Sem vložte zadání Vaší práce.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra ... (softwarového inženýrství)

Bakalářská práce

InfoWeb - Nástroj získávání informací z webů

Vedoucí práce: Ing. Jiří Hunka

15. dubna 2017

Poděkování

Chtěl bych poděkovat za trpělivost vedoucímu, Ing. Jiřímu Hunkovi.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. dubna 2017 .....

Odkaz na tuto práci

Tuček, Jakub. *InfoWeb - Nástroj získávání informací z webů* . Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

Klíčová slova Nahraďte seznamem klíčových slov v češtině oddělených čárkou.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahraďte seznamem klíčových slov v angličtině oddělených čárkou.

Obsah

Úvod **1**

1 Cíl práce **3**
1.1 Analytické cíle 3
1.2 Praktické cíle 3

2 Popis problematiky získávání informací z webů **5**
2.1 Problematika 5
2.2 Výběr dat 6
2.3 XML Path Language 6
2.4 CSS Selector 6
2.5 Současný stav řešení potřeb internetových obchodů 7

3 Analýza týmového projektu **11**
3.1 Cíl týmového projektu 11
3.2 Webové rozhraní 11
3.3 Interní část 12

4 Vývoj a implementace týmového projektu **13**
4.1 Pojmy 13
4.2 Vývoj 14
4.3 Implementace 15
4.4 Má role 16

5 Zhodnocení týmového projektu **17**
5.1 Pojmy 17
5.2 Stav 17

6 Návrh na vylepšení **19**

7 Realizace vylepšení **21**

8 Zhodnocení provedených vylepšení **23**
8.1 Analýza nového řešení 23

9 Realizace **25**
9.1 Způsob realizace stávajícího řešení 25
9.2 Implementace vylepšení 25

Závěr **27**

Literatura **29**

A Seznam použitých zkratek **31**

B Obsah přiloženého CD **33**

Seznam obrázků

Úvod

V předmětech BI-SP1 a BI-SP2 v prostředí FIT ČVUT byl realizován týmový projekt umožňující získávání informací z webů s primárním zaměřením na potřeby obchodů. Projekt řešil problém automatizace získávání dat z webů, jelikož stávající služby neposkytují veřejné rozhraní nebo mají velkou chybovost dat.

Práce pojednává o požadavcích internetových obchodů, které jsou především tvořeny nutností držet krok s trhem a sledovat vývoj cen prodávaných produktů u konkurence.

Cílem této práce je popsat požadavky internetových obchodů, stávající stav a možná řešení. Dále na základě těchto poznatků zhodnotit vytvořené řešení a včetně korektnosti zvolených postupů navrhnout vylepšení. Ty implementovat, řádně vylepšení otestovat a zhodnotit výsledný stav projektu.

Kapitola 1

Cíl práce

1.1 Analytické cíle

1. Rešerše aktuálního stavu získávání dat pro potřeby internetových obchodů
2. Analýza vzniklého řešení týmového projektu vzniklého v prostředí ČVUT FIT, včetně důrazu na použité postupy při softwarovém vývoji
3. Návrh a zhodnocení implementovaných vylepšení

1.2 Praktické cíle

1. Implementace vylepšení systému

Kapitola 2

Popis problematiky získávání informací z webů

V této kapitole se budu nejprve zabývat samotnou problematikou získávání informací z webů s důrazem na internetové obchody. Jelikož je tato problematika již řešena existujícími službami, je nejprve nutné služby zhodnotit a zanalyzovat jejich funkcionalitu.

2.1 Problematika

Získávání informací z webů je efektivní možnost jak získat databázi informací, které se na internetu vyskytují. Tato činnost však stojí na problematice data získávat a uchovávat v potřebné struktuře, jelikož jinak z dat nejsme schopni vyčíst potřebné informace. Vzhledem k specificitě dat, které jsou v kontextu činnosti zajímavá a dále kvůli unikátnosti webových stránek není možné jednoznačně určit jednotný a zcela automatizovaný postup, jak data získat v požadovaném formátu.

2.2 Výběr dat

Nejčastější řešení je kombinace automatizace a prvku lidské inteligence. To je obvykle dosaženo roboty, kteří data stahují a lidské práce určující jaké informace nás ve stažených datech zajímají.

Získávání informací ze stažených stránek lze poté zjednodušit na problematiku určení elementů v HTML, které jsou pro nás zajímavé. Lokaci elementu v HTML se kterým je potřeba pracovat lze poté jednoznačně určit pomocí dvou možností:

1. XPath
2. CSS Selector

2.3 XML Path Language

XML Path Language[1] nazývaný zkráceně XPath je jazyk, který slouží k výběru elementu v dokumentu ve formátu XML[2] .

XML chápeme jako jazyk popisující strukturu dat, které jsou strojově i lidsky čitelné. HTML lze chápat jako strukturu podobnou XML, ačkoliv se přímo o XML dokument nejedná [3]. HTML popisuje obsah dat pro prezentaci ve webovém prohlížeči pomocí předem definované struktury, které prohlížeče rozumí. Díky této vlastnosti lze použít XPath pro definování cesty k prvku (a jeho obsahu), který uchovává potřebnou informaci na webové stránce.

2.4 CSS Selector

Jazyk CSS je používán pro vizuální popis prezentace webové stránky definované v HTML. Jazyk k určení prvků se kterými chce pracovat používá selektory, které označují prvek v HTML. Jako selektor může být použit jak samotný název prvku, tak vlastní definované třídy.[4]

Pomocí řetězení těchto selektorů jsme schopni jednoznačně získat element v HTML.

2.5 Současný stav řešení potřeb internetových obchodů

I v kontextu malého trhu jako Česká republika se lze bavit o velké konkurenci na poli maloobchodů prodávající své zboží na internetu. Internetové obchody potřebují monitorovat konkurenci a trh. Vzhledem k jejich zaměření je tedy nejvíce zajímají obchody prodávající stejné zboží. Potřebné informace o prodávaných produktů konkurencí se skládají z následujících hlavních atributů:

1. Cena
2. Inzerovaný název
3. Dostupnost

2.5.1 Srovnávače cen

Data lze získat pomocí srovnávačů cen jako *zbozi.cz*[5] nebo *heureka.cz*[6]. Problém u těchto spočívá v určení pro koncové zákazníky, kterým umožňuje pro nalezení nejlepší ceny na trhu pro hledaný produkt. S tím souvisí to, že největší srovnávače cen neposkytují data nebo rozhraní přes která by je bylo možné jednoduše získat.

2.5.2 Existující služby

Problematiku sledování trhu s důrazem na firemní klientelu, řeší aktuálně několik existujících služeb.

Služby mají v zásadě velmi podobnou povahu služeb. Rámcově se jedná o porovnávání cen včetně historie na různých internetových obchodem či na srovnávačích cen. Uživatel si zadá okruh či seznam produktů, buďto formou manuální či vstupem ze souboru, případně přímých napojením na e-shop. Následně je možné konkrétní data zobrazit v grafech označující vývoj cen, trendů či náhlých změn. Dále umožňují externí výstup do souboru v dostupných formátech.

Největší rozdíl služeb je zda jsou data získávána přímo z obchodů nebo ze srovnávačích. Další odlišností je možnost zda služba dokáže sledovat i zahraniční trh.

Cena služeb se nejvíce odvíjí od počtu sledovaných produktů a četnosti aktualizací. Proto se měsíční platby mohou pohybovat od stovek korun po desítek tisíc korun.

2.5.2.1 Price checking

2.5.2.2 Pricing intelligence

2.5.2.3 Sledování trhu

2.5.2.4 Pricebot

2.5.2.5 Zahraniční nástroje

Tyto nástroje jsou obecněji zaměřené a obvykle požadují od uživatele techničtější zaměření, jelikož je nutné přesně specifikovat kde, co a jak chce sledovat. Vzhledem k tomuto omezení nejsou přímo pro provozovatele e-shopů vhodné kvůli nedostatečným technickým kapacitám a pro tuto práci důležité.

Bodový seznam zahraničních nástrojů:

1. Screen scraper [7]
    - Webová služba
    - procházení web skrz odkazy
    - potvrzování formulářů
    - využití interního vyhledávání
    - export do širokého množství formátu souborů
    - cena: $549 - $2,799 za měsíc
2. Web extractor [8]
    - Windows Aplikace
    - procházení zadaných stránek
    - hledání stránek pomocí klíčových slov
    - export do csv formátu
    - cena: $99 - $199 jednorázově
3. Web Scraper [9]

Kapitola 3

Analýza týmového projektu

V kapitole analýza týmového se budu věnovat řešení vytvořeného v rámci školní výuky na ČVUT FIT v akademickém roce 2015/16. Nejprve popíšu cíl, který měl projekt za úkol řešit, jaké byly použité postupy při vývoji a mou roli v tomto projektu.

3.1 Cíl týmového projektu

V předmětech BI-SP1 a BI-SP2 byl realizován týmový projekt, v souladu s osnovami těchto předmětů byl nejdříve v BI-SP1 vytvořen návrh systému a v BI-SP2 implementován.

Hlavní funkcionalita systému spočívá v maximální možné míře automatizace získávání informací o produktech prodávaných konkurencí. Důraz je především kladem na optimalizaci počtu nutných lidských úkonů v rukách administrátora u kterého se předpokládá minimální technické vzdělání. Jediná nutná problematika, co musí administrátor znát je parsování HTML stránek.

Návrh popisuje rozdělení aplikace na část poskytující veškeré webový rozhraní a na část zpracovávající všechny interní procesy. Vzhledem k požadavkům na škálovatelnost aplikace, druhá část se skládá z více samostatných menších služeb - modulů komunikující spolu pomocí front. Díky tomu, že každý modul zajišťuje určitou funkcionalitu umožňující vytvářet více jeho instancí, je možné procesy zpracovávat paralelně. Uživatelská a interní část spolu sdílejí data pomocí relační databáze[10].

3.2 Webové rozhraní

Webové rozhraní lze rozdělit na dvě části. Uživatelskou část obsahující množinu podstránek určených pouze pro konečné uživatele služby. Uživatelská část umožňuje vytvořit kampaň. Kampaň je proces trvající určitý časový úsek, který sleduje vložené produkty u konkurence. V těchto běžících kampaních

má poté uživatel možnost uživatel vidět vizualizaci získaných dat, případně je umožněn export dat do formátu csv nebo xlsx. Získaná data obsahují, kde se sledované produkty nacházejí a za jakou cenu se prodávají.

Druhá část je určena pouze pro administrátory a slouží k monitorování kampaní uživatelů a řešení chyb, které systém není schopný vyřešit. Chyby jsou typicky problémy s parsováním webových stránek, párování produktu ke stránce nebo potvrzení zda jsou získaná data validní.

3.3 Interní část

Interní část je rozdělena do samostatných modulů, které spolu komunikují pomocí front. Moduly jsou detailně popsány v následujících podsekcích.

3.3.1 Manager

Manager je hlavní modul, který jako jediný má možnost připojení přímo do sdílené databáze a jeho instance může existovat pouze jednou. Manager má za úkol plánování práce pro ostatní části systému a zpracování vstupů a výstupů z front.

3.3.2 Finder

Finder je modul, který má za úkol získávat URL adresy internetových obchodů a na nich vyhledávat URL adresy vedoucí na požadované detaily produktů. Detailem produktu je myšlena webová stránka, kde jsou obsažený podrobné informace o prodávaném produktu. Typicky je na takové stránce pouze jeden produkt, případně odkazy na jiné detaily prodávaného zboží.

3.3.2.1 DataProvider

DataProvider je module, který zpracovává adresy vedoucí na detaily produktu. Po stažení stránky, se z ní pokusí získat požadované hodnoty. V případě neúspěchu odešle chybovou hlášku, v opačném případě data zanalyzuje korektnost získaných cen vůči historickým datům pokud existují. Výsledek je poté odeslán k zpracování „Managerem“

Kapitola 4

Vývoj a implementace týmového projektu

Nakonec rozeberu výslednou funkcionalitu oproti nárokům na plně funkční systém.

4.1 Pojmy

4.1.1 Verzovací systém Git

Git je pro sdílení jednotlivých verzí každého vývojáře, umožňující jednoduchý přehled nad rozpracovanými částmi každého vývojáře. Úložiště systému se nazývá repositář, který obsahuje veškerý kód. Základní jednotkou tvoří verze, které jsou postupně vytvářeny vývojáře po vytvoření každé malé funkcionality. Verze jsou poté uchovávány v jednotlivých větví programu. Vedlejší větve slouží pro samotnou postupnou práci. Hlavní větve poté tento kód spojují. Git také zajišťuje základní nástroje pro slučování kódu v případě spojování nebo slučovaní vedlejších větví do větví hlavních.

4.1.2 Jednotkové a integrační testy

Jednotkovými testy se rozumí sada kladných a záporných testů ověřující funkcionalitu jedné třídy. Integrační testy pokrývají komunikaci více tříd.

4.1.3 Statická analýza kódu

Statická analýza kódu je analýza softwarového produktu, která běží bez spuštění samotné aplikace. Kontroluje pouze samotný kód. Označuje kritické konstrukce vedoucí k chybám nebo nedodržení programátorských konvencí daného jazyka.

4.1.4 Průběžná integrace

Průběžnou integrací se rozumí sada nástrojů sloužící k rychlému nalezení případných chyb. Základní součástí je vzdálený repozitář, kde na základě každé jeho změny se vytváří nový build. Při buildu jsou poté spuštěny jednotkové a integrační testy. Na jejich základě je poté možné zjistit případné chyby.

4.2 Vývoj

Vývoj byl rozdělen do 5 iterací, z nichž každá obsahovala 10 sprintů. Před začátkem vývoje byla rozdělena práce do těchto částí, s tím, že na konci každé iterace probíhala prezentace vyučujícímu. Každý sprint se skládal z jednotlivých úkolů, které byly přiřazeny členům týmu. Stav úkolů byl uchováván na systému Redmine, ten umožňuje přehledné řízení projektu a možnost sledování objevených chyb.

Jako verzovací systém byl zvolen systém Git, zajišťován službou Gitlab. Gitlab umožňuje možnost uchovávat repozitář na vzdáleném serveru a poskytuje webové rozhraní pro snadnou správu. Projekt byl v rámci repozitáře rozdělen na 4 části (větve):

- Master - hlavní větev uchovávající verze určené k nasazení na produkční server
- Develop - vývojová větev obsahující aktuální stav vývoje
- Feature - vedlejší větev vytvořená pro konkrétní úkol přidávající novou funkcionalitu
- Fix - vedlejší větev určená pro úkoly opravující naleznou chybu

Jelikož přístup k přidání verze do Master a Develop měl pouze vedoucí projektu, musel být pro každou Feature a Fix větve vytvořen požadavek o zařazení. Po kontrole vedoucím byl požadavek zařazen nebo vrácen k opravě.

Na konci každé iterace byla poslední verze vždy označená ve verzovacím systému a poté prezentována vedoucímu. Označení bylo zvoleno na základě pořadí iterace. 1. iterace je označena verzí „0.1“.

Pro vývoj se využil princip průběžné integrace. Každá verze byla zkompilována, otestována a zanalyzována na vzdáleném serveru. Tyto činnosti zajišťovaly systém Jenkins a Gitlab. Jenkins aplikaci zkompiloval, spustil testy a statickou analýzu kódu zajištěnou systémem SonarQube. Výsledky poté publikoval ve svém webovém rozhraní a zároveň v rozhraní Gitlab.

4.3 Implementace

4.3.1 Webové rozhraní

Webové rozhraní je implementováno v jazyce PHP verze 7. Základní kamenem aplikace zajišťuje aplikační rámec Nette. Nette obsahuje nástroje pro automatickou správu závislostí, komunikaci s databází, vytváření bezpečných formulářů, zabezpečení aplikace, šablonovací systém a rozhraní pro tvorbu jednotkových testů. Dále automaticky vynucuje MVC architekturu, která odděluje prezenční a logickou vrstvu.

TODO = OBRAZEK MVC

Snadnou správu závislostí umožňuje balíčkovací systém Composer. Na základě definujícího souboru, kde jsou uložený názvy požadovaných knihoven a jejich verze jsou automaticky stáhnuté z centrálního repositáře. Tím je mimo automatického stažení zajištěno, že každý člen týmu pracuje se stejnými verzemi knihoven.

4.3.2 Interní část

Interní část je implementována v jazyce Java verze 8. Kompilaci, spouštění testů a správu závislostí zajišťuje Gradle. Gradle je nástroj sloužící k automatickému sestavení aplikace. Umožňuje správu závislostí, které stahuje z centrálního repositáře. V rámci sestavení lze pustit testy, včetně přídavných doplňků. Projekt používá doplněk Cobertura, který na základě spuštěné testovací sady vytváří zprávu obsahující pokrytí větví programu. Díky tomu lze jednoduše zjistit jaké větve aplikace nejsou otestované.

Aplikace je rozdělena do nezávislých modulů běžící jako služby. Jednotlivé moduly spolu komunikují pomocí posílání zpráv definovaných front. Komunikaci zajišťuje systém RabbitMQ Server implementovaný v jazyce Erlang. Zprávy jsou serializovatelné objekty, jejichž definice je sdílena napříč všemi moduly. Serializace představuje proces, kdy je objekt serializovaný do posloupnosti bitů, které je poté možné poslat jako zpráva. Vzhledem k sdílené podobě objektu je poté možné na druhé straně objekt deserializovat zpět do Java objektu.

Použité knihovny:

- Google Guice - automatická správa závislostí
- Hibernate - objektově relační zobrazení databázových entit a práce s nimi
- Apache Commons - pomocné knihovny pro práci s řetězcemi a soubory
- RabbitMQ - zajišťuje komunikaci s frontami

4.4 Má role

Kapitola 5

Zhodnocení týmového projektu

Pro nutnost návaznosti na kapitolu o vylepšeních je nejprve nutné uvést v jakém kontextu jsou vylepšení navrhovány. K tomu je třeba popsat výsledný stav projektu a jeho funkcionalitu.

5.1 Pojmy

5.1.1 JSON

JSON označuje specifikaci formátu pro výměnu dat[11]. Jedná se o formát, který je čitelný jak pro lidské oko tak pro stroj[11] a jeho zpracování je implementováno pro velké množství programovacích jazyků[12]. Základní stavební jednotkou tvoří kolekce párů klíč a hodnotu spolu se seřazeným polem hodnot.

5.1.2 Web API

Rozhraní, které na definovaný dotaz vrací data, ke kterým má poskytující služba přístup. Data jsou standardně vracena ve formátech JSON nebo XML.

V kontextu této práce se jedná o rozhraní běžící v rámci webového rozhraní.

5.2 Stav

Ačkoliv stav projektu odpovídal nárokům na úspěšné odevzdání. Na druhou stranu nebyla dosažena implementace všech procesů, co by umožňovali reálné použití systému.

Odevzdávaný stav obsahoval funkční webové rozhraní, to se skládalo ze základní funkcionality pro uživatele a pro administrátory. Část pro administrátory obsahovalo správu uživatelů, možnost parsování stránek, evidenci známých obchodů a produktů či sledování chyb vzniklých v interní části. Uživatelská část umožňovala především správu a vytvoření kampaně či kampaní.

Jejich implementace splňovala návrh z analytické částí, která byla zmíněna výše. Kvůli potřebě jiného jsme vytvořili REST rozhraní poskytující získané data.

Kapitola 6

Návrh na vylepšení

TODO

Kapitola 7

Realizace vylepšení

Kapitola 8

Zhodnocení provedených vylepšení

TODO

8.1 Analýza nového řešení

TODO

Kapitola 9

Realizace

9.1 Způsob realizace stávajícího řešení

TODO

9.2 Implementace vylepšení

TODO

Závěr

Literatura

[1] Extensible Markup Language (XML) 1.0 (Fifth Edition). 2008. Dostupné z: https://www.w3.org/TR/2008/REC-xml-20081126/#sec-intro

[2] Virginia Tech - College of engineering: Department of computer science. 2002. Dostupné z: http://courses.cs.vt.edu/~cs1204/XML/htmlVxml.html

[3] HTML5: A vocabulary and associated APIs for HTML and XHTML. 2014. Dostupné z: https://www.w3.org/TR/html5/introduction.html#html-vs-xhtml

[4] Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. 2011. Dostupné z: https://www.w3.org/TR/CSS21/selector.html#q5.0

[5] Heuréka. Dostupné z: http://www.heureka.cz

[6] Zboží. Dostupné z: http://www.zbozi.cz

[7] Screen scraper. Dostupné z: http://www.screen-scraper.com

[8] Web extractor. Dostupné z: http://www.webextractor.com

[9] Web scraper. Dostupné z: http://www.webscraper.io

[10] The Java™ Tutorials. 2015. Dostupné z: https://docs.oracle.com/javase/tutorial/jdbc/overview/database

[11] Standard - ECMA - 404: The JSON Data Interchange Format. 2013: s. 1–14. Dostupné z: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

[12] Introducing JSON. Dostupné z: http://www.json.org/

Příloha **A**

Seznam použitých zkratek

XML Extensible markup language

HTML Hypertext Markup Language

CSS Cascading style sheets

JSON JavaScript Object Notation

Příloha **B**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `exe` ..... adresář se spustitelnou formou implementace
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF
  - `thesis.ps` ..... text práce ve formátu PS