhraní s jedním vstupním polem, kde se textově specifikuje, co se hledá. Následně zobrazené výsledky jsou bez obrázků, pouze odkazy s popisem a cenou, u které je hlídána její historie. Jednotlivé odkazy vedou na konkrétní servery, odkud daná informace byla vytěžena.

Server Realingo oproti tomu vsadil na bohaté a zajímavé uživatelské rozhraní včetně práce s mapou, kde se zobrazují vytěžené nemovitosti, a to i s obrázky a popisy. Každá nemovitost umožňuje zobrazit její detail v prostředí Realinga, nebo přesměrovat na zdrojový server. Bohužel se mi v některých případech nedařilo zobrazit požadované informace, zapříčiněné chybami, nebo nepochopením uživatelských prvků.

14.2 Obecné web scrapery

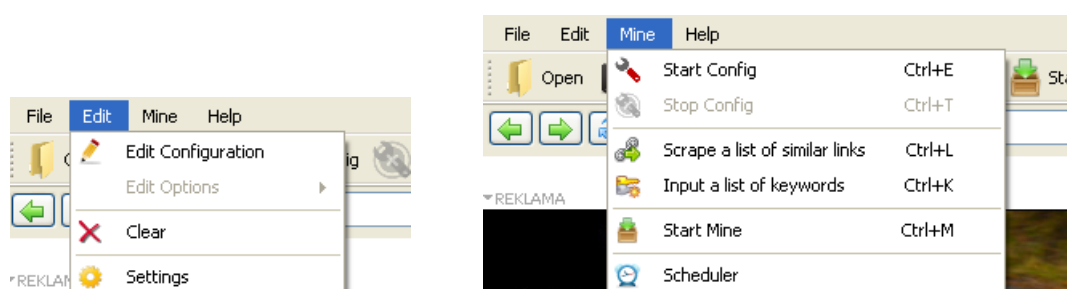
Vyzkoušel jsem i některé nabízené web scrapery. Jako zástupce jsem vybral dva, které mě z nějakého důvodu zaujaly. Oba dva zástupci jsou z řad desktopových aplikací a obsahují svůj kompletní interní prohlížeč.

14.2.1 WebHarvy



Obr. 14: Web Harvy scraper – hlavní stránka

WebHarvy [28] je vizuálně jednoduchý web scraper. Jedná se o desktopovou aplikaci. Lze ji stáhnout v okrájené demo-verzi, která umožňuje stahování jen omezeného počtu informací na dvou stránkách, stejně jako zadání pouze dvou časovačů. Díky tomu zkoušení s tímto programem bylo trochu omezené.

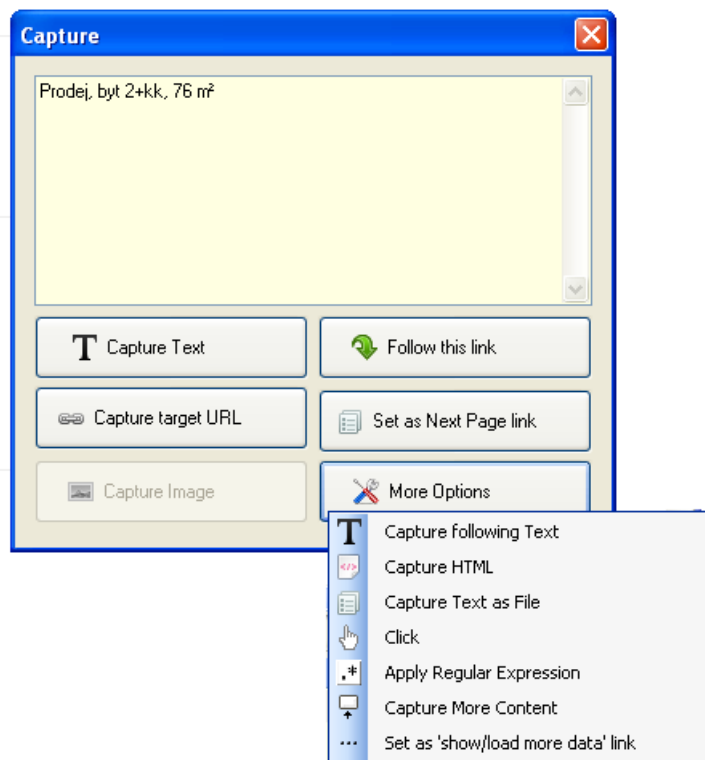


Obr. 15: Menu aplikace umožňující nastavení

Přestože program je relativně jednoduchý na ovládání, nedařilo se mi v něm udělat některé pro mě potřebné věci. Když se mi povedlo definovat stahování podobných odkazů z tzv. "index page", už se mi program nepodařilo přimět k označení následující stránky index page. Ze získaných odkazů byl schopen vytáhnout informace, které mu byly nadefinovány.

Ale nemyslím si, že by šlo nastavit alternativy (pokud to nezískáš zde, tak se podívej támhle).

Při vytváření nastavení pro scraping si program ukládá absolutní cesty skrze html, což může být náchylné na poruchy při změně i drobného nesouvisejícího tagu nebo třídy. Práce s programem byla jednoduchá, ale omezená. Nicméně na jednoduché pravidelné stahování pár informací asi plně dostačující.

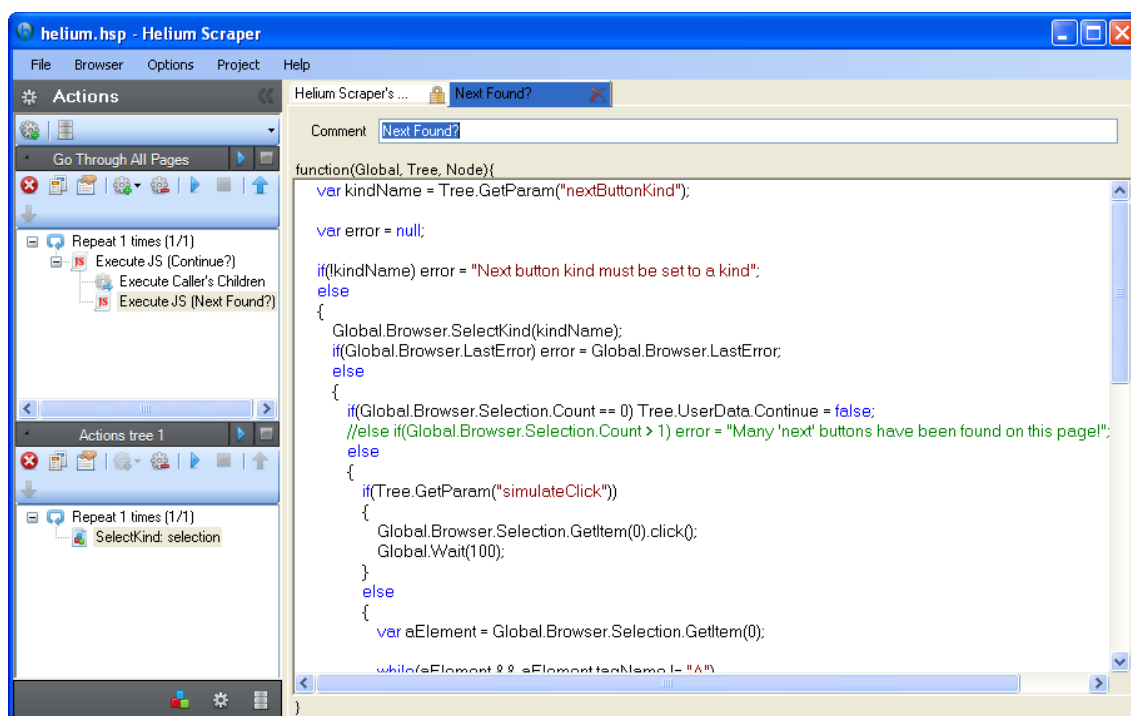


Obr. 16: Nastavení jednoho stahovaného prvku

Kontrola nad daty mi připadala nedostatečná, obzvláště pro další interpretaci. Např. při najití podobného vzoru v tabulce pěkně vzal sloupce s pojmenováním dat a vedlejší s hodnotami dat, ale ve výsledku se díky jedné chybějící hodnotě v prvním sloupečku řádky posunuly, což z nich tvoří odlišné výsledky.

Výstup byl do tabulky uspořádané podle zadávaných a pojmenovaných skupin dat. Výstupy lze ukládat do několika typů souborů a do dvou typů databáze.

14.2.2 Helium



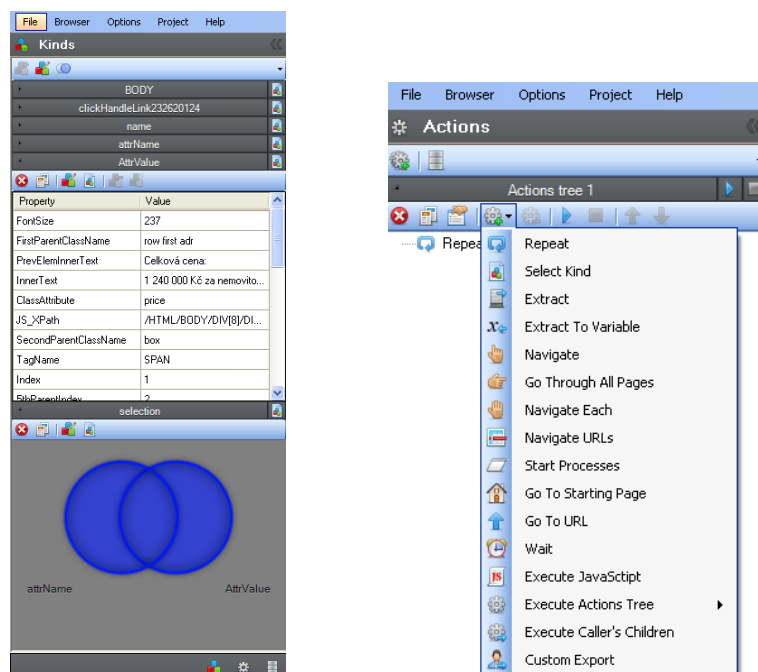
Obr. 17: Helium Scraper – hlavní stránka

Druhý web scraper, který jsem zkoušel je na první pohled více sofistikovaný. Jmenuje se Helium Scraper [29] a opět se jedná o desktopovou aplikaci. Disponuje vlastním internetovým prohlížečem. Je možné pracovat s javascripty ve stránce, ale i možnost zadávání javascriptího kódu pro ovládání vlastního scrapování.

Protože program umožňuje velké množství různých nastavení, už se nejedná o příliš intuitivní cesty, jak toho dosáhnout. Bez studování tutoriálů, nebo návodů se mi nepovedlo nastavit stahování podobných prvků na stránce pro „index page“, natož následně vybrat části tabulek podobného ražení pod každým odkazem.

Nastavení a definování filtrů je velice sofistikované, ale při složitějším využití aplikace se bez návodu asi neobejdete. Na straně druhé, definování jednoduchého filtru není problém a pomocí myši jej „naklikáte“ během chvilky.

Program ukládá filtry pod vlastním binárním formátem. Vlastní výstupy pro data jsou podobné jako u předchozí popsané aplikace.



Obr. 18: Menu aplikace umožňující nastavení

14.2.3 Rozdíly

Aplikace se porovnávají velice těžko, jelikož byly vytvořeny pro jiné účely. Ve srovnání s mnohou vytvořenou aplikací se v předchozích případech jedná o uživatelsky příjemnější zpracování.

Možnosti zadávání filtru pro jeden prvek jsou podobné, kdy si lze vybrat, co z daného prvku potřebuji – text, atribut a pod jakým jménem, vnitřní html, popřípadě přiřadit regulární výraz pro lepší vyjmutí požadované hodnoty. Na rozdíl od obecných scraprů má aplikace neukládá cestu k danému prvku absolutně, ale jen relativně, tudíž není tak náchylná na změnu struktury stránky. Také umožňuje definovat více způsobů jak získat jeden prvek v případě, že se jej nepovede získat prvním nastaveným způsobem (dynamicky se měnící podobné stránky podle obsahu – délka popisu, jeden telefon u kontaktu, ...). Stejně tak byla navrhována s podmínkou nutnosti rozumět stahovaným datům, tudíž ví, co se pod jedním scrapovaným prvkem nachází za data, i když se jedná o tabulku jméno a hodnota. Mám striktně oddělený způsob, jak definovat data pro „index page“ a jak pro vlastní detaily.

Výše popsané aplikace jsou desktopové, na rozdíl od webové v mém podání, tudíž dostupné odkudkoliv přes internet. Pravidelnost vykonávání zajišťují tzv. „crony“, které spouští sám server ve specifikovaném čase. Má aplikace těží z toho, že byla napsána na

konkrétní data a typ serverů, tudíž si poradí se specialitami pro toto odvětví. Na stranu druhou není univerzální; její výstupy se nedají aplikovat na jakýkoliv server a typ dat, aniž by bylo nutné upravit kód aplikace.

ZÁVĚR

Cílem diplomové práce bylo navrhnout a vytvořit aplikaci, která z HTML dokumentu získá všechna relevantní data týkající se nemovitostí. HTML dokumenty jsou získávány z veřejně dostupných dat agregátorů realitních nemovitostí. Získaná data jsou automaticky pomocí slovníků překládána na databázové objekty.

K dosažení tohoto cíle bylo zapotřebí seznámit se s problematikou web scrapingu. Zjistit postupy pro získávání dat z různých prostředí a seznámit se s problémy, do nichž se při použití web scrapingu můžeme dostat.

Web scraping může být použit jako legální cesta, jak se na internetu dostat k informacím všeho druhu, je ovšem důležité zvážit i možné právní důsledky. Existuje mnoho případů, kdy je web scraping považován za nelegální, s možností soudní dohry. Je důležité si uvědomit, že tato technika je nástroj a je na uživateli jak jej použije.

Na základě získaných znalostí jsem navrhl webovou aplikaci s šablonovitým nastavením pro různé agregační servery. Návrh šablon, stejně jako návrh aplikace a uživatelského rozhraní počítá s rozšiřováním samotných šablon, ale i objektů, které jsou následně vyplňovány.

Vytvořenou aplikaci jsem otestoval na dvou agregačních serverech, které čítali přes 100 tisíc nemovitostí každý, ačkoliv tyto nebyly nikdy stáhnuty všechny. Pro testovací účely se scrapuje zhruba 10 tisíc nemovitostí denně ve všech kategoriích aktuální nabídky na všech testovaných serverech. Aplikace byla testována nejen na přesnost stahovaných dat, ale i na překladové schopnosti na univerzální objekty.

Získaná data mají z mého pohledu dobrou úroveň, přestože nejsou 100% identická se zdrojem. Tato úroveň je hodně závislá na kvalitě překladových slovníků, které se za používání aplikace musí teprve vytvořit. Slovníky určují, jak datům rozumíme a jak přesně je opět dokážeme interpretovat zejména při vyhledávání. Aplikaci považuji za funkční a použitelnou, přestože není určena široké veřejnosti a k ostrému nasazení v budoucnu budou potřeba úpravy.

Jedním z úkolů bylo i zjištění možností a realizace filtrace (čištění) získaných dat. Po analýze problému jsem do funkčního řešení zařadil pouze filtraci duplicit prvního druhu (viz. kapitola 10.1), kde lze s určitostí říct, že dané dvě nemovitosti jsou stejné. Další filtrace by se musela řídit procentuální shodou pod 100%, kdy už výsledky mohou být

značně nepřesné. Navíc vyžadují lidský faktor pro kontrolu, zejména při vytváření a optimalizaci algoritmu řešící procentuální shodu subjektů.

Přínos práce vidím v prohloubení znalostí o web scrapingu. Přestože je vnímán vesměs negativně, je nejen ve světě hojně využíván především v žurnalistice, službách, ale i širokou veřejností. Důkazem slouží spousta online (převážně placených) služeb dostupných na internetu. Jako u všech technologií záleží na jejím použití a následném využití takto získaných dat.

ZÁVĚR V ANGLIČTINĚ

The aim of the thesis is to create an application that can extract all relevant facts from HTML documents relating to real estate topic. The HTML documents are gathered from publicly available master data for real estate aggregators. The collected data are then automatically translated to database objects.

To achieve this goal, it was necessary to understand the background of web scrapping, establish procedures for data gathering from multiple sources and comprehend all possible issues that can rise while using the web scrapping technology.

Web scraping can be used as a legitimate way to obtain various data but it's important to be aware of any legal implications. There are many cases where web scraping was considered illegal with possibility of judicial consequences. It's important to understand that this technique is a tool and it depends on the user how the findings are utilized.

Based on the obtained knowledge I have designed a web application with a template for different aggregation servers. The draft template, application, user interface as well as individual objects that will consequently be filled in accounts for a need for expansion.

The final application was tested on two aggregation servers which held over one hundred thousand properties each, although not all data have been downloaded. For testing purposes approximately ten thousand properties are being scrapped daily in all actual tender in all test servers. The application has been tested not only on the precision of downloaded data but also on translation capability to universal objects.

In my opinion the outcome is a good quality even though it is not 100% identical with the original data. This level is heavily dependent on the quality of translation which the application must action first. The dictionaries directs how we understand the data and how precisely these can be interpreted mainly during the search. I consider the application as functional and useful although it is not meant for wider public and will require further adjustments before release in future.

Another task was to find further possibilities and filtration (cleaning) of the outcome data. After analysis of the issue I have added only a duplication filter (chapter 10.01) which resulted in clear understanding that given two properties are the same. Further filter would have to be set below 100% mark which would result in inaccurate data. In addition these will require human intervention for control, mainly in setting up the percentage formulas.

This analysis has helped me in deeper understanding of web scrapping. Although the feeling towards it is mostly negative it's been actively used not just in journalism and services but in wider public as well. The proof lays in plentiful online services (mainly chargeable) that are available on internet. As with all technologies all depends on how the data are used and interpreted.

SEZNAM POUŽITÉ LITERATURY

- [1] YOUNG, Michael J., XML: Krok za krokem. Brno: Computer Press, a. s., 2006. 471 s. ISBN 80-251-1070-2.
- [2] MLÝNKOVÁ, Irena, et al. XML technologie: Principy a aplikace v praxi. Praha: Grada, 2008. 272 s. ISBN 978-80-247-2725-7.
- [3] HEROUT, Pavel. XSLT 2.0 a SVG prakticky. 1. vyd. České Budějovice: Kopp, 2010, 293 s. ISBN 978-80-7232-406-4.
- [4] SKONNARD, Aaron a Martin GUDGIN. XML - pohotová referenční příručka: referenční příručka programátora ke XML, XPath, XSLT, XML Schema, SOAP a dalším. 1 vyd. Praha: Grada, 2006, 342 s. ISBN 80-247-0972-4.
- [5] POYNTER, Ray. The handbook of online and social media research: tools and techniques for market researchers. New York: Wiley, 2010. ISBN 978-0-470-71040-1.
- [6] SCHRENK, Michael. Webbots, spiders, and screen scrapers: a guide to developing Internet agents with PHP/CURL. San Francisco: No Starch Press, c2007. ISBN 15-932-7120-4.
- [7] HOLMES, Lee. Windows PowerShell cookbook. ISBN 978-144-9320-683.
- [8] ERIC VAN DER VLIST. Professional Web 2.0 programming. Indianapolis, IN: Wiley, 2007. ISBN 978-047-0121-054.
- [9] KOUKAL, Pavel a Jan NECKÁŘ. Autorská práva a práva související v daňových souvislostech: ochrana autorských děl, zaměstnanecká díla, licenční smlouvy, databáze, zdaňování příjmů, zdravotní pojištění, pojistné na sociální zabezpečení. 1. vyd. Olomouc: ANAG, 2011, 247 p. ISBN 978-807-2636-877.
- [10] ČERMÁK, Jiří. Internet a autorské právo. 2. aktualizované a rozšířené vyd. Praha: Linde Praha, 2003, 251 p. ISBN 80-720-1423-4.
- [11] MATEJKA, Ján a Jan NECKÁŘ. Internet jako objekt práva: hledání rovnováhy autonomie a soukromí. 1. vyd. Praha: CZ.NIC, 2013, 256 s. CZ.NIC. ISBN 978-809-0424-876.
- [12] SLÍŽEK, David. Tomáš Bleša (Pravednes.cz): Autorské právo? Počkejte, jak s ním zamává 3D tisk: [online]. [cit. 2014-03-18]. Dostupné z:

- <http://www.lupa.cz/clanky/tomas-blesa-pravednes-cz-autorske-pravo-pockejte-jak-s-nim-zamava-3d-tisk/>
- [13] PAUKERTO VÁ, Veronika. Elektronická informační kriminalita: ha. [online]. [cit. 2014-03-18]. Dostupné z: <http://www.ikaros.cz/elektronicka-informacni-kriminalita>
- [14] JANOVS KÝ, Dušan. *Jak psát web* [online]. [cit. 2014-03-22]. Dostupné z: <http://www.jakpsatweb.cz/>
- [15] Client URL Library. *PHP Manual* [online]. [cit. 2014-04-06]. Dostupné z: <http://cz1.php.net/curl>
- [16] PHP Simple HTML DOM Parser. *Sourceforge.net* [online]. [cit. 2014-04-06]. Dostupné z: <http://simplehtmldom.sourceforge.net/>
- [17] BARNETT, Bruce. AWK. [online]. [cit. 2014-04-06]. Dostupné z: <http://www.grymoire.com/Unix/Awk.html>
- [18] Web Basics with LWP. *Perl.com* [online]. [cit. 2014-04-06]. Dostupné z: <http://www.perl.com/pub/2002/08/20/perlandlwp.html>
- [19] Welcome to Scrapy. *Scrapy.com* [online]. [cit. 2014-04-06]. Dostupné z: <http://scrapy.org/>
- [20] Beautiful Soup. *Programovací jazyk Python* [online]. [cit. 2014-04-06]. Dostupné z: <http://www.py.cz/BeautifulSoup>
- [21] WWW::Mechanize. *CPAN* [online]. [cit. 2014-04-06]. Dostupné z: <http://search.cpan.org/~ether/WWW-Mechanize-1.73/lib/WWW/Mechanize.pm>
- [22] Nokogiri. *Nokogiri* [online]. [cit. 2014-04-06]. Dostupné z: <http://nokogiri.org/>
- [23] Html Agility Pack. *CodePlex* [online]. [cit. 2014-04-06]. Dostupné z: <http://htmlagilitypack.codeplex.com/>
- [24] Automated testing that doesn't hurt. *Watir.com* [online]. [cit. 2014-04-06]. Dostupné z: <http://watir.com/>
- [25] Watir WebDriver. *Watir WebDriver* [online]. [cit. 2014-04-06]. Dostupné z: <http://watirwebdriver.com/>
- [26] What is Selenium. *SeleniumHQ* [online]. [cit. 2014-04-06]. Dostupné z: <http://docs.seleniumhq.org/>

- [27] ALBAHARI, Joseph. Threading in C#. *Albahari* [online]. [cit. 2014-04-06]. Dostupné z: <http://www.albahari.com/threading/>
- [28] What is Web Scraping?. *WebHarvy* [online]. [cit. 2014-04-06]. Dostupné z: <https://www.webharvy.com/articles/what-is-web-scraping.html>
- [29] Helium scraper. *Helium scraper* [online]. [cit. 2014-04-06]. Dostupné z: <http://www.heliumscraper.com/en/index.php?p=home>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASF	původně ActiveX Streaming Format, pak Advanced Streaming Format a naposledy Advanced Systems Format
CAPTCHA	Anglická zkratka pro Completely Automated Public Turing test to tell Computers and Humans Apart
CSS	Cascade Style Sheet
CSV	v tomto textu Comma-separated values
DBS	v tomto textu Database system
DOM	Document Object Model
DTD	Document Type Definition
GUI	Graphic User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IE	Internet Explorer
IP	Internet Protocol
RSS	Rich Site Summary
SEO	Search Engine Optimization
SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation

SEZNAM OBRÁZKŮ

Obr. 1: Úryvek zdrojového kódu HTML stránky	17
Obr. 2: Úryvek HTML kódu – méně čitelný (pokud by bylo vypnuté zalamování řádků, je pouze na jednom)	18
Obr. 3: Ukázka zdrojového kódu XML	19
Obr. 4: Ukázka zdrojového kódu pro XSLT.....	20
Obr. 5: Ukázka zdrojového kódu javascriptu	21
Obr. 6: Databázový model aplikace.....	35
Obr. 7: Stránka seznamu nemovitostí s filtrem.....	37
Obr. 8: Detail nemovitosti	37
Obr. 9: Úvodní stránka po přihlášení do administrace	39
Obr. 10: Stránka s filtry	39
Obr. 11: Stránka s výpisem chyb s jednoduchým filtrem.....	40
Obr. 12: Stránka s řešením překladových problémů.....	40
Obr. 13: Stránka s nastavením slovníků	41
Obr. 14: Web Harvy scraper – hlavní stránka	53
Obr. 15: Menu aplikace umožňující nastavení	53
Obr. 16: Nastavení jednoho stahovaného prvku	54
Obr. 17: Helium Scraper – hlavní stránka	55
Obr. 18: Menu aplikace umožňující nastavení	56

SEZNAM TABULEK

Tab. 1: Průměrná doba zpracování nemovitosti v různých prostředích během vývoje 49

SEZNAM PŘÍLOH

PI CD s textem diplomové práce a zdrojovými kódy aplikace