VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

NÁSTROJ PRO AUTOMATICKÉ ZÍSKÁVÁNÍ INFORMACÍ Z WEBU
TOOL FOR AUTOMATIC INFORMATION OBTANING FROM THE WEB

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Jakub Poliak
AUTHOR

VEDOUCÍ PRÁCE — Ing. Lukáš Povoda
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Jakub Poliak ***ID:*** 125290
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Nástroj pro automatické získávání informací z webu

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte nástroj, pomocí kterého bude možné shromáždit kladně a záporně hodnocené příspěvky z webu na základě hodnocení produktů a komentářů uživatelů. Nástroj připravte pro nasazení na serveru, kde bude sbírat a zapisovat komentáře postupně do SQL databáze.

DOPORUČENÁ LITERATURA:

[1] HOUSTON, Pete. Instant jsoup how-to effectively extract and manipulate HTML content with the jsoup library. Online-Ausg. Birmingham [England]: Packt Pub, 2013. ISBN 978-178-2167-990.

[2] DEITEL, Paul J. a Harvey M. DEITEL. Java: how to program. 9th ed. Upper Saddle River, N.J.: Prentice Hall, c2012. ISBN 978-013-2575-669.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 8.6.2017

Vedoucí práce: Ing. Lukáš Povoda
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Bakalářská práce se zabývá vytvořením nástroje pro shromáždění kladných a záporných komentářů z jednoho předního čínského e-shopu do databáze. Ta bude následně využita pro tzv. hluboké učení umělé neuronové sítě, která má rozeznávat pozitivní a negativní význam z textu. Nástroj byl napsán v programovacím jazyce Java s využitím knihoven JSON-simple a jsoup.

KLÍČOVÁ SLOVA

HTML, CSS, e-shop, pavouk, Java, jsoup, JSON

ABSTRACT

This bachelor thesis deals with programming of a tool for collecting positive and negative comments from one of the most popular Chinese e-shop to a database. It will be used for deep learning of an artificial neural network which should distinguish positive text from negative. Application was programmed in Java with the use of JSON-simple and jsoup libraries.

KEYWORDS

HTML, CSS, e-shop, crawler, Java, jsoup, JSON

POLIAK, Jakub. *Nástroj pro automatické získávání informací z webu*. Brno, 2017, 54 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Lukáš Povoda

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Nástroj pro automatické získávání informací z webu“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Velmi rád bych poděkoval vedoucímu mé bakalářské práce Ing. Lukáši Povodovi za podnětné připomínky a odborné rady. Poděkování si jistě zaslouží také moje manželka, rodina a přátelé za jejich trpělivost a podporu.

Brno ....

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM VÝPISŮ

ÚVOD

Umělá neuronová síť je ve výpočetní technice model, který se využívá ke strojovému učení, v počítačové vědě a v dalších výzkumných disciplínách. Je založen na velkém souboru navzájem spojených jednoduchých jednotek, tzv. „umělých neuronů“, které jsou analogické k neuronům v biologickém mozku. Takové systémy mohou být vycvičeny k celé řadě úkolů, jako je například počítačové vidění, rozpoznávání řeči a analýza textových informací, které je obtížné řešit běžným programováním založeným na pravidlech. [6]

Hluboké učení (anglicky deep learning) je postup pro umělé neuronové sítě a příbuzné algoritmy strojového učení, který umožňuje počítačům se samostatně učit bez přímého dozoru a řízení. Aplikace hlubokého učení tedy nejsou programovány, ale jsou cvičeny na velkém objemu dat. Pro textovou analýzu se používají co největší databáze textových informací. Jedním z využití takového systému může být upřednostnění textových zpráv s větší prioritou. Umělá inteligence se musí naučit rozeznat, jestli má text pozitivní nebo negativní význam. [6]

Velké textové databáze s kladnými a zápornými příspěvky však nejsou běžně dostupné. Na diskuzních fórech, v komentářích zpravodajských webů, blogů atp. nejde většinou na první pohled poznat, jestli se jedná o pozitivní nebo negativní komentáře. Je proto potřeba vytvořit databázi o velkém objemu pozitivních a negativních příspěvků, které se nacházejí na nějakém internetovém obchodě. Konkrétně byl vybrán jeden z předních čínských e-shopů, jelikož obsahuje velké množství produktů a navíc je celý v čínštině. Na půdě VUT v Brně se pracuje na neuronové síti, která již má poskytnuty databáze ve španělském, anglickém a českém jazyce. S čínštinou to bude znamenat další krok k univerzálnosti tohoto systému pro rozeznávání kontextu textových zpráv. Aby mohla být použita tak velká databáze, je nutné vytvořit nástroj pro automatické získávání komentářů z webu, což je také hlavním cílem této bakalářské práce.

Internetové obchody (e-shopy) jsou v dnešní době běžnou součástí života moderní populace. Díky nim si může člověk objednat zboží přes internet až z druhého konce planety a přitom si užívat pohodlí domova. Velké e-shopy již nebývají zaměřené pouze na jedno odvětví, nýbrž nabízejí několik kategorií, a to od technických výrobků až po módní doplňky. Nabídka jde ruku v ruce s poptávkou, avšak zboží může zákazník vybírat nejen podle parametrů, barvy, či ceny, ale i podle hodnocení a komentářů vlastníků produktu. Ti vkládají hodnocení na základě svých zkušeností a přímo na stránkách e-shopu popisují své dojmy, které mohou být pozitivní, negativní nebo jen průměrné. Jeden produkt pak může mít stovky, nebo až tisíce komentářů, které mohou usnadnit zájemcům jejich rozhodování.

V současnosti existují nástroje, které procházejí World Wide Web a ze zvolených

webových stránek si stahují potřebné texty, ukládají obrázky nebo dokumenty. Takových nástrojů využívají například internetové vyhledávače. Nástroj vypracovaný v rámci této bakalářské práce funguje na podobném principu. Dle zvolené webové adresy produktu prochází zdrojový kód stránky, ve kterém si vyhledá určitý element. Z daného elementu si následně stáhne záporné i kladné komentáře, které uloží do odpovídající databáze.

V následující kapitole jsou vysvětleny určité teoretické základy webových technologií, které e-shopy používají. Druhá kapitola se věnuje vývojovým nástrojům, které jsou nebo by mohly být použity pro vypracování aplikace. Vlastní návrh řešení je popsán ve třetí kapitole, která představuje také fungování výsledného nástroje pro ukládání komentářů do databáze. V poslední kapitole jsou zhodnoceny dosažené výsledky a uvedeny možné drobné úpravy programu.

1 TEORIE

E-shopy používají podobné technologie jako většina ostatních webů. Pro lepší představu fungování internetových stránek je níže popsán stručný vývoj webu, HTML a JSON, JavaScript a další základní technologie. Tyto znalosti jsou nezbytné pro řešení problému této práce. Tato kapitola je zaměřena i na to, jak fungují programy pro získávání dat z webu a jestli jsou volně dostupné.

1.1 Web

World Wide Web (WWW neboli web) v překladu znamená celosvětová síť a jedná se o informační prostor pro prohlížení, ukládání a odkazování dokumentů a dalších webových zdrojů v prostředí Internetu. Webové stránky uložené na serverech jsou navzájem propojeny pomocí hypertextových odkazů a prohlížíme si je pomocí webových prohlížečů. Odkazy se do prohlížeče vkládají pomocí tzv. URL (Uniform Resource Locator – česky jednotná adresa zdroje). Ta definuje adresu domény serveru, umístění na serveru a protokol, díky kterému je možné přistupovat ke zdroji. [7]

Autorem World Wide Webu je Sir Timothy Berners-Lee, který pracoval jako výzkumný pracovník v Evropské organizaci pro jaderný výzkum CERN. Kolem roku 1980 pracovalo v CERNu přibližně deset tisíc lidí, kteří používali různý hardware a software s individuálními požadavky. Předávání informací probíhalo pomocí elektronické pošty, ale vědci potřebovali sledovat různé věci a také nesouvisející projekty od ostatních kolegů z celého světa. Tim Berners-Lee uvažoval o jednodušším systému pro výměnu informací a dokumentů. V roce 1989 společně s Robertem Cailliauem navrhli vytvoření distribuovaného hypertextového systému a o rok později vytvořili jazyk HTML a protokol HTTP. Následně Berners-Lee napsal také první webový prohlížeč s názvem WorldWideWeb, který byl určený speciálně pro počítač firmy NeXt. [7, 1]

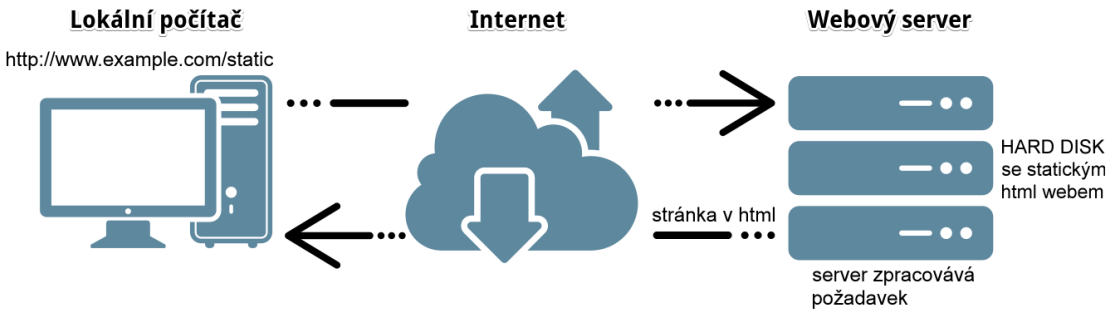
1.1.1 URL

URL bývá neformálně označována jako webová adresa. Znamená však jednoznačné umístění nějakého zdroje nebo dokumentu v síti, ke kterému je možné přistupovat. [35] Typická adresa URL má tvar `https://www.vutbr.cz/o-univerzite`, kde `https` značí protokol, `www` je doménou třetího řádu, `vutbr` je doménou druhého řádu, `cz` je generickou doménou (nejvyššího řádu) a `o-univerzite` je cesta ke stránce. Dalšími parametry ještě mohou být kotva, formulářové metody nebo server. [20, 35]

Pomocí URL lze zadat i přihlašovací informace oddělené od sebe navzájem dvojtečkou a od domény zavináčem, například: `https://jmeno:heslo@www.domena.com/`.

1.1.2 Statické stránky

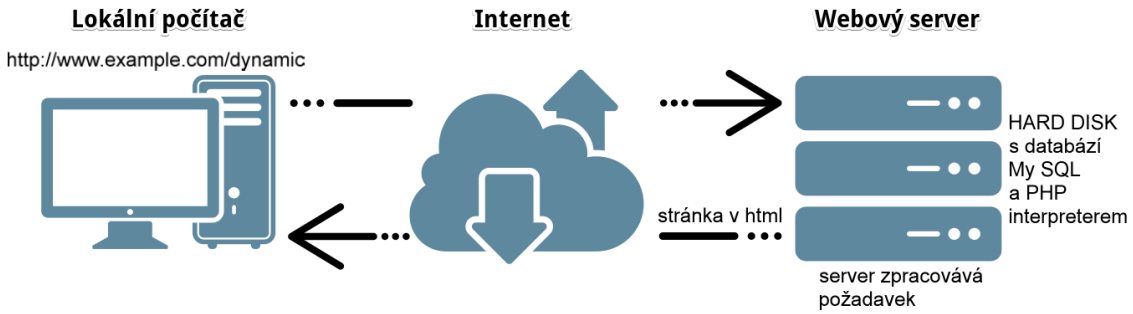
Statická webová stránka je webovým prohlížečem zobrazena přesně tak, jak je uložena na serveru. Zobrazuje se tedy pro všechny uživatele stejně, s výhradou moderních webových serverů, které dokáží zprostředkovat žádaný obsah nebo jazyk dokumentu, pokud jsou k tomu nakonfigurovány. Statické webové stránky jsou často HTML dokumenty uložené v souborovém systému na serveru, viz obr. 1.1. Jsou vhodné pro obsah, který potřebuje být zřídkakdy aktualizován. Příkladem může být informativní webová stránka pro nějakou službu, např. kadeřnictví nebo obchod, které obsahují pouze kontakt a fotogalerii. [7]



Obr. 1.1: Doručení statické webové stránky

1.1.3 Dynamické stránky

Zobrazení dynamické webové stránky je řízeno aplikačními skripty na straně serveru. [7] Oproti statickým stránkám se obsah dynamických stránek mění v závislosti na čase (např. blog), kontextu (např. přizpůsobení parametrů), uživateli (např. po přihlášení uživatele), uživatelské interakci (např. komunikace na sociálních sítích) nebo kombinaci předchozích.



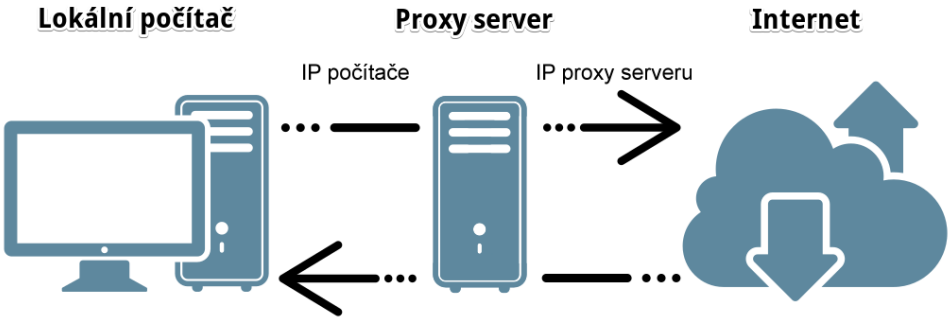
Obr. 1.2: Doručení dynamické webové stránky pomocí PHP a MySQL

Dynamické webové stránky využívají tři typy zobrazení obsahu. Prvním typem je skriptování a vytváření obsahu na straně uživatele, kdy příslušný kód webové stránky reaguje na interakci uživatele pomocí klávesnice nebo myši. Skriptovacím jazykem na straně uživatele je například JavaSript, JScript nebo ActionScript. Druhým typem je skriptování a vytváření obsahu stránek na straně serveru, kdy se nejdříve sestaví obsah na straně serveru a poté se zobrazí uživateli, jak je znázorněno na obr. 1.2. Takové stránky využívají jazyky jako například PHP nebo Perl. Posledním typem jsou kombinace stran klienta a serveru, které využívají technologii AJAX. [7, 12]

1.1.4 Proxy server

V počítačových sítích slouží proxy server (počítačový systém nebo aplikace) jako prostředník pro požadavky klientů, kteří hledají zdroje z různých serverů. Klient se připojí k proxy serveru a požaduje nějakou službu, např. soubor, webovou stránku, připojení nebo nějaký dostupný zdroj z jiného serveru. Proxy server zhodnotí požadavek, který následně předá na vzdálený server, jako by byl sám klient. Jakmile dostane odpověď od cílového serveru, přepošle ji klientovi (viz obr. 1.3). [21]

Původní uživatel je tedy cílovému www serveru skrytý. Nejčastějším využitím proxy serveru je tedy poskytnutí anonymity nebo např. prohlížení blokovaných stránek pro danou zemi atd.



Obr. 1.3: Surfování po webu přes proxy server

1.2 Protokol HTTP

Internetový protokol HTTP (Hypertext Transfer Protocol) je základem datové komunikace pro World Wide Web. [1] Hypertext je strukturovaný text, ve kterém fungují některé výrazy jako tzv. hypertextové odkazy (hyperlinky). Těmito odkazy se dá přenést na určitou

část dokumentu nebo na dokument úplně jiný. HTTP je internetovým (ASCII orientovaným) aplikačním protokolem pro výměnu nebo přenos hypertextu mezi WWW serverem a klientem. [7]

O vývoj HTTP se zasloužil Tim Berners-Lee, který definoval HTTP v roce 1989. Vývoj standardů byl koordinován společnostmi Internet Engineering Task Force (IETF) a World Wide Web Consortium (W3C) a jeho výstupem byly série publikací Requests for Comments (RFC). První definice verze HTTP/1.1 byla vytvořena v roce 1997 – RFC 2068 (viz [15]) a později vylepšena v roce 1999 - RFC 2616 (viz [16]). Nejnovější verze HTTP/2 - RFC 7540 (viz [17]) byla standardizována v roce 2015 a v současné době je podporována hlavními webovými servery. [20]

1.2.1 Funkce HTTP

Protokol HTTP definuje přenášený tvar dat a také podobu dotazu a odpovědi v modelu klient-server. Klientem může být například webový prohlížeč, který zobrazuje webovou stránku běžící na hostingu serveru. Server používá standardně TCP port 80, případně i port 8080. Síťový port je číslo, které slouží pro komunikaci a rozlišení aplikace v počítači v rámci počítačové sítě. HTTP server naslouchá na portu a čeká na zprávu o požadavku od klienta. Požadavek je ve formě textu s označením požadovaného dokumentu, informace o prohlížeči a podobně. Po obdržení žádosti server pošle zpět odpověď o stavu (např. „HTTP/1.1 OK“) a vlastní zprávu. V těle zprávy je typicky požadovaný výsledek, ale mohou být vrácena i chybová hlášení či jiné informace. [20]

Od verze HTTP/1.1 je možné během jednoho spojení přenést i více objektů. Po určité době nečinnosti je toto spojení serverem ukončeno. Protokol HTTP umí kromě HTML stránek a elektronické pošty přeposílat i jakýkoli jiný soubor, přičemž využívá také další protokoly, jako je např. FTP nebo SMTP. HTTPS je nadstavbou protokolu HTTP, který využívá asymetrické šifrování pro zabezpečení spojení mezi klientem a serverem před odposloucháváním. Standardním portem HTTPS je 443 a data jsou šifrována pomocí protokolu SSL (Secure Sockets Layer) nebo TLS (Transport Layer Security). [20]

1.2.2 Protokol HTTP/2

Protokol HTTP/2 (původně pojmenovaný HTTP/2.0) je optimalizovaná hlavní verze protokolu HTTP. Byl vytvořen z dřívějšího experimentálního protokolu SPDY, který byl původně vyvinutý společností Google. Podporuje všechny základní funkce HTTP/1.1 a má za cíl být efektivnější. [17]

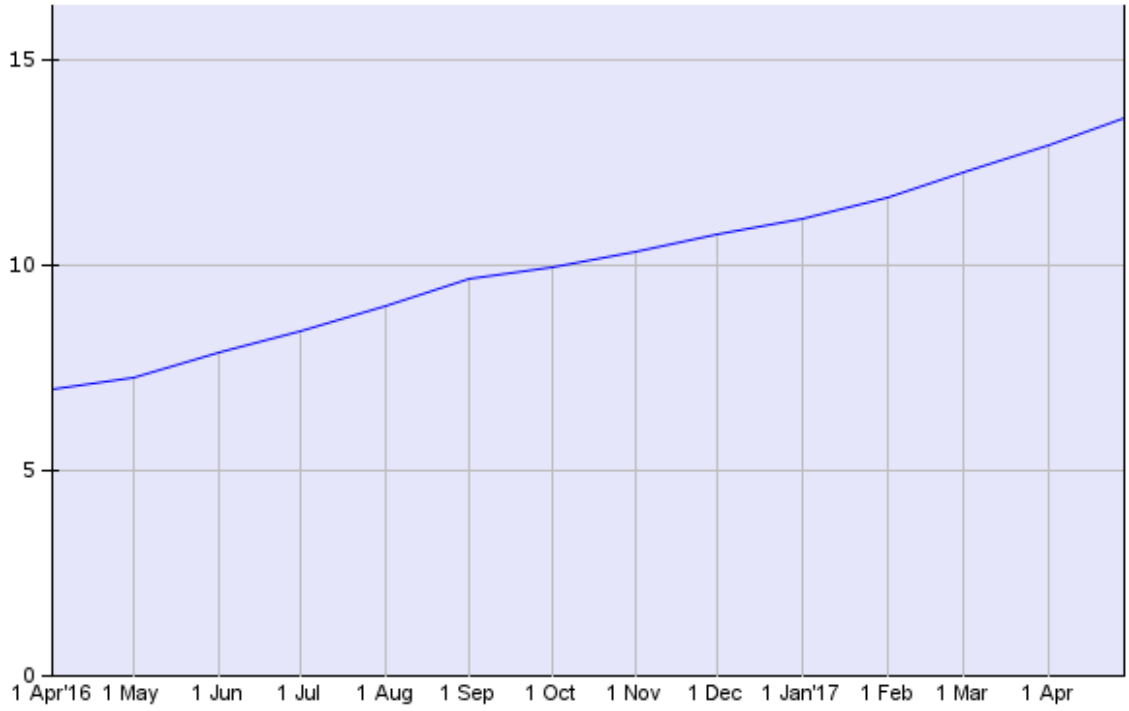
HTTP/2 umožňuje účinnější využívání síťových prostředků a snížené vnímání latence tím, že zavádí kompresi pole záhlaví a umožňuje více souběžných výměn požadavků a odpovědí na stejném připojení. Multiplexování požadavků je dosaženo tím, že každá výměna požadavků/odpovědí HTTP je spojena s vlastním datovým proudem. Řízení toku a stanovení priorit zajistí, aby byla přenášena pouze data, která mohou být přijímačem používána.

Stanovení priorit udává, že omezené zdroje mohou být nejprve směrovány na nejdůležitější proudy. HTTP/2 přidává nový režim interakcí, díky němuž server může posílat reakce na klienta, pokud server předpokládá, že je klient bude potřebovat. [17]

Vzhledem k tomu, že pole záhlaví protokolu HTTP používaná ve spojení mohou mít velké množství redundantních dat, rámce, které je obsahují, jsou komprimovány. To je obzvlášť výhodné na dopad velikosti požadavků, jelikož mnoho požadavků může být komprimováno do jediného paketu. [17]

Většina hlavních prohlížečů (Firefox, Chrome, Safari, Internet Explorer 11 a další) přidala podporu HTTP/2 během roku 2015. Tento protokol využívají nejpopulárnější webové servery jako je: Google.com, Youtube.com, Facebook.com, Wikipedia.org, Yahoo.com, Twitter.com atd.

Ke konci dubna 2017 používalo HTTP/2 více jak 13,6 % ze všech webů. [36] Vývoj je znázorněn na obr. 1.4.



Obr. 1.4: Používání HTTP/2 na webu, 29. dubna 2017, W3Techs.com [36]

1.2.3 Metody GET a POST

HTTP definuje metody (`method`), které označují požadovanou akci, která má být provedena na identifikovaném zdroji. Zdroj často odpovídá souboru nebo výstupu souboru (dokumentu) umístěnému na serveru. Specifikace HTTP/1.0 definovala metody `GET, POST` a `HEAD` a specifikace HTTP/1.1 přidala pět nových metod: `OPTIONS, PUT, DELETE, TRACE`

a `CONNECT`. Každý klient může použít jakoukoliv metodu a server může být nakonfigurován tak, aby podporoval libovolnou kombinaci těchto metod. [7]

Metoda `GET` požaduje reprezentaci zadaného zdroje. Požadavky zadané pomocí `GET` by měly pouze načíst data (zobrazení webových stránek, RSS kanálů aj.) a neměly by mít žádný jiný účinek. Jedná se o nejpoužívanější z metod. [7]

Dotazovací metoda `POST` slouží pro získání webové stránky (či nějakého objektu, např. obrázku) a navíc umožňuje předat serveru metaproměnné, které může server dále zpracovat. Data `POST` mohou být např. anotace pro stávající zdroje, zpráva pro diskuzní skupinu, e-mail nebo poznámka, blok dat, který je výsledkem odeslání webového formuláře, nebo položka, kterou chceme přidat do databáze. [7]

1.3 Příklady značkovacích jazyků

Značky slouží k oddělení obsahu dokumentu od jeho formy, pokud je to možné. Účel značek je nezávislý na značkovacím jazyku (anglicky Markup Language, ML). Metajazyk (například SGML nebo XML) je jazyk, který slouží pro popis značkovacích jazyků. [28]

Používají se dva typy značek:

- Logické (strukturní) značky – slouží k definici struktury dokumentu a k přiřazení logickému významu části dokumentu, například nadpis, citovaný odstavec, běžný text atd. Logické značky neovlivňují vzhled dokumentu.
- Vizuální značky – formátují a ovlivňují vzhled dokumentu, například tučný text, velikost textu, kurzíva atd. Vizuální značky mohou být i místa, která nelze vidět, jako bílá místa, odsazení textu a podobně.

1.3.1 SGML

Standardní jazyk pro popis značkovacích programů (Standard Generalized Markup Language neboli SGML) je metajazyk, který se vyvíjel poměrně dlouho. V roce 1986 se stal standardem (ISO 8879). Je navržený tak, aby byl složitý, ale univerzální, kvalitní a rozšiřitelný. Díky velké míře volnosti, umožňuje definovat další značkovací jazyky, jako jeho podmnožiny. Takhle vznikl například jazyk XML, HTML a další. SGML se používá ve velkých společnostech, v elektronickém průmyslu, v nakladatelstvích a podobně. [28]

Dokument SGML se skládá ze tří částí:

1. Deklarace (pokud není uvedena, použije se standardní): pravidla pro pojmenovávání, rezervovaná jména, role oddělovačů atd.
   Když tedy v HTML napíšeme odstavec ve tvaru `<p> text </p>`, po použití SGML deklarace dle ukázky 1.1 může být zapsán jako `!!p?? text @@p!!`.
2. Definice typu dokumentu neboli DTD: elementy (vytvářející stromovou strukturu) s jejich atributy a entitami mohou být součástí dokumentu nebo v externím souboru.
3. Označkovaný text: značky korespondují s deklarací a definicí typu dokumentu.

Výpis 1.1: Příklad deklarace v SGML

```
GENERAL
  SGMLREF
  STAGO "!!"
  ETAGO "@@"
  TAGC  "??"
```

1.3.2 XML

Rozšiřitelný značkovací jazyk (eXtensible Markup Language neboli XML) je rovněž meta-jazyk a jde o podmnožinu jazyka SGML. Jedná se vlastně o zjednodušenou verzi SGML, díky čemuž čelí velké popularitě. Organizace W3C definovala XML verzi 1.0 v roce 1998. Ta se v dnešní době používá mnohem více než její novější pátá revize 1.1 z roku 2006. [28]

XML se používá pro serializaci dat, transformování do jiného typu dokumentu nebo do jiné XML aplikace a pro publikování dokumentů. Původním účelem byl hlavně přenos elektronických dokumentů přes web. Na XML jsou založeny další jazyky, jako je například RSS, XHTML, SVG, XPS, MusicXML a další. [28] Příklad textové části XML dokumentu bez DTD je zobrazen na ukázce 1.2

Deklarace XML dokumentu je nepovinná a uvádí se formou první značky v souboru: `<?xml version="číslo verze" encoding="kódování" standalone="yes|no"?>`

Výpis 1.2: Příklad textové části XML dokumentu bez DTD

```
<?xml version="1.0" encoding="utf-8"?>
<rodina>
  <osoba pohlavi="zena" vek="25">
    <jmeno>Jane</jmeno>
    <prijmeni>Doe</prijmeni>
  </osoba>
  <osoba pohlavi="muz" vek="27">
    <jmeno>John</jmeno>
    <prijmeni>Doe</prijmeni>
  </osoba>
</rodina>
```

1.4 HTML a CSS

Jazyk HTML (HyperText Markup Language, česky hypertextový značkovací jazyk) slouží pro vývoj webových stránek a aplikací. Spolu s CSS styly tvoří základní prvek v systému World Wide Web pro publikování webových stránek a dokumentů v prostředí Internetu. Pro prohlížení webových stránek slouží webový prohlížeč, který umožňuje zobrazit obsah na obrazovce počítače, či jiném zařízení. [7]

Tento jazyk vznikl kvůli potřebě formátovat a sdílet vědecké materiály pro výzkumníky v CERNu. Pro tvorbu dokumentů se v té době používaly složité jazyky PostScript, SGML a TeX. Tim Berners-Lee požadoval něco jednoduššího, proto navrhl jazyk HTML, který je charakterizován množinou tagů (značek) a jejich atributů (vlastností). Tento způsob zápisu byl převzat z jazyka SGML (Standard Generalized Markup Language). [7, 2]

HTML se stal velmi oblíbeným jazykem a díky rychlému rozvoji webu bylo nutné definovat jeho standardy. Berners-Lee založil roku 1994 World Wide Web Consortium (W3C), které je zodpovědné za sjednocení webových standardů pro World Wide Web, aby bylo využito jeho plného potenciálu. Standard pro verzi HTML5 vznikl po patnáctileté odmlce v roce 2014. Tato specifikace přidala podporu mnohých moderních technologií a opravuje velké množství chyb z předešlé verze. Největší změnou HTML5 je, že kromě vylepšených elementů starších verzí přináší i rozhraní API (aplikační programové rozhraní) pro vývoj webových aplikací. Například je možné vytvořit offline aplikaci uloženou lokálně v uživatelově počítači, která ale funguje ve webovém prohlížeči. V HTML5 můžeme dokonce kreslit nebo vkládat audio a video záznamy do dokumentů. Poslední doporučení W3C pro HTML bylo vydáno 1. listopadu 2016 s názvem HTML 5.1. [7]

Kromě jazyka HTML existuje i jazyk XHTML, kde počáteční X je zkratkou pro eXtensible (česky rozšiřitelný). XHTML je podporován většinou moderních prohlížečů, ale jedná se v podstatě o stejný jazyk jako HTML. Nepřináší žádné nové možnosti, spíše omezení. [14]

1.4.1 Struktura jazyka HTML

Aby byl element rozeznán od ostatního textu, bývá ohraničen mezi úhlové závorky `<` a `>`. Konkrétní elementy v textu jsou pojmenovány tagy. Mezi tagy je umístěn požadovaný text. [7]

Některé značky jsou samostatné, jako např. vnořený tag `<br>`, který slouží pro odsazení textu na nový řádek. Většinu však tvoří značky párové s otevírací a ukončovací značkou pro oddělení nějakého obsahu. Ukončovací tag se pozná podle lomítka za úhlovou závorkou. Příklad bloku textu s párovým a nepárovým tagem, jehož výsledkem je odstavec s dvěma řádky textu, je uveden v ukázce 1.3.

Ke značkám je však možné přidat i další vlastnosti. Lze tedy například vytvořit hypertextový odkaz, který se po rozkliknutí otevře v novém okně prohlížeče (viz příklad 1.3).

Nejjednodušší cesta, jak psát kód v HTML, je například pomocí notepadu v operačním systému Windows. Soubor musí být uložený jako `.html` a musí obsahovat základní strukturu z párových tagů: `html, head` a `body`. Obsah webu se pak nachází v těle stránky, tedy mezi tagy `<body> </body>`, jako na poslední ukázce v příkladu 1.3. [9]

Výpis 1.3: Struktura jazyka HTML

```
<!-- Párový a nepárový tag - odstavec s dvěma řádky -->
<p> Následující text <br> bude na dalším řádku. </p>

<!-- Hypertextový odkaz - otevře se vnovém okně -->
<a href=http://www.vutbr.cz target="\_blank">
  Odkaz na VUT
</a>

<!-- Základní struktura souboru *.html -->
<html>
  <head>
    <title> Ukázková stránka </title>
  </head>
  <body>
    Vítejte na mé první stránce!
  </body>
</html>
```

1.4.2 CSS

Zatímco HTML je základem pro definování obsahu webových stránek, CSS (Cascading Style Sheets, česky kaskádové styly) specifikuje, jak tyto stránky budou vypadat. Díky kaskádovým stylům můžeme definovat například vlastnosti písma, textu, barvy textu a pozadí, způsoby zobrazení nebo také umisťování elementů. [8] Tento jazyk navrhl v roce 1994 Håkon Wium Lie a verze CSS1 byla oficiálně vydána společností W3C v roce 1996. [31]

Šablona kaskádových stylů je obyčejný textový soubor s koncovkou `.css` a bývá k HTML připojen určením cesty ve značce `link` v hlavičce HTML kódu. Druhým způsobem je vnoření CSS přímo do HTML elementu pomocí atributu `style`. Obě možnosti lze porovnat na ukázce 1.4

Výpis 1.4: Připojení CSS k HTML

```
<!-- Připojení v hlavičce HTML -->
<link rel="stylesheet" type="text/css"
href="css/main-style.css">

<!-- Vnoření CSS do HTML -->
<div style="color:blue;text-decoration:underline">
  Tento text je modrý a podtržený.
</div>
```

Soubor CSS obsahuje jedno nebo více pravidel, která definují, jak by se měly určité elementy zobrazovat, například modré pozadí stránky nebo text zarovnaný do bloku. Každé pravidlo obsahuje dvě hlavní části:

- selektor, jenž určuje, jaké elementy budou mít aplikované pravidlo,
- deklarační blok, který se skládá z deklarací, z nichž každá obsahuje jednu nebo více dvojic vlastnost-hodnota.

Na příkladu 1.5 je zobrazena struktura stylu, kde h1 je selektor a mezi složenými závorkami se nachází deklarační blok s vlastností (color = barva) a hodnotou (blue = modrá):

Výpis 1.5: Struktura stylu CSS

```
1 h1 {
2   color: blue;
3 }
```

V současné době je nejlépe podporovanou verzí CSS2, avšak moderní prohlížeče používají již CSS3, který je ještě ve fázi specifikace. Verze CSS3 obsahuje oproti starším verzím hlavně vylepšení vizuálních efektů, jako například barevné přechody, zaoblené rohy, stíny textu, průhlednost, animace atd. [8]

1.5 JavaScript

JavaScript byl vytvořen v květnu roku 1995 Brendanem Eichem, který tou dobou pracoval ve společnosti Netscape. Po obdržení licence na používání ochranné známky od společnosti Sun dostal jazyk název JavaScript. To byl marketingový tah, jelikož v té době byl velmi populární programovací jazyk Java. Nemohou být ale vůbec srovnávány, jelikož se jedná o dva naprosto odlišné jazyky. [5]

JavaScript je natolik rozšířený, že se jako jediný nachází téměř na všech počítačích po celém světě. Jedná se o skriptovací jazyk, který ve spojení s HTML a CSS tvoří základ pro dynamické webové aplikace. Všechny moderní webové prohlížeče jej implementují, jelikož je JavaScript používán na většině webových stránek.

1.6 JSON

JavaScript Object Notation (česky JavaScriptový objektový zápis) je odlehčený formát pro výměnu dat. [18] Pro člověka je snadné jej číst a psát a pro stroje je snazší jej generovat a analyzovat. Je založen na podmnožině programovacího jazyka JavaScript a vznikl na počátku roku 2000. Oficiálními standardy JSONu jsou ECMA-404 z roku 2013 a RFC 7159 z roku 2014. [34, 10]

JSON je textový formát, který je zcela nezávislý na jazyce, ale používá konvence, které jsou programátorům známé z jazyků C, C++, C#, Java, JavaScript, Perl, Python,

PHP5 a mnoha dalších. [19] Tyto vlastnosti dělají z JSONu ideální jazyk pro výměnu dat. Nejčastěji se používá jako datový formát pro asynchronní komunikaci webového prohlížeče a serveru. Do značné míry nahrazuje značkovací jazyk XML a dokonce i YAML, který převádí objekty (libovolně složité) do sériové podoby. [19]

Výpis 1.6: JSON formát k popisu osoby

```
{
  "jmeno": "Jane",
  "prijmeni": "Doe",
  "vek": 25,
  "jeNazivu": true,
  "zamestnani": "programmer",
  "adresa": {
    "ulice": "20␣East␣1st␣Street",
    "mesto": "New␣York",
    "stat": "New␣York",
    "zipCode": "10003"
  },
  "telefonniCisla": [
    {
      "type": "home",
      "number": "215␣555-5678"
    },
    {
      "typ": "soukromy",
      "cislo": "325␣567-1435"
    }
  ],
  "stav": vdana,
  "deti": []
}
```

JSON je postaven ze dvou struktur: [18]

- Kolekce dvojic název-hodnota: V různých jazycích je realizována jako objekt (object), záznam (record), slovník (dictionary), hashovací tabulka (hash table), struktura (struct), klíčový seznam (keyed list) nebo asociativní pole (associative array).
- Seřazený seznam hodnot: Ve většině jazyků je realizován jako pole (array), vektor (vector), seznam (list) nebo sekvence (sequence).

Jedná se o univerzální datové struktury. Prakticky všechny moderní programovací jazyky v nějaké formě podporují tu či onu podobu. Proto byl JSON založen tak, aby měl formát nezávislý na jazyce. [18] JSON sice pochází z JavaScriptu, ale i mnoho jiných

programovacích jazyků používá tento formát ke generování a stahování dat. JSON používá pro názvy souborů koncovku `.json`. [19] Tyto soubory mohou být přenášeny v prostředí intranetu nebo internetu - například díky technologii AJAX.

V JSONu se používají následující formáty: [10]

- Objekt: množina párů název-hodnota. Objekt začíná a končí složenými závorkami `{}`. Za každým názvem následuje dvojtečka a páry jsou od sebe odděleny čárkou.
- Pole: seřazený soubor hodnot. Začíná a končí hranatými závorkami `[]`, hodnoty jsou od sebe odděleny čárkou.
- Hodnota: může být řetězec (string) v uvozovkách, číslo, boolean (pravda = true nebo nepravda = false), prázdná hodnota (null), objekt nebo pole. Tyto struktury mohou být vnořené.
- Řetězec: posloupnost nula a více znaků Unicode v uvozovkách. Lze využít únikové sekvence (escape sequence) pomocí zpětného lomítka. Řetězec je velmi podobný řetězci v programovacím jazyce C nebo Java.
- Číslo: rovněž podobné číslu v jazyce C nebo Java, pouze nevyužívá formáty v osmičkové a šestnáctkové soustavě.

Na ukázce 1.6 je uveden vzorový popis osoby. Mezi jednotlivé prvky a hodnoty se mohou vkládat bílé znaky (whitespace), které ale ve výsledku nic nemění. [10] JSON neřeší kódování, avšak výchozím kódováním je UTF-8. Na rozdíl od formátu YAML neposkytuje žádnou syntaxi pro komentáře.

1.7 AJAX

AJAX je zkratkou pro Asynchronous JavaScript And XML. [12] XML se v názvu vyskytuje, protože je to jeden z formátů, který AJAX umí zpracovat. Pokud jsou data JSON, někdy se označuje jako AJAJ, ale rozšířenější je zkratka AJAX.

Jedná se o soubor vývojových technik na straně klienta pro vytváření asynchronních webových aplikací. Tyto aplikace dokáží posílat a načítat data ze serveru na pozadí (asynchronně), aniž by zasahovaly do stávající stránky. Data jsou oddělena od prezenční vrstvy a AJAX může měnit obsah dynamicky, aniž by bylo nutné znovu načíst celou webovou stránku. [12]

Mezi formáty dat používané AJAXem patří:

- HTML (nebo XHTML) a CSS pro prezentaci,
- DOM pro dynamické zobrazení a interakce s daty,
- JSON nebo XML pro výměnu dat,
- XMLHttpRequest objekt pro asynchronní komunikaci,
- JavaScript ke vzájemné spolupráci zmíněných technologií.

1.8 Pavouk

Webový pavouk (anglicky crawler) je internetový bot (počítačový program), který systematicky prochází World Wide Web a pomocí protokolu HTTP si ukládá důležitá data webových stránek, jako například jejich obsah, metadata, případně i zpětné odkazy. Rovněž si ukládá hypertextové odkazy, ze kterých si vytváří seznam URL adres pro rychlejší vyhledávání. Podobně pracuje i tzv. linkchecker, který prochází množinu stránek a detekuje odkazy na již neexistující weby. Pavouk může ověřit hypertextové odkazy, HTML kód a může být také použit pro získávání dat z webu. [32]

Webové fulltextové vyhledávače, jako například Seznam.cz nebo Google, jsou schopny projít velké množství stránek v reálném čase. Principiálně fungují po zadání dotazu ve třech krocích: [32]

- sbírání dat vyhledávačem data do databáze,
- indexování,
- výpis výsledků.

1.9 Získávání dat z webových stránek

Získávání dat z webových stránek (web scraping, web harvesting či web data extraction) je proces, který umožňuje rozebrat (parsovat) určité množství dat na webu k získání požadované informace. [29] Může být provedeno manuálně uživatelem, ale také některým z nástrojů bot nebo pavouk. Aplikace pro získávání dat přistupuje k WWW přímo pomocí HTTP nebo prostřednictvím webového prohlížeče. Tímto způsobem se dají stáhnout například obrázky určité velikosti, konkrétní textová část webu (jména, telefonní čísla, adresy), specifická metadata apod. Výsledky mohou být ukládány do lokální databáze, do textového souboru nebo do tabulkového procesoru zpravidla ve formátu `*.csv` pro pozdější použití nebo analýzu. Aby stahování informací nebylo trestné, je nutné dodržovat podmínky používání a autorských práv použitých webových stránek. [29]

Získávání dat se používá jako součást aplikací pro webové indexování, pro získávání kontaktů, dolování dat (data mining), online sledování cen a srovnání cen, vyhledávání recenzí produktů, monitorování údajů o počasí, detekci změny obsahu webových stránek, výzkum apod.

Webové stránky jsou postaveny na značkovacích jazycích (HTML a XHTML) a často obsahují velké množství užitečných dat v textové podobě. Nicméně většina webových stránek je určena pro koncové uživatele a ne pro automatizované použití. Z tohoto důvodu byly vytvořeny sady nástrojů, které procházejí datový obsah. Společnosti jako je Amazon a Google poskytují nástroje a služby pro získávání dat z webu volně koncovým uživatelům. [29]

Novější formy zahrnují poslech zdroje dat webových serverů. Například JSON je běžně používaným mechanismem mezi klientem a webovým serverem. Některé webové stránky

používají metody k zabránění stahování jejich dat, jako je například detekce a zákaz botům v prohlížení stránky.

V dalších podkapitolách jsou popsány některé nástroje pro získávání dat z internetových stránek.

1.9.1 Googlebot

Googlebot je pavouk (webový bot) od společnosti Google. Jedná se zřejmě o jeden z nejpopulárnějších webových pavouků na internetu. Procházení (anglicky crawling) je proces, kterým Googlebot zjistí nové a aktualizované stránky, které jsou přidány do indexu Google. Používá algoritmický proces, kde počítačové programy určují, které stránky se budou procházet, jak často a kolik stránek bude načítáno z každého webu. Googlebot začíná seznamem adres URL webových stránek generovaných z předchozích procesů procházení rozšířeným o data souborů Sitemap poskytnutá webmastery. Googlebot navštíví každou z těchto webových stránek, zjistí na nich odkazy (src a href) a přidá je do svého seznamu procházených stránek. Nové weby, změny stávajících stránek a mrtvé odkazy jsou zaznamenány a použity k aktualizaci Google indexu. [11]

Googlebot nevidí kompletní webové stránky, vidí pouze jednotlivé komponenty dané stránky. Pokud některá komponenta (HTML, CSS, JavaScript, obrázek) není pro Googlebot přístupná, nebude je posílat do indexu Googlu.

Existuje devět různých typů Googlebotu: [33]

- Googlebot (Google – hlavní webový vyhledávač),
- Google Smartphone,
- Google Mobile,
- Google Adsense,
- Google Mobile Adsense,
- Google Adsbot,
- Googlebot Images,
- Googlebot News,
- Googlebot Video.

1.9.2 Advanced Web Scraper

Jedná se o jednoduchý a výkonný nástroj využívající CSS selektory pro vytváření agentů pro získávání dat z webových stránek. Pro své fungování používá JavaScript, pracuje na cloudu a umožňuje extrahovat data z webových stránek, AJAXu, XML, JSONu a dalších. Pomocí point-and-click CSS selektorů se vybere náhled, který se může dále uložit do JSON, CSV nebo TSV dat. [3]

Tato aplikace je zdarma a funguje jako rozšíření v internetovém prohlížeči Chrome. Je možné ji použít pro procházení webových stránek, stahování dat z webu nebo e-mailu, extrahování milionů webových stránek atd.

1.9.3 Apifier

Apifier je cloudově založený pavouk, který slouží k extrahování strukturovaných dat z jakékoliv webové stránky nebo k provádění automatických akcí na webu pomocí JavaScriptu. Apifier umožňuje vývojářům proměnit jakoukoliv webovou stránku do API, díky čemuž mohou rychle vytvářet aplikace využívající data z webů třetích stran. [4]

Umožňuje například ukládat nové články z informačních webů jako RSS čtečka, nebo monitorovat produkty každou minutu na stovkách internetových obchodů a podávat notifikace např. pokud se daný produkt naskladní nebo zlevní.

1.9.4 Norconex HTTP Collector

Tento open-source pavouk sbírá obsah webových stránek nebo jakýkoli jiný repozitář dat. Může se spouštět samostatně nebo se může přidat do jiných aplikací. Pracuje na libovolném operačním systému. [27]

Funkce Norconex HTTP Collectoru:

- může procházet miliony dat z jednoho serveru,
- extrahuje text z různých formátů souborů (HTML, PDF, Word atd.),
- extrahuje metadata dokumentů,
- podporuje stránky vykreslené pomocí JavaScriptu,
- detekuje jazyk a podporuje překlad,
- normalizuje adresy URL,
- detekuje upravené a smazané dokumenty,
- a mnoho dalšího.

2 VÝVOJOVÉ NÁSTROJE

Tato kapitola popisuje vývojové nástroje, které používají pro automatické získávání informací z webu různé programy včetně toho, kterým se zabývá tato práce. Vypsané nástroje jsou jednoduché a výsledná aplikace může být díky nim modifikovaná pro další použití, nejen pro stahování komentářů.

2.1 Nástroj jsoup

Mnoho webových stránek a služeb poskytuje informace prostřednictvím RSS (Rich Site Summary) kanálů pro čtení novinek na webu, nebo dokonce prostřednictvím API webových služeb. [13] Nicméně spousta webů neposkytuje takové možnosti. To je důvod, proč vzniká mnoho HTML aplikací pro využití při získávání dat z webových stránek.

Jsoup je Java knihovna pro načítání, vyhledávání a manipulaci s HTML dokumenty. To poskytuje velmi pohodlné API pro extrakci a manipulaci s daty s využitím DOM (objektový model dokumentu), CSS, jQuery a podobných metod. Jsoup implementuje specifikaci WHATWG HTML5 a převádí HTML do stejného DOM jako moderní prohlížeče. WHATWG je rostoucí komunita lidí zaměřená na vývoj HTML a API potřebné pro webové aplikace. Knihovna jsoup je navržena tak, aby uměla pracovat se všemi verzemi HTML. Jsoup je volně přístupný a lze jej dále modifikovat a zlepšovat díky svobodné MIT licenci. [23]

Knihovna jsoup může být použita pro: [23]

- získávání HTML dokumentů z internetu pomocí URL, ze souboru uloženého na disku nebo z proměnné (řetězec string),
- extrahování dat použitím DOM a CSS selektorů,
- manipulaci s HTML elementy, atributy a textem,
- výpis všech obrázků a odkazů,
- transformování a čištění HTML kódu pro požadované zobrazení.

Díky jsoup je možné extrahovat cokoliv, co je staticky vykresleno na stránce na straně serveru, ale nikoli pomocí JavaScriptu, což je hlavní nevýhodou této knihovny. [23]

2.2 Selenium

Selenium je nástrojem pro automatizaci webových prohlížečů a aplikací pro testovací účely. Je vyvinutý v programovacím jazyce Java, díky němuž je možné jej používat na různých platformách. Je podporován většinou prohlížečů a různými programovacími jazyky jako např. Java, C#, PHP atd. Selenium je technologií v bezpočtu automatizačních nástrojů, API a frameworků. [30] Skládá se z několika navzájem spolupracujících komponent, které ve výsledku tvoří univerzální testovací systém.

Selenium projekty: [30]

- Selenium WebDriver – může řídit prohlížeč lokálně nebo na vzdálených serverech.

- Selenium Grid – umožňuje spouštět automatizační testy na mnoha serverech najednou, což zkracuje čas potřebný k otestování více prohlížečů a operačních systémů.
- Selenium IDE – je rozšíření prohlížeče Firefox, které umožňuje snadno nahrávat a přehrávat testy. Může být použito pro generování kódu ke spouštění testů se Selenium Remote Control.
- Selenium Remote Control – jedná se o klient-server systém, který umožňuje ovládat webové prohlížeče lokálně nebo na vzdáleném počítači s použitím téměř libovolného programovacího jazyka.

Hlavním cílem automatizačních nástrojů pro testování softwaru je časová úspora člověka. Když se testy spouštějí a vyhodnocují automaticky, šetří se čas a předchází se chybovosti pracovníků testujících manuálně. Testeři totiž mohou při procházení stále stejných testovacích scénářů přehlédnout nějaké chyby, a právě tomu má automatizace zabránit.

Zjednodušeně řečeno, Selenium umožňuje simulovat postup člověka při procházení webové aplikace – např. kliknutí myší (i vícenásobné), zadávání URL, psaní znaků apod. Selenium je kromě automatizovaných testů možné využít také pro vytváření objednávek v e-shopu, odeslání velkého množství e-mailů, vyplňování formulářů a mnoho dalších procesů.

Původním záměrem bylo použití Selenia pro procházení e-shopu v rámci této práce. Tato myšlenka nebyla nakonec uskutečněna, jelikož by procházení a ukládání dat bylo značně pomalé. Navíc Selenium nezaručuje stoprocentní funkčnost. Pokud je například na stránce změněn nebo přidán nějaký element, tak se může Selenium nečekaně ukončit neboli spadnout.

2.3 JSON-Simple

Jak již název napovídá, jedná se o jednoduchý, ale mocný nástroj, který poskytuje čtení a zápis JSON streamů. [22] Tato knihovna je plně shodná s JSON specifikací RFC 4627. [34] Je zaměřena hlavně na kódování, dekódování a získávání JSON kódu. Podporuje streamovací výstup JSON textu, umožňuje vysoký výkon, není závislá na externích knihovnách, je flexibilní a jednoduchá pro opakované použití. [22]

Jazyk Java nepodporuje JSON zápis. Místo toho, aby si vývojáři vytvářeli svoje řešení, je lepší použít bezplatnou open-source knihovnu. Mezi další knihovny patří například org.json, Jackson, XStream, Gson a další. JSON-simple není nejlepším nástrojem, ale díky své jednoduchosti a účinnosti je velmi často využíván. Proto byla tato knihovna zvolena jako nejvhodnější pro řešení této bakalářské práce.

2.4 SQL databáze

Databáze se dá představit jako pracovní sešit programu Excel, který se skládá z několika listů, zvaných tabulky, které mají svůj vlastní název. Tabulky se skládají z řádků a sloupců. V jednom sloupci se většinou nachází nějaká informace jako například jméno studenta.

Řádek pak tvoří skupina informací, které se vztahují například k danému studentovi - jméno, příjmení, identifikační číslo, věk apod. Data se ukládají tam, kde se řádek protíná se sloupcem.

Databáze je soubor tabulek a tabulky soubory řádků a sloupců s daty. Databázový řídící systém (DŘBS, angl. Database Management System) je aplikace, která umožňuje pracovat s daty v databázi, např. ukládat informace, zobrazovat je, třídit, mazat apod. V DŘBS se pracuje pomocí instrukcí strukturovaného dotazovacího jazyka (SQL, angl. Structured Query Language). [24]

2.5 Hash

Hashovací (nebo též hašovací) funkce je funkce matematická, která se používá k zajištění integrity dat. Jedná se o algoritmus, který přemění vstupní data o různých délkách do hashovaných zpráv o stejně definované délce. Dvě různě dlouhé zprávy tedy mají po zahashování stejnou délku. Z hashované zprávy nelze odvodit původní zprávu, jelikož byla jednosměrně zašifrovaná. [25]

Hashovací funkce urychlují vyhledávání v tabulce nebo v databázi, například při zadávání hesla uživatele. Jedním z využití je detekce duplicitních záznamů ve velkém souboru dat. Příkladem může být nalezení podobných úseků v sekvencích DNA. Jsou taky užitečné v kryptografii, ve čtečce otisků prstů, pro ukládání a kontrolu přihlašovacích hesel, pro korekci chyb apod. V rámci bezpečnosti například zůstává případnému útočníkovi skryto, kdo a jaké heslo používá. Díky hashování lze zabránit útokům hrubou silou nebo slovníkovým útokům.

```
Text zprávy:
    Jména 'John Doe' pro muže nebo 'Jane Doe' pro ženy slouží jako zástupné názvy,
    jejíž pravá identita není známa nebo musí být zadržena v soudním řízení, případu
    nebo diskusi.

Hash zprávy - MD5:
    866BA843319F3C05820767BC30F9B4EE

Hash zprávy - SHA-1:
    A7D55B2655CC42B268BEA90F50FD17767D6EA989

Hash zprávy - SHA-256:
    23C0ECBA0B6BFEAF151F4DE58B7F1709AE407AF68C1FFC207B8F88D8734A039F

Hash zprávy - SHA-512:
    A1F84D37A8C7B214835DD00627FD8EAC31C60B544E4D184B023DDFBBB6664BEF94F7D22CD758FF42
    DAC1532FF9ACA107254612458F652848F445BD1C13154FB7
```

Obr. 2.1: Ukázka zahashované zprávy pomocí několika šifrovacích algoritmů

Známé kryptografické hashovací funkce:

- MD5 (Message Digest 5) otisk vstupních dat má délku 128 bitů, v současné sobě se již moc nepoužívá, jelikož byla tato šifra prolomena, [25]
- SHA-1 (Secure Hash Algorithm) šifruje data ve 160 bitech, [26]
- SHA-2 je rodinou čtyř algoritmů: SHA-224, SHA-256, SHA-384 a SHA-512, kde číselná označení určují výstupní bitovou délku. [26]

Na obrázku 2.1 jsou znázorněny různé hashovací funkce. Je vidět, jak se každý hash liší svojí délkou, ale přitom nezávisí na počtu znaků v těle zprávy.

3 VLASTNÍ NÁVRH ŘEŠENÍ

Cílem této bakalářské práce je vytvoření automatického nástroje pro získávání informací z webu. Tato kapitola se zabývá způsobem vytvoření daného nástroje a také jeho použitím. Zároveň je zde popsáno, jaké problémy bylo nutné řešit během vývoje.

3.1 Zvolené řešení

Nejjednodušším způsobem, jak stáhnout komentáře z e-shopu, by bylo použití aplikací z podkapitoly 1.9. Tato práce však zachází více do hloubky problému. Úkolem je vytvořit nástroj pro získání obsáhlé databáze komentářů, která je rozdělena na pozitivní a negativní text. Kdyby zmíněné aplikace delší dobu stahovaly data z čínského e-shopu, pravděpodobně by byly detekovány jako zdroj nekalého chování a byl by jim znemožněn přístup na web. Tyto programy navíc neřeší problémy s připojením a nedají se libovolně editovat. Pokud by například nastal jakýkoli problém na straně webu, program by jednoduše spadnul. Navíc by bylo nutné vyřešit i automatické ukládání textů do SQL databáze.

Jako reakce na zabránění scrapingovým systémům v prohlížení webu musí být tyto techniky programovány tak, aby simulovaly lidské chování při procházení webu a nebyly přitom detekovány. Tento problém by mohlo řešit použití programovacího jazyka Java spolu se Seleniem. Jak již bylo ale uvedeno v části 2.2 této práce, takové řešení by bylo značně pomalé. Selenium se hodí spíše pro vytváření automatických testů a pro simulování chování člověka na webové stránce. Pro vytvoření pavouka pro získávání informací z webu by řešení bylo zbytečně složité.

Zároveň není žádoucí, aby procházení bylo příliš pomalé. Použití Javy s knihovnou jsoup tedy jednoznačně vyhrává, zejména z důvodu jednodušší implementace. Je možné vytvořit program přesně podle potřeb, zahrnout ošetření problémů s připojením nebo s neexistujícím produktem v e-shopu, lépe maskovat chování programu apod. Komentáře na daném e-shopu jsou uloženy v JSONu a vykreslovány AJAXem. Bylo tedy nutné použít ještě knihovnu JSON-simple pro práci s JSON daty a knihovnu java.sql pro připojení k databázi. Komentáře jsou ukládány do MySQL databáze. Pro účely testování programu byla vytvořena MySQL databáze pomocí nástroje phpMyAdmin a XAMPP.

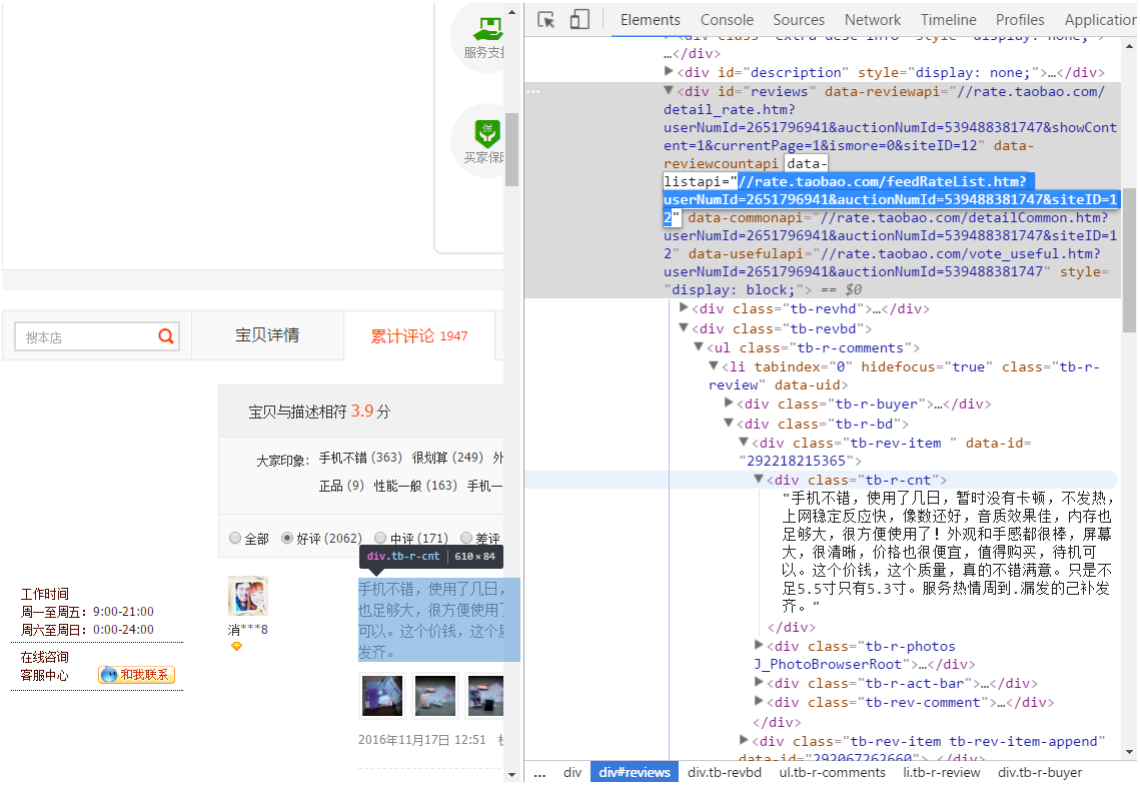
3.2 Průzkum e-shopu

Dnešní moderní e-shopy již nevyužívají pouze webové stránky se statickým obsahem, ale především stránky dynamické. To znamená, že ve zdrojovém HTML kódu je možné vidět pouze statickou „kostru“ webu a dynamická část je skryta v JavaScriptu nebo v JSONu, který zobrazuje AJAX. Například po kliknutí uživatelem na nějaké tlačítko webu se může zobrazit nový text, který však na první pohled nelze nalézt ve zdrojovém kódu.

Zadaný web je čínský e-shop, na kterém je velké množství produktů, z nichž drtivá většina obsahuje různě dlouhé komentáře přímo od majitelů zboží. E-shop slouží pouze

pro Čínu a je celý v čínském jazyce. Jazyková sada není pro programové řešení překážkou, jelikož pro různé jazyky zůstává způsob řešení stejný. Daný e-shop má podobnou strukturu jako známé internetové obchody Ebay, AliExpress nebo Alibaba. Obsahuje celkem dvanáct kategorií s několika tisíci produkty od oblečení až po spotřební elektroniku. Každý produkt má svou vlastní webovou adresu a obsahuje klasické náležitosti, jako jsou fotografie, popis zboží, listy s komentáři, cenu, hodnocení, reklamy, výběr počtu kusů a mnohé další.

Pro účely této práce je zajímavá především karta s komentáři. Ta obsahuje číselné hodnocení produktu, celkový dojem a výběr skupin komentářů pomocí tlačítek typu „radio button“. Příspěvky lze rozdělit celkem do šesti skupin: všechny komentáře, pozitivní, průměrné, negativní, poptávka a komentáře s obrázky. Dále je zde možné nalézt celkový počet komentářů, který je uveden i u každé hodnotící skupiny.



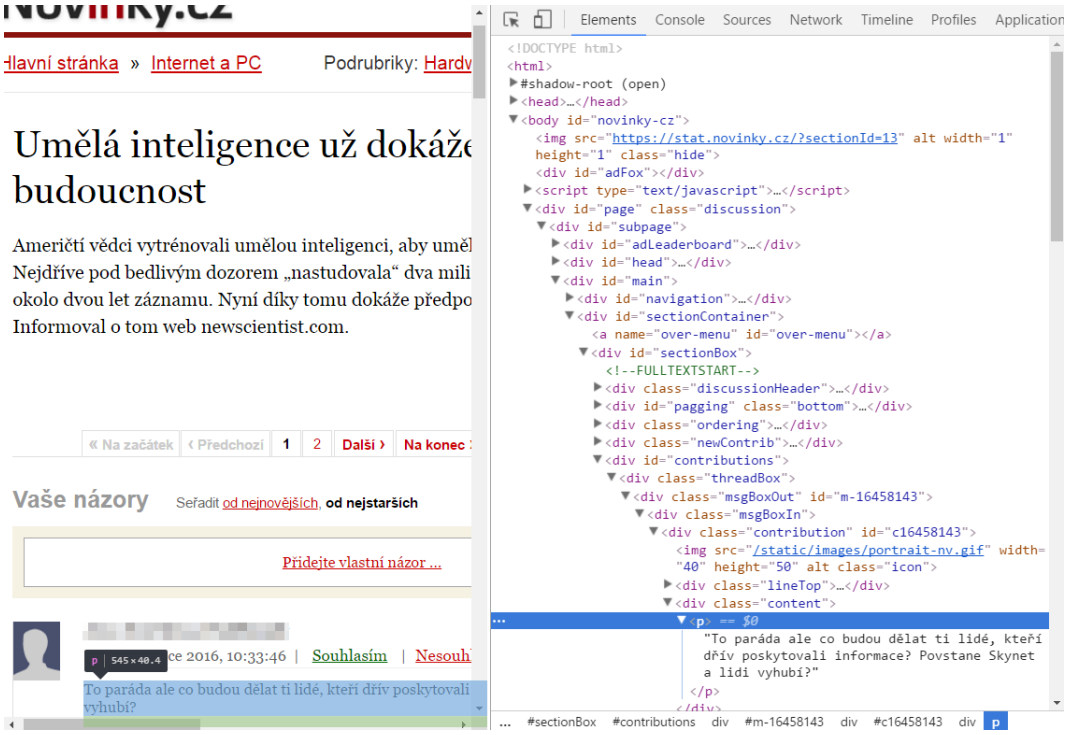
Obr. 3.1: Zobrazení JSON adresy komentářů ve zdrojovém kódu e-shopu

Při prohlížení zdrojového kódu stránky produktu ve webovém prohlížeči nelze jednoduše komentáře nalézt. Na první pohled jako by tam nebyly. Prozkoumáním správného odstavce (např. pomocí funkce „Prozkoumat“ v prohlížeči) lze zjistit, že texty jednotlivých komentářů jsou vnořené v elementu `<div id="reviews">`. V tomto elementu se nachází další API atributy, ze kterých je nejdůležitější `data-reviewcountapidata-listapi`, který obsahuje adresu s JSON daty, viz obr. 3.1. Když v této adrese upravíme parametr `rateType` na rovný nule nebo jedné, zobrazí se JSON data s kladnými nebo zápornými komentáři.

Z toho vyplývá, že získání komentářů nepůjde jednoduše stáhnout z HTML kódu. Bude nutné vytvořit program, který dokáže vyhledat na stránce produktu adresu s JSON daty, upravit tuto adresu podle typu hodnocení a poté stáhnout požadovaný text, který se bude ukládat do databáze.

3.3 Prvotní testování knihovny jsoup

Jak bylo uvedeno již v kapitole 2.1, knihovna jsoup dokáže uložit vše, co se nachází ve statické části webové stránky. Problém tedy přichází, jestliže je třeba získávat dynamickou část webu napsanou např. v JavaScriptu, jelikož s tímto jazykem neumí jsoup pracovat. Pro první seznámení s prací na nástroji byly použity komentáře ze zpravodajského serveru https://www.novinky.cz.



Obr. 3.2: Zobrazení komentáře ve zdrojovém kódu serveru Novinky.cz

Komentáře k článkům se nacházejí odděleně na další webové stránce a jsou statické, takže stahování pomocí JSONu nebyl problém. Text s komentáři se poměrně jednoduše podařilo uložit do textového souboru. Testovací program vyhledával všechny elementy s tagy `<div class="content">` a v nich odstavec `<p>` (viz obr. 3.2).

Na podobném principu můžeme stáhnout příspěvky z diskuzního fóra, odkazy nebo obrázky atd. Kromě ukládání určitého odstavce je možné například stáhnout celý HTML kód, či pouze jeho část ve značkách `body` a pomocí knihovny commons.lang3.StringEscapeUtils

uložit text bez HTML entit. Pro zadaný e-shop však není řešení tak jednoduché. Komentáře jsou totiž zobrazované dynamicky, jak již bylo zmíněno výše.

3.4 Představení programů

Aby nástroj pro automatické získávání informací z webu pracoval automaticky bez manuálního zásahu uživatele, musely být naprogramovány aplikace dvě - `ProductsCrawler.jar` a `CommentsCrawler.jar`. Programy byly napsány ve verzi Javy SE 8 ve vývojovém prostředí Eclipse. Jeden program získává linky produktů a ukládá je do databáze. Druhý tyto linky postupně prochází, připojuje se k nim, vyhledává kladné a záporné komentáře a i ty ukládá do MySQL databáze.

Obr. 3.3: Struktura databáze

Schéma databáze je zobrazeno na obr. 3.3. Tabulka `products` pro ukládání linků obsahuje sloupce `id, hash, link` a `last_check`. Tabulka `comments` pro ukládání čínských příspěvků se skládá ze sloupců `id, lang, positive, hash` a `text`.

Součástí je ještě třída `PublicMethods.class` s uloženými veřejnými metodami, které využívají obě výše zmíněné aplikace. Tato třída slouží pouze pro větší přehlednost a pro zamezení vzniku chyb při duplikování zdrojového kódu.

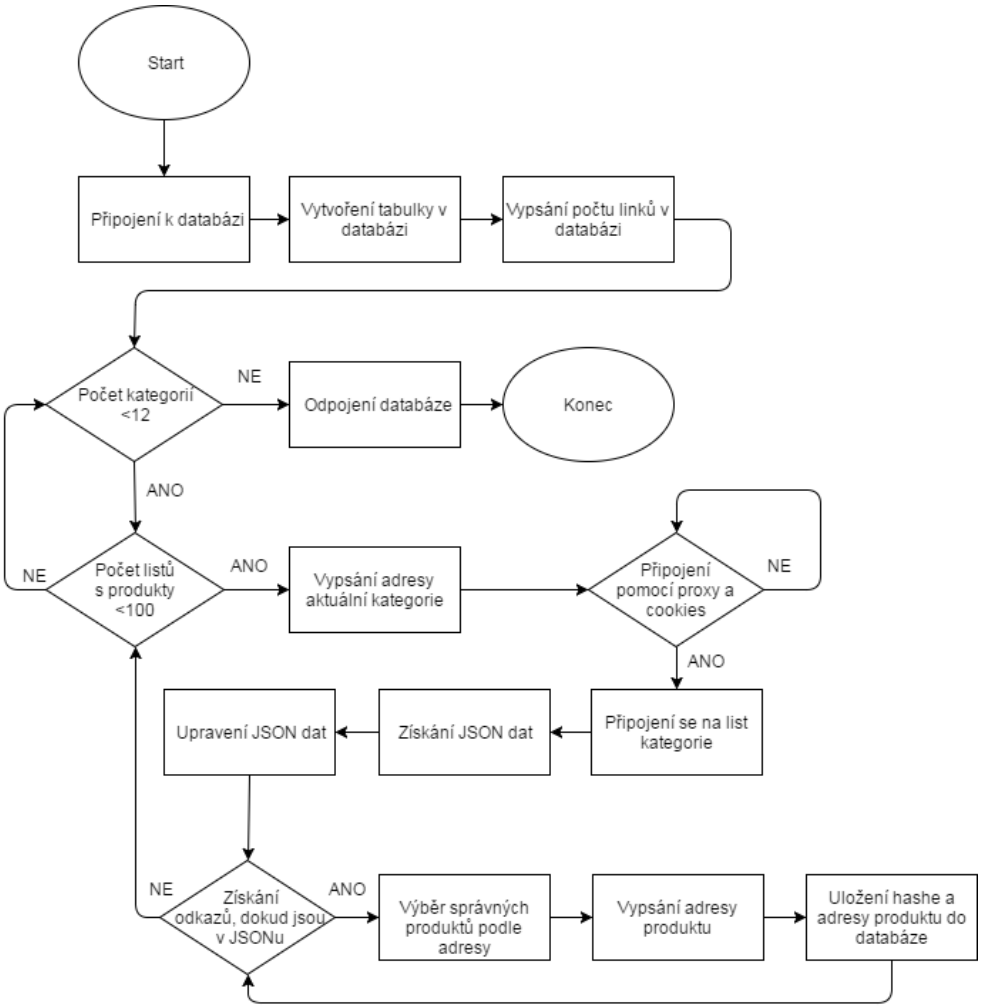
3.4.1 Postup programu pro získávání linků

Aplikace budou nasazené na serveru a data se budou ukládat do databáze, proto je nejprve nutné uložit odkazy na produkty. Na začátku programu `ProductsCrawler.jar` se tedy naváže spojení s databází a vytvoří se tabulka `products`, pokud ještě nebyla vytvořena. V případě, že již tabulka existuje, zjistí se také počet linků v databázi.

Zadaný čínský e-shop obsahuje celkem dvanáct kategorií skládajících se převážně z dámského a pánského oblečení, elektroniky, hraček, sportovních potřeb a dalších. Program prochází postupně všechny tyto kategorie. Odkazy na ně jsou pevně uloženy v kódu programu. Každá kategorie má sto listů s produkty a každý list zobrazuje na stránce šedesát produktů, tzn. šedesát odkazů. K odkazu na kategorii je proto nejprve přidán parametr

`data-value` rovnající se nule, což značí první list. Při každém dalším cyklu se k hodnotě tohoto parametru přičítá šedesát. Takovým způsobem se navyšuje hodnota až do šesti tisíc (tj. do stého listu) a poté se přechází na další kategorii.

Dále je na konec odkazu přidán řetězec `&module=page&nid=&type=&uniqpid=`. Teprve na takto upravený odkaz se může program připojit. Při připojování se nejprve provede zkouška připojení na proxy adresu a port z pole proxy serverů. Pokud je vše v pořádku, program pokračuje následujícím krokem. V opačném případě se pro připojení použije další ze seznamu proxy serverů, a to se opakuje do doby, než se nalezne bezproblémové připojení na některou proxy. Toto řešení je nezbytné pro navození anonymity, a tedy zabránění blokování stahování JSON dat.



Obr. 3.4: Vývojový diagram aplikace pro ukládání linků produktů

Program se tedy úspěšně připojí na upravenou stránku, která sice obsahuje JSON data, ale ta ještě nejsou v požadovaném formátu. Knihovna JSON-simple by je neuměla identifikovat. Aby se data zobrazovala správně, je nutné odebrat prefix `if(window.__jsonp_cb)`

`{__jsonp_cb(` a konečnou závorku. Teprve po této úpravě může knihovna s JSON daty pracovat.

Nejprve jsou v JSONu vyhledány odkazy na produkty a podle pole `itemList` a jeho elementu `href` se ověří, zda jsou skutečně uloženy na zadaném e-shopu. Některé produkty, většinou okolo pěti případů na listu (někdy však více), se totiž mohou nacházet na jiných e-shopech, jež mají odlišnou strukturu, a proto jsou programem vynechávány.

Linky na produkty aktuální kategorie jsou postupně stahovány do databáze do té doby, než je načteno všech sto listů. Webová adresa produktu se ukládá do sloupce `link` s pořadovým číslem `id`. Do sloupce `last_check` je defaultně vložena hodnota `NULL`. Později bude přepsána časem posledního použití daného linku při vyčítání komentářů programem `CommentsCrawler.jar`. Vzhledem k tomu, že produkty se na jednotlivých listech kategorie objevují náhodně, je možné, že se při získávání linků některý opakuje. Z tohoto důvodu jsou jejich adresy šifrovány pomocí SHA-256 a do databáze se tedy ukládá i hash, který zaručuje, aby v ní nebyly duplikáty.

Z jedné kategorie je možné uložit až šest tisíc odkazů. Po projetí jedné kategorie program pokračuje další, dokud se neuloží poslední produkt ze dvanácté kategorie. Poté se ukončí spojení s databází a program končí.

Zjednodušený vývojový diagram průběhu aplikace `ProductsCrawler.jar` je znázorněn na obr. 3.4.

3.4.2 Postup programu pro stahování komentářů

Stejně jako `ProductsCrawler.jar` i program `CommentsCrawler.jar` ukládá data do databáze, proto se nejprve zajistí připojení. Pokud ještě neexistuje tabulka `comments`, vytvoří se a bude sloužit pro ukládání komentářů. Jestliže tabulka již existuje, program zjistí počet řádků v tabulce, aby mohl pokračovat na novém řádku s dalším identifikátorem v pořadí.

Program se následně připojí také k tabulce `products`, která již byla vytvořena předchozím programem `ProductsCrawler.jar` a obsahuje odkazy na jednotlivé produkty. Z tabulky `products` je vyčtena webová adresa zboží, ze které se budou stahovat komentáře. Zároveň je uloženo i aktuální datum a čas k danému odkazu v tabulce `products` do sloupce `last_check`. Tento časový údaj slouží k rozeznání posledního použitého linku. Pokud by byl program přerušen a opět spuštěn, začne od produktu s nejmenší časovou hodnotou, případně načítá produkty, které ještě nebyly použity (`last_check = 0`) v pořadí podle `id`.

Knihovna jsoup při připojování na e-shop použije jednu z proxy adres včetně parametrů cookies a hlavičky získaných z několika prohlížečů. Po každém úspěšném připojení je volána funkce sleep, která je blíže popsána v kapitole 3.5.1 a zajistí simulování chování člověka, aby nebyl program blokován. Pokud by nastal problém s připojením, použije se další proxy server, cookies a hlavička v pořadí, dokud nebude připojení v pořádku.

Jsoup na produktové stránce vyhledá element `<div id="reviews">` a z něho uloží JSON adresu. K takto nalezené adrese musí být přidán prefix `http://` a také postfix `&currentPageNum=1&rateType=1&orderType=sort_weight&showContent=1`

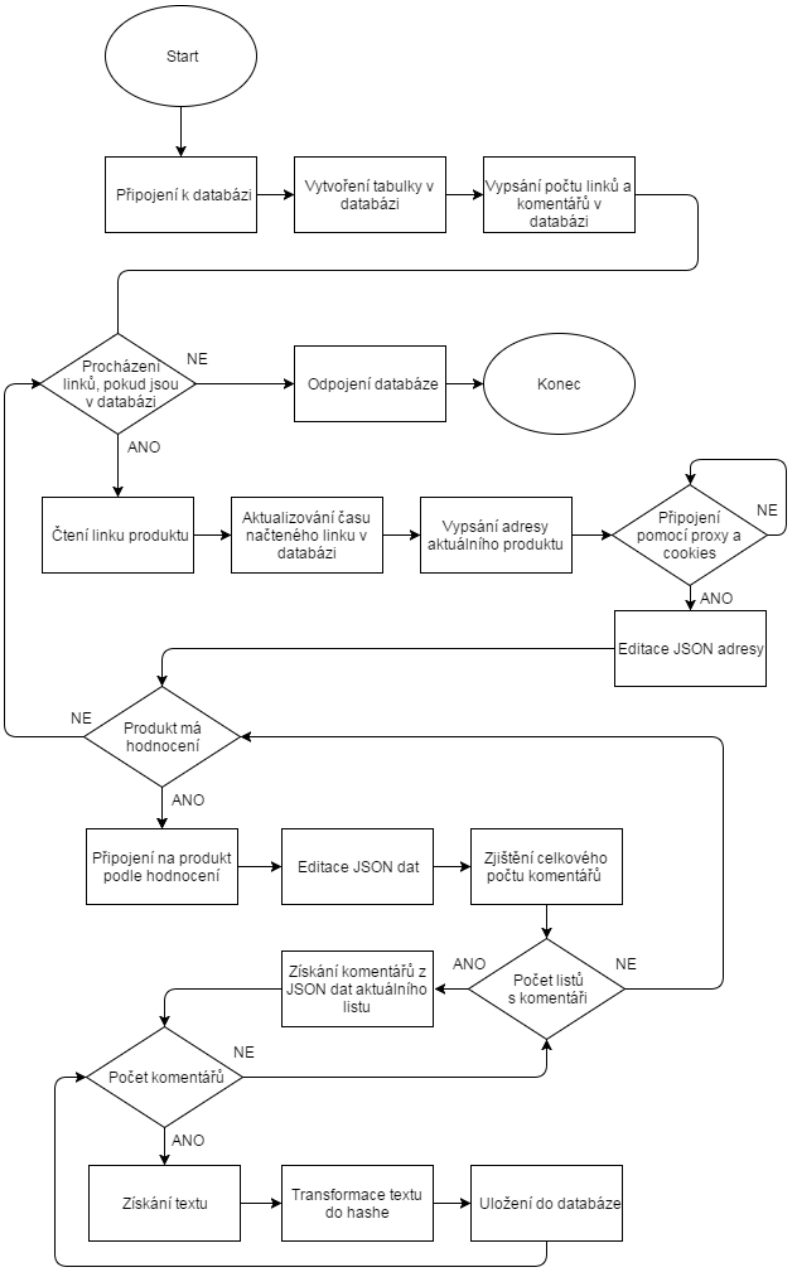
`&attribute=`. Hodnoty parametrů `currentPageNum` a `rateType` se budou v průběhu programu měnit. Parametr `currentPageNum` udává číslo aktuálního listu s komentáři a parametr `rateType` určuje, zda se jedná o komentáře pozitivní nebo negativní. Jsoup se připojí k výsledné adrese a v obsahu stránky odstraní na začátku a konci závorky, se kterými by knihovna JSON-simple neuměla pracovat.

V další části se nejprve vyhledávají všechny záporné (tj. `rateType = 0`) a poté kladné (tj. `rateType = 1`) komentáře. Celkový počet komentářů se v JSON datech nachází v objektu `total`, ze kterého lze při podělení počtem komentářů na jednom listu (20) zjistit, kolik listů bude nutno cyklicky procházet. V JSONu se pak také nachází pole `comments` obsahující objekty `content` s čínskými komentáři. Tyto komentáře se nejprve transformují do hashe pomocí algoritmu SHA-256 a spolu s čínským textem jsou ukládány do tabulky `comments`. Aby byly čínské znaky v databázi čitelné, musí být kódovány pomocí UTF-8.

Stává se, že se některá hodnocení opakují, ať už u stejného nebo u dalších produktů. V databázi by se proto nacházelo několik stejných textů, což není pro finální účel žádoucí. Díky porovnávání hashe je ale možné ošetřit, že se neukládají duplicitní záznamy. Pokud k této duplicitě dojde, komentář je jednoduše přeskočen a ukládá se až další v pořadí. Proto jsou hashovány i linky při ukládání pomocí programu `ProductsClawler.jar`. Pokud by se totiž v databázi nacházelo více stejných odkazů, zbytečně by se zdržoval průběh programu `CommentsCrawler.jar`. Komentáře takového linku by podruhé již byly v databázi, ale přesto by program zbytečně procházel všechny jeho listy.

Cyklus pokračuje, dokud daný list obsahuje komentáře, poté přechází na další list. Po stažení všech negativních příspěvků program vykoná stejné operace pro příspěvky pozitivní. Jestliže produkt neobsahuje záporné komentáře, cyklus pokračuje ve stahování kladných komentářů. Stává se, že produkt neobsahuje komentáře žádné. V takové situaci je produkt přeskočen.

Zjednodušený vývojový diagram průběhu aplikace `CommentsCrawler.jar` je znázorněn na obr. 3.5.



Obr. 3.5: Vývojový diagram aplikace pro získávání komentářů

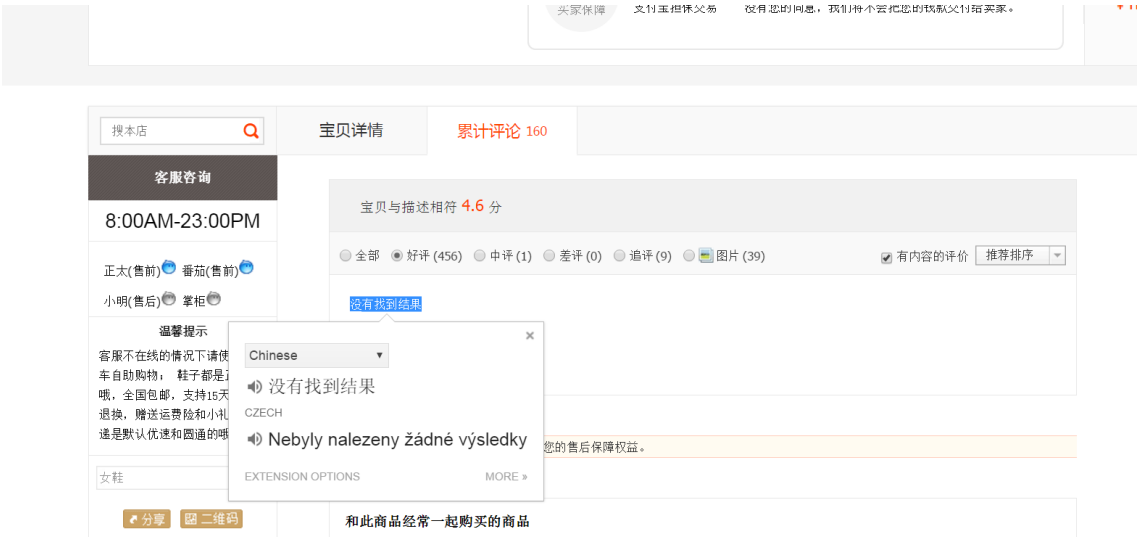
3.5 Problémy s e-shopem

Při programování a práci s e-shopem nastává několik problémů, které přístup ke komentářům znemožňují. První komplikací bylo samotné vyřešení automatického získávání komentářů z JSON dat, další jsou však na straně e-shopu.

3.5.1 Problém s dolováním dat

Největší problém nastal v téměř finální verzi programu `CommentsCrawler.jar`. Program byl spuštěný cca dvacet hodin a stáhlo se 883 351 čínských komentářů. Poslední produkt, ze kterého se stahovalo, měl pořadové číslo 611 z celkového počtu 37 592 linků. Poté program spadnul a vypsal chybu, podle které nemohl stáhnout JSON, a proto nebylo možné zjistit ani celkový počet komentářů u aktuálního produktu.

Provozovatelé e-shopu zřejmě zjistili, že jim nějaký bot doluje data a zablokovali proxy server počítače, ze kterého byl program spuštěný. Při manuálním průzkumu posledního použitého produktu mělo být komentářů 160, ale místo nich se zobrazovala pouze chybová hláška, která v překladu znamená „Nebyly nalezeny žádné výsledky“, viz obrázek 3.6. Místo adresy na JSON data, byla v elementu `<div id="reviews">` adresa pro přihlášení aplikační podpory. Na jiném stroji a jiné proxy se však komentáře a JSON zobrazovaly v pořádku.



Obr. 3.6: Ukázka chybové hlášky místo zobrazení komentářů

Program pro stahování čínských komentářů však po tomto updatu ze strany internetového obchodu nebylo možné dále používat bez úprav. E-shop od té doby dokáže detekovat, že byl zatížen botem a přestal vracet požadovaná `GET` data. Proto bylo nutné program co nejlépe zamaskovat, aby se choval co nejpodobněji člověku. Když se uživatel připojí na nějakou webovou stránku, daný web obdrží informace, o jaký prohlížeč se jedná, jaké je

rozlišení obrazovky, jaký používá jazyk atd. v hlavičce (header). Prohlížeč také stahuje informace o stránce formou cookies a při každé další návštěvě tyto informace odesílá zpět serveru. Každý uživatelský počítač má jiné parametry v hlavičce i v cookies. Tyto parametry bylo tedy nutné přidat do příkazu `GET` při připojování na stránku produktu pomocí knihovny jsoup. Aby se při každém připojení odesílala jiná skupina cookies a hlavičky pro lepší simulování prohlížení stránek člověkem nikoli botem, byla použita pole těchto hodnot z více počítačů.

Takové řešení však nemusí stačit. Když bude pavouk spuštěný delší dobu (například týden), aplikační podpora by si pravděpodobně všimla, že jim někdo stahuje data, a tím zatěžuje server ze stále stejné IP adresy. I když se mění hodnoty hlavičky a cookies, je pro větší maskování vhodnější použití více proxy serverů z různých států světa. Tím bude zaručeno, že nebude bot tak rychle identifikovatelný. Po zavedení této úpravy tedy pavouk mění při každém připojení na produkt proxy adresu.

Součástí je i funkce sleep, která pozastaví program v náhodném rozmezí pěti až tří set sekund. Díky tomu nebude činnost aplikace vypadat podezřele. Člověk totiž také není schopný například stokrát za sekundu vykonat na e-shopu nějakou činnost. Sice se tím sběr dat může zdržet, ale bude zajištěn dlouhodobý běh programu bez chyb a padání.

3.5.2 Všeobecné problémy

Při klasickém prohlížení některých produktů (i se stovkami komentářů) ve webovém prohlížeči se při otevření karty s komentáři zobrazí pouze chybová hláška „undefined“, viz obr. 3.7. I po několika obnoveních stránky se někdy komentáře zobrazí, někdy ne. Chyba je způsobena špatným načítáním JavaScriptu na straně e-shopu a nelze jí zabránit. Na funkci programu to však nemá vliv, jelikož se komentáře stahují z JSONu.

Obr. 3.7: Ukázka chybové hlášky „undefined“

Vypracovaný program má však také občas problém se k některým produktům připojit. Pokud k takové situaci dojde, pokus se opakuje s následující proxy adresou, dokud není připojení úspěšné. Podobný problém nastává i při připojování k JSON adrese listu s komentáři. I v takovém případě se opakuje připojení s další proxy.

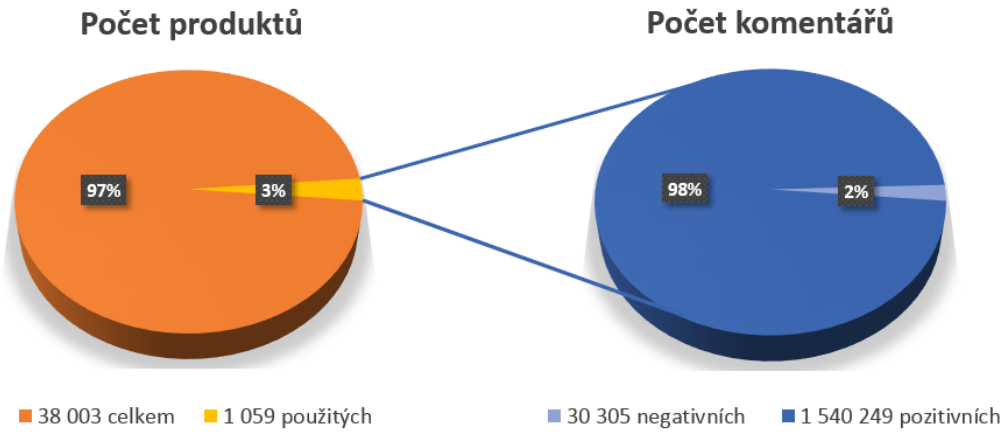
Může se stát, že některé produkty jsou z nabídky e-shopu kompletně vymazány nebo nejsou podrobnosti již dostupné. Ani tady ale není problém, produkt se jednoduše vynechá.

4 VÝSLEDKY A DISKUZE

Způsob vytvoření nástroje pro automatické získávání komentářů z čínského e-shopu byl popsán v předchozí kapitole. Vypracovaný projekt dokáže získat linky produktů internetového obchodu a ukládat jejich kladné a záporné komentáře do databáze. Celkový počet získaných komentářů při týdenním běhu byl více než jeden a půl milionu.

4.1 Dosažené výsledky

Jak bylo popsáno v kapitole 3.5.1, jeden z největších problémů nastal téměř v konečné fázi tvorby programu. V předchozí verzi chybělo doplnění polí proxy serverů, cookies, hlavičky a také funkce sleep. Program pracoval cca dvacet hodin bez problému, ale poté spadl a proxy počítače byl zablokován přístup na e-shop. Podařilo se stáhnout 37 592 linků a 883 351 komentářů. V této verzi byl také problém v tom, že pokud se nepodařilo připojit na nějaký produkt, link byl okamžitě přeskočen. Proto bylo za tu dobu použito celkem 611 odkazů. Kdyby některé linky nebyly nepřeskočeny, tak by pro stažení stejného počtu komentářů mohlo být použito méně produktů. Poměrně velký počet komentářů se podařilo stáhnout za cca dvacet hodin díky rychlému internetovému připojení, ale především díky tomu, že se na e-shop nebylo nutné připojovat přes proxy servery, což celý průběh zpomaluje. Řešení pomocí proxy serverů a doplňování hlavičky a cookies do parametru `GET` znamenalo zvýšení robustnosti programu. Díky tomuto vylepšení je možné mít program spuštěný nekonečně dlouho nebo minimálně do doby, než se vyčerpá místo na disku, a tedy i kapacita databáze.



Obr. 4.1: Graf srovnání shromážděných dat za sedm dní

Ve finální verzi se za sedm dní podařilo stáhnout 1 570 554 čínských komentářů, z nichž 1 540 249 bylo pozitivních a 30 305 negativních. Poslední použitý link měl identifikační číslo

1 059 z celkem 38 003 produktů, viz obr. 4.1. Náhledy do databáze produktů a výběr z kladných a záporných komentářů lze vidět na obrázcích 4.2, 4.3, resp. 4.4. Snížení rychlosti stahování komentářů způsobilo jednak spouštění na počítači s výrazně pomalejším připojením a také používání různých proxy severů. Očekávání bylo, že bude staženo alespoň sto tisíc komentářů s kladným i záporným hodnocením. Jak lze vidět, lidé ale zřejmě nekupují produkty, které obsahují větší množství negativních příspěvků, a proto žádné další ani nepřibývají. Všechny linky, které program během těchto sedmi dnů stihl projít, navíc spadaly do kategorie oblečení. Dá se očekávat, že větší počet negativních komentářů se objevuje např. u produktů ze sekce elektroniky. Program pracoval celý týden bez nějakého velkého narušení ze strany e-shopu. Pokud se vyskytl problém s připojením, byl poskytnut další proxy server ze seznamu. Některé adresy proxy serverů fungovaly bez problému po celou dobu běhu programu. S jinými se naopak někdy podařilo připojit ke stránce produktu a někdy ne. Pro zjištění funkcionality proto bylo použito téměř sto volně dostupných proxy adres z webů http://free-proxy-list.net/ a http://free-proxy.cz/cs/. Pro zajištění anonymity a vůbec simulování chování člověka se do metody pro připojování vkládají parametry cookies a header, které byly uloženy ze třinácti různých počítačů.

id	hash	link	last_check
1	44d57759388a5dcd3b806794e8cc9d3ca7fafc99ff35c684f2...	https:/ /item/548090511874.htm#det...	2017-06-01 10:10:11
2	d809b9e1300b99c481d841ba29a9c5bb857f3779ae0fc710e7...	https:/ /item/536200987552.htm#det...	2017-06-01 10:11:47
3	68d9300f25ed53d99dc1a021dd064339178d42955c57b15e17...	https:/ /item/545322248594.htm#det...	2017-06-01 10:27:10
4	d0c8c9b9fa9125c830b2700171e8689a35da74cbe934492ab8...	https:/ /item/548186944940.htm#det...	NULL
5	35820b6481146ca55248101617c6d37dcb4c417c12d1a74fc2...	https:/ /item/546667134183.htm#det...	NULL
6	c1566cbac2272a10d78bd2c4207bb152f14d547f90e9c4eb44...	https:/ /item/546514097677.htm#det...	NULL
7	b7f113590f61091b4c12f81f272519ba141027a23c68add923...	https:/ /item/545224167566.htm#det...	NULL
8	90dadece168e8ea5d49c5270a1237025b231f16cbc80939b6d...	https:/ /item/524058286643.htm#det...	NULL
9	58d8f8287783ee1402a1ed2fdaae297bf905fccbb7832829a2...	https:/ /item/529105143644.htm#det...	NULL
10	95752f7c00afb109122ec0bc5bf46cb7d1276089ef1ce21f71...	https:/ /item/526924655691.htm#det...	NULL

Obr. 4.2: Výpis linků z databáze

Nakonec tedy bylo dobře, že vývojáři použitého čínského e-shopu přidali ochranu proti dolování dat. Díky vyřešení tohoto problému se alespoň program stal více anonymním.

Pokud by byl nástroj pro získávání komentářů spuštěný například dva měsíce, mohlo by se teoreticky stáhnout cca 12 500 000 komentářů, z toho zhruba 250 000 negativních a 12 250 000 pozitivních. Program by za takovou dobu prošel přibližně 8 500 produktů, což jsou stále jen asi tři kategorie ze dvanácti. Tyto výpočty jsou samozřejmě pouze orientační. Hodně záleží na tom, jestli se komentáře neopakují především v rámci stejné kategorie. Každá kategorie ale obsahuje odlišné druhy produktů, proto by i hodnocení mělo mít jiné znění.

id	lang	positive	hash	text
1570627	CHI	1	f7af5a25b10dc6ff5cc1438e74bf4def6d8a	面料很好 价钱值
1570628	CHI	1	09f4ecf254675e457f81c15fd141b724444	物流速度快，衣服质量好是纯棉的哈哈哈，卖家服务态度好，生意兴隆啊，
1570629	CHI	1	d97990f3b87d15fcadb27ce0753e05728c	没带手机去学校，所以一直没评价，衣服很好，是我喜欢的，第一次感觉在网上买东西是买对了的，很开心⊙▽⊙
1570630	CHI	1	7c5a5417e97ba87bf176712c2107963bcc	挺舒服的，价钱优惠,值得购买
1570631	CHI	1	c42b7854af249dc1eaf6391d710076963c	爱死料子了 第二次买了
1570632	CHI	1	3260951f85f216cc59780a6e04425762fa5	洗的时候感觉布料挺好
1570633	CHI	1	2616e6f7162383070b5094a3872201690(	衣服有点偏大，质量呗，能买好的就买质量好的吧！这衣服可以打底穿，有点小透！
1570634	CHI	1	ac24b1284306ac22832469c6d4a82ea0d	衣服很合身，质量也可以，很喜欢
1570635	CHI	1	327c649bc52775bdf8e4d4b559e66b514(	很舒服，客服推荐的大小正合适，很喜欢，洗了还不掉颜色
1570636	CHI	1	40542f846a27952c7cb2117098e670f50b	跟图片描述一样，挺好的
1570637	CHI	1	31d6178bd0493b8104da09f23ebb7af502	宝贝非常好，这是第四次购买，只有好的宝贝才会有回头客，况且价钱好质量也不错，中间出了一点小问题，都解...
1570638	CHI	1	04e4060db938def2a181490111261d38d(	好着呢，纯棉的，挺柔软的，值得购买！
1570639	CHI	1	487a399b5a21e07470b8198de0d428f4e(	第二次买了，无条件好评！强烈推荐给各位仙女宝宝！
1570640	CHI	1	51643ae602d2a5989e96a982e5b65703f(	质量还可以，就是是有点贵

Obr. 4.3: Výpis pozitivních komentářů z databáze

id	lang	positive	hash	text
1567697	CHI	0	975f7380fb3411daec97204dbd7af3eef4	遇到这种商家真的是恶心到家，明明买的是两件短裤，回来只有一件，后来问客服也不吭声了，为了这点钱至于吗...
1567698	CHI	0	7286c028da58101dadcad1bf195730f7e	严重色差，而且码数也不对，我也是醉了，想想价钱摆在那里，气死了，就当花钱买教训了了吧
1567699	CHI	0	fc408a907a537d656822faf1ad6f6eea9f(	什么卖家，我买了两件就发我一件，找他解决还不回信息！快递还慢的要死，一星期才到。鞋子还没给我发！短裤...
1567700	CHI	0	b586b002e5fd9b9b92f1a1b73136cbb09	垃圾，扣子都没有，我也是醉了。问客服解决问题，都不理人，都是自动回复。
1567701	CHI	0	90bd34b85038ac7484bda72b462028c0	什么裤子喔，拉链都是坏的还敢寄出来，我的天哪，怎么做生意的
1567702	CHI	0	dac876ae3b8492ebaf80b6f014d86eacb	色差严重 而且裤子裤腿超级大 味道也大
1567703	CHI	0	5074475fe4875db740ae184e6741d83f2	第一次遇到这样的客服。物流也慢。我没话说。看聊天记录。这是我第一次买东西给差评。
1567704	CHI	0	f2ac4362de870470d1512e65c2aef09f1(	这种裤子的质量也是醉了。买的m码就像xxxl一样，而且裤子就像是用烂布改做成的。不忍直视！第一次买这...
1567705	CHI	0	258014009c4cad8baed763a69411a384(	就是服务态度超级差，物流超级慢，而且还少分了一条码数不正
1567706	CHI	0	151eeac2686e0312657d91b25bd8d289	有点色差 味道又很大 偏小
1567707	CHI	0	14a6fc76cf9450131a9e897e4f7a67d7bc	明明是深蓝色发过来变成浅蓝色 呵呵呵 而且没穿就起球
1567708	CHI	0	f8026868f649af9a3036c7db32a5ed90b(	差评差评 裤子买回来没有绳子，本来是应该有两个绳子系上的 可好不给我绳子 穿个屁啊 我也是倒霉了 *...
1567709	CHI	0	6e418afadea70170ab0db1e54b2404b1(	衣服都发错了 要的是浅蓝的发灰色的 醉了
1567710	CHI	0	e104de93c67b212ddc9bfd586e3f60912	裤子有很大的异味，然后商家居然还给我发错码子！只有物流还行！

Obr. 4.4: Výpis negativních komentářů z databáze

4.2 Možné úpravy programu

Jak bylo popsáno v předchozí podkapitole, funkce sleep a připojování přes proxy server může nástroj pro získávání komentářů `CommentsCrawler.java` značně zpomalit. Ve zdrojovém kódu je však možné tyto parametry upravit. Pokud by bylo potřeba program zrychlit, je možné změnit parametry pro počátek (`MIN_VALUE`) a konec (`MAX_VALUE`) intervalu pozastavení programu na nižší hodnoty. Například pět až deset sekund. Stejně tak by v poli proxy serverů mohla být pouze jedna konkrétní, zpravidla ta, na které je počítač běžně připojen k internetu. Po takových úpravách je ale třeba počítat s možností, že dané proxy může být po čase zablokován přístup.

5 ZÁVĚR

Tato práce se zabývá teoretickým základem, ale i samotným vývojem aplikace pro automatické získávání informací z webu. Nezbytnou součástí jsou dva programy, přičemž první dokáže ze zadaného čínského internetového obchodu stáhnout odkazy na všechny produkty. Druhý program je pak používá k vytvoření objemné databáze pozitivních a negativních komentářů z hodnocení jednotlivých produktů. Důvodem shromažďování těchto čínských textů je jejich následné využití pro trénování umělé inteligence. Podrobnější motivace je zahrnuta v úvodu práce.

V první kapitole je shrnuta stručná historie vývoje World Wide Webu. Je zde popsán základní protokol HTTP a značkovací jazyk HTML, díky kterému výrazně vzrostl zájem o použití internetu. Další částí je teorie o CSS, JSONu a AJAXu, na kterých zmíněné aplikace staví. Mimo to jsou zde popsáni i weboví pavouci a získávání informací z webu.

Druhá kapitola se zabývá základními vývojovými nástroji pro vytvoření aplikace. Ta byla naprogramována v jazyce Java. V této kapitole jsou také popsány knihovny jsoup a JSON-simple, které slouží jako základní kameny pro stahování komentářů. Technologie Selenium je uvedena hlavně proto, že i ona by mohla být použita jako nástroj pro automatické procházení e-shopu.

Další část práce se věnuje samotnému řešení. Jsou zde uvedeny poznatky z prvních testovacích pokusů o získávání komentářů pomocí knihovny jsoup na statické webové stránce. Pro dolování komentářů z dynamického obsahu muselo být řešení upraveno. V této části je popsán postup vypracování programu a jsou zde navíc zmíněny i okolnosti, které ztěžují jeho použití.

Po teorii a popisu samotné aplikace už jsou v kapitole čtyři uvedeny úspěšné výsledky tohoto automatizovaného nástroje. Ze zadaného e-shopu je možné automaticky ukládat kladné a záporné komentáře jednotlivých produktů do databáze. Při týdenním běhu aplikace bylo staženo 1 540 249 pozitivních a 30 305 negativních příspěvků. I přes některé problémy na straně internetového obchodu se tedy vývoj aplikace povedl a je schopna pracovat takřka nepřetržitě.

LITERATURA

[1] About The World Wide Web. *W3C* [online]. [cit. 2016-11-14]. Dostupné z: <https://www.w3.org/WWW/>

[2] About W3C. *W3C* [online]. 2016 [cit. 2016-11-15]. Dostupné z: <https://www.w3.org/Consortium/>

[3] Advanced Web Scraper. *datascraping.co* [online]. [cit. 2017-05-10]. Dostupné z: <https://www.datascraping.co/chrome-app.aspx>

[4] Apifier - The web crawler that works on every website. *Apifier* [online]. [cit. 2017-05-10]. Dostupné z: <https://www.apifier.com/>

[5] A Short History of JavaScript. *W3C* [online]. [cit. 2017-04-15]. Dostupné z: <https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript>

[6] BRDLIČKA, B. Co dokáží stroje schopné hlubokého učení. *RVP* [online]. 2016 [cit. 2017-05-10]. Dostupné z: <http://clanky.rvp.cz/clanek/c/Z/20855/co-dokazi-stroje-schopne-hlubokeho-uceni.html/>

[7] BROWN, T. B., BUTTERS, K., PANDA, S. *HTML5 okamžitě: Ovládněte HTML5 za víkend*. Brno: Computer Press, 2014, 256 s. ISBN 978-80-251-4296-7.

[8] CASTRO, E., HYSLOP, B. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012, 440 s. ISBN 978-80-251-3733-8.

[9] CSS Tutorial. *W3C* [online]. 2015 [cit. 2016-11-15]. Dostupné z: <https://www.w3.org/Style/Examples/011/firstcss.en.html>

[10] Ecma international. *STANDARD ECMA-404: The JSON Data Interchange Format* [online]. [cit. 2016-11-23]. Dostupné z: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

[11] Googlebot. *support.google.com* [online]. [cit. 2017-05-08]. Dostupné z: <https://support.google.com/webmasters/answer/182072?hl=en>

[12] HOLDENER III., A. T. *Ajax: The definitive guide*. Beijing: O'Reilly, 2008, 982 s. ISBN 978-059-6554-972.

[13] HOUSTON, P. *Instant jsoup how-to effectively extract and manipulate HTML content with the jsoup library*. Birmingham: Packt Publishing, 2013, 38 s. ISBN 978-178-2167-990.

[14] HTML and XHTML. *W3Schools* [online]. [cit. 2016-11-16]. Dostupné z: <http://www.w3schools.com/html/html_xhtml.asp>

[15] Hypertext Transfer Protocol - HTTP/1.1, RFC 2068. *IETF Tools* [online]. 1997 [cit. 2016-11-14]. Dostupné z: <https://tools.ietf.org/html/rfc2068>

[16] Hypertext Transfer Protocol - HTTP/1.1, RFC 2616. *IETF Tools* [online]. 1999 [cit. 2016-11-14]. Dostupné z: <https://tools.ietf.org/html/rfc2616>

[17] Hypertext Transfer Protocol Version 2 (HTTP/2). *IETF Tools* [online]. 2015 [cit. 2016-11-14]. Dostupné z: <https://tools.ietf.org/html/rfc7540>

[18] Introducing JSON. *JSON* [online]. [cit. 2016-11-22]. Dostupné z: <http://www.json.org/>

[19] JACKSON, W. *JSON Quick Syntax Reference*. Lompoc: Apress, 2016, 142 s. ISBN 978-148-4218-624.

[20] JEŘÁBEK, J. *Komunikační technologie*. V Brně: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013, 172 s. ISBN 978-80-214-4713-4.

[21] JEŘÁBEK, J. *Pokročilé komunikační techniky*. V Brně: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015, 193 s. ISBN 978-80-214-4636-6.

[22] JSON-simple. *Google Code: Archive* [online]. [cit. 2016-11-24]. Dostupné z: <https://code.google.com/archive/p/json-simple/>

[23] Jsoup: Java HTML Parser. *Jsoup* [online]. 2015 [cit. 2016-11-15]. Dostupné z: <https://jsoup.org/>

[24] KEOGH, J. *Java bez předchozích znalostí*, Brno: CP Books, 2005, 274 s. ISBN 80-521-0839-2

[25] KLÍMA, V. Hašovací funkce, principy, příklady a kolize. *cryptography.hyperlink.cz* [online]. 2005 [cit. 2017-04-15]. Dostupné z: <http://cryptography.hyperlink.cz/2005/cryptofest_2005.htm>

[26] National Institute of Standards and Technology. *SECURE HASH STANDARD* [online]. [cit. 2017-04-15]. Dostupné z: <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf>

[27] Norconex HTTP Collector. *Norconex* [online]. [cit. 2017-05-13]. Dostupné z: <http://www.norconex.com/collectors/collector-http/>

[28] RAJMIC, P. *Základy počítačové sazby a grafiky*. V Brně: Vysoké učení technické, 2012. ISBN: 978-80-214-4451-5

[29] RYAN, M. *Instant web scraping with Java*. Birmingham: Packt Publishing, 2013, 72 s. ISBN 978-1-84969-688-3.

[30] Selenium Projects. *SeleniumHQ* [online]. [cit. 2016-11-20]. Dostupné z: <http://www.seleniumhq.org/projects/>

[31] SHENOY, A. *Thinking in CSS*. Packt Publishing, 2014, 24 s. ISBN 978-17-835-5266-5.

[32] ŠIMKO, M. Princip fungování fulltextových vyhledávačů I. *Programujte.com* [online]. 2014 [cit. 2016-11-19]. Dostupné z: <http://programujte.com/clanek/2014010200-princip-fungovani-fulltextovych-vyhledavacu-i-crawler/>

[33] The Googlebot guide. *Varvy SEO tool* [online]. [cit. 2017-05-08]. Dostpné z: <https://varvy.com/googlebot.html>

[34] The JavaScript Object Notation (JSON) Data Interchange Format. *IETF Tools* [online]. [cit. 2016-11-22]. Dostupné z: <https://tools.ietf.org/html/rfc7159>

[35] Uniform Resource Locators (URL). *IETF Tools* [online]. 1994 [cit. 2016-11-14]. Dostupné z: <https://tools.ietf.org/html/rfc1738>

[36] Usage of HTTP/2 for websites. *W3Techs* [online]. [cit. 2017-04-28]. Dostupné z: <https://w3techs.com/technologies/details/ce-http2/all/all>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AJAX	asynchronní JavaScript a XML – Asynchronous JavaScript And XML
API	rozhraní pro programování aplikací – Application Programming Interface
CERN	Evropská organizace pro jaderný výzkum – Conseil Européen pour la recherche nucléaire
CSS	kaskádové styly – Cascading Style Sheets
CSV	čárkami oddělené hodnoty – Comma Separated Values
DCT	diskrétní kosinová transformace – Discrete Cosine Transform
DNA	Deoxyribonukleová kyselina – Deoxyribonucleic acid
DOM	objektový model dokumentu – Document Object Model
DTD	definice typu dokumentu – Document Type Definition
EMC	elektromagnetická kompatibilita – Electromagnetic compatibility
FTP	protokol pro přenos souborů mezi počítači – File Transport Protocol
GNU	svobodný počítačový operační systém
HTML	hypertextový značkovací jazyk – HyperText Markup Language
HTTP	hypertextový transportní protokol – Hypertext Transfer Protocol
HTTPS	zabezpečený hypertextový transportní protokol – Hypertext Transfer Protocol Secure
IDE	integrované vývojové prostředí – Integrated Development Environment
IETF	komise pro technickou stránku internetu – Internet Engineering Task Force
JSON	JavaScriptový objektový zápis – JavaScript Object Notation
MD5	hašovací funkce – Message Digest algorithm
MIT	Massachusettský technologický institut – Massachusetts Institute of Technology
RFC	série publikací v informatice – Request For Comments
RSS	technologie pro odběr novinek z webu – Rich Site Summary
SGML	univerzální značkovací metajazyk – Standard Generalized Markup Language
SHA	hašovací funkce – Secure Hash Algorithm
SMTP	protokol pro přenos elektronické pošty – Simple Mail Transfer Protocol
SPDY	experimentální síťový protokol společnosti Google – Speedy
SQL	strukturovaný dotazovací jazyk – Structured Query Language
SSL	vrstva bezpečných socketů – Secure Sockets Layer
SVG	škálovatelná vektorová grafika – Scalable Vector Graphics
TCP	primární přenosový protokol – Transmission Control Protocol
TeX	program pro počítačovou sazbu
TLS	transportní zabezpečená vrstva – Transport Layer Security
TSV	tabulátorem oddělené hodnoty – Tab Separated Values
URL	lokátor zdrojů – Uniform Resource Locator
UTF	technická norma definující kódování a zpracování textů pro většinu písem – Unicode Transformation Format

W3C	mezinárodní konsorcium pro vývoj standardů WWW – World Wide Web Consortium
WHATWG	skupina navrhující nové HTML technologie – The Web Hypertext Application Technology Working Group
WWW	celosvětová síť – World Wide Web
XHTML	rozšířený hypertextový značkovací jazyk – Extensible HyperText Markup Language
XML	rozšířený značkovací jazyk – eXtensible Markup Language
XPS	formát dokumentů XML – XML Paper Specification
YAML	formát pro serializaci strukturovaných dat

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD

- `/` ..... kořenový adresář přiloženého CD
  - `.settings` ..... adresář nastavení
    - `org.eclipse.core.resources.prefs`
    - `org.eclipse.jdt.core.prefs`
  - `bin` ..... adresář tříd
    - `jsoup`
      - `CommentsCrawler.class`
      - `ProductsCrawler.class`
      - `PublicMethods.class`
  - `CommentsDB` ..... adresář testovací databáze
    - `CommentsDB.sql`
  - `src` ..... adresář zdrojových kódů
    - `jsoup`
      - `CommentsCrawler.java`
      - `ProductsCrawler.java`
      - `PublicMethods.java`
  - `.classpath` ..... soubor CLASSPATH pro eclipse
  - `.project` ..... soubor PROJECT pro eclipse
  - `Bakalářská práce - Jakub Poliak.pdf` ..... bakalářská práce v PDF
  - `CommentsCrawler.jar` ..... aplikace pro stahování komentářů
  - `commons-lang3-3.5.jar` ..... knihovna commons-lang
  - `java-json.jar` ..... knihovna java-json
  - `json-simple-1.1.jar` ..... knihovna json-simple
  - `jsoup-1.10.1.jar` ..... knihovna jsoup
  - `mysql-connector-java-5.1.40-bin.jar` ..... knihovna mysql-connector
  - `ProductsCrawler.jar` ..... aplikace pro stahování linků

B MANUÁL KE SPUŠTĚNÍ APLIKACÍ

Tento manuál slouží k programu pro získávání linků `ProductsCrawler.jar` a programu pro získávání komentářů `CommentsCrawler.jar`. Před spuštěním aplikací je důležité mít na operačním systému Windows nebo Linux nainstalovanou lokální databázi. Pro vzdálenou databázi by se musel změnit `localhost` v metodě `getConnection()` ve zdrojovém kódu `PublicMethods.java` na její IP adresu. Následně by bylo nutné znovu zkompilovat `ProductsCrawler.java` i `CommentsCrawler.java` v nějakém vývojovém prostředí (například eclipse).

B.1 Program pro stažení linků

Pokud ještě nebyla vytvořena databáze `commentsdb` a naplněna tabulka `products`, je nutné nejprve spustit `ProductsCrawler.jar`. V operačním systému Windows je jednou z možností, jak program spustit, použití příkazové řádky. Pomocí příkazů pro procházení je třeba přejít do kořenového adresáře CD s aplikací `ProductsCrawler.jar` a napsat příkaz `java -jar ProductsCrawler.jar`, čímž se spustí stahování odkazů do databáze. Program se ukončí po stažení všech linků nebo po stisknutí kláves `Ctrl+C`.

Stejný postup lze použít i na operačním systému Linux, ale spouštění se provádí v linuxovém terminálu, ne v příkazovém řádku.

B.2 Program pro stažení komentářů

Program `CommentsCrawler.jar` se spouští stejně jako předchozí příklad, pouze se příkaz mění na `java -jar CommentsCrawler.jar`. Před samotným spuštěním musí být ale vytvořena databáze s tabulkou `products` z předchozí aplikace. Program po spuštění běží tak dlouho, dokud nachází linky v tabulce `products`, dokud není manuálně ukončen uživatelem nebo dokud není databáze plná.