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

Extrakcia dát z webu

[WebExtraction]

Dokumentácia k inžinierskemu dielu

Tím:	č. 16, WebX
Vedúci tímu:	Ivan Srba
Členovia tímu:	Ján Brechtl, Tomáš Juhaniak, Martin Kalužník, Rastislav Krchňavý, Michal Kren, Martin Lacek, Andrej Vaculčiak
Akademický rok:	2016/2017
Autor:	Ján Brechtl, Tomáš Juhaniak, Martin Kalužník, Rastislav Krchňavý, Michal Kren, Martin Lacek, Andrej Vaculčiak
Verzia číslo:	1.0
Dátum poslednej zmeny:	16.11.2016

1 Úvod

V dnešnom modernom svete plnom digitálnych technológií nepredstavuje až taký problém prístupnosť informácií, ako ich nekonzistentné uchovávanie, resp. ich reprezentácia. Poznáme rôzne druhy uchovávania informácií, či už pomocou textových alebo iných typov súborov. Keď z nich však chceme jednotlivé informácie získavať, nehovoriac o ich uchovávaní na jednom mieste, nastáva dosť ošemetná situácia, ktorá sa vyznačuje časovo náročným zberom dát z rôznych zdrojov a ich následná úprava do jednotného formátu, z ktorým sa následnej pracuje ďalej.

To isté platí aj pre webové služby. Aj keď aj v tejto oblasti existujú určité štandardy, stále to nie je len jeden, a teda je potrebné vedieť pracovať s viacerými alternatívami reprezentácie.

Tento problém otvára priestor pre realizáciu systému, ktorý bude vykonávať zber dát z rôznych webových serverov uchovávajúcich informácie v rôznej podobe. Jednou zo základných a veľmi užitočných vlastností by mala byť možnosť opakovaného zberu dát, nakoľko je dobré udržiavať dáta neustále aktuálne a kontrolovať ich manuálne je dosť nepraktické.

Obsahom dokumentu je okrem celkového pohľadu na systém aj detailnejší popis jednotlivých jeho modulov. Ku každému modulu sú uvedené informácie o analýze, návrhu a následnej implementácií, pričom sa nesmie vynechať testovanie, ktoré je ťažiskom pri tvorbe jednotlivých modulov.

2 Globálne ciele pre ZS

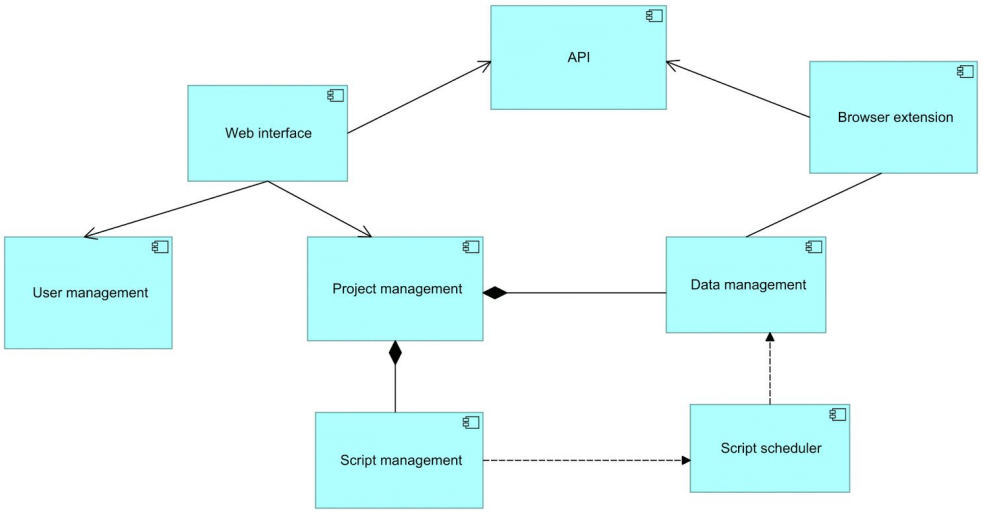
Keďže ide o tímový projekt, na ktorom spolupracujú ľudia, ktorí predtým spolu nepracovali, celý zimný semester, no najmä prvá polovica sa vyznačuje inicializáciou činností, dohadovaním a konfiguráciou mnohých komponentov a podporných nástrojov, no v neposlednom rade rozdelením si zodpovedností a začatím implementácie prvých verzií vybraných modulov systému.

Na konci tohto semestra by mal byť k dispozícii prvý funkčný prototyp, ktorý umožní vytvoriť jednoduchý skript, jeho uloženie a vykonanie s uložením extrahovaných dát do databázy.

Okrem implementácie modulov a funkcií systému sa väčšina tímu učí pracovať s Ruby on Rails, vytvárať testy a ďalšie aktivity súvisiace s projektom, ako napr. kontinuálna integrácia a správa webovej prezentácia projektu.

3 Celkový pohľad na systém

Podľa informácií od zákazníka, resp. vlastníka produktu je hlavnou funkciou systému automatizované zbieranie preddefinovaných dát z vybraných webových stránok. Používateľ si pomocou grafického rozhrania (webová stránka) môže spravovať vlastné projekty, skripty a dátové schémy, podľa ktorých sa vykonáva extrakcia. Anotácia dát na požadovanej stránke je realizované pomocou rozšírenia do webového prehliadača.



Systém ako taký je tvorený dvoma základnými časťami. Prvou je webová aplikácia poskytujúca prístup k účtu používateľa, s možnosťou správy skriptov na zber dát. Tak isto je možné spravovať svoj vlastný účet (napr. zmeniť heslo).

Druhá časť je už spomínané rozšírenie do prehliadača, pomocou ktorého používateľ jednoducho definuje, aké dáta a z ktorej stránky sa majú extrahovať.

4 Moduly systému

Vychádzajúc z predchádzajúcej kapitoly sa dekomponoval systém na menšie časti (moduly), ktoré sa následnej ešte podrobnejšie rozdeľujú na US (User stories) a tie na úlohy.

Identifikované sú tieto moduly (podrobnejšie info v jednotlivých podkapitolách):

- User management
- Project management
- Browser extension
- Data management
- Script management
- Script scheduler

Jednotlivé moduly postupne prechádzajú 4 základnými procesnými štádiami. Sú dekomponované na jednoduchšie US a tie prechádzajú procesom analýzy a návrhu zväčša priamo na stretnutí, vo fáze prideľovania US do šprintu.

Nasleduje samostatná fáza implementácie v šprinte, počas ktorej sa návrh môže meniť. Aby boli jednotlivé US uznané ako hotové, musí prebehnúť aj úspešné testovanie.

4.1 User management

4.1.1 Analýza

Na používanie aplikácie je potrebné mať vytvorené používateľské konto. Základom je modulu je prihlásenie a registrácia, obsahuje tiež možnosť spravovať tieto kontá a obmedziť prístup k určitým častiam/funkciám aplikácie.

4.1.2 Návrh

Každý používateľ obsahuje atribút “rola”, ktorý predstavuje jeho práva. Momentálne v systéme evidujeme dve roli - bežný používateľ a administrátor. Administrátor má navyše od bežného používateľa prístup k “Administrácií”, kde má možnosť upraviť niektoré atribúty ostatných používateľov, napr. rolu.

Na vytvorenie konta slúži registračný formulár, kde okrem povinných polí e-mail a heslo, má používateľ možnosť vyplniť meno a priezvisko. Po registrácií je nutné overiť si svoje konto pomocou inštrukcií, ktoré dostane na registračnú emailovú adresu. Bez potvrdenia účtu je možný prístup do aplikácie len 2 minúty od registrácie. Do aplikácie sa vstupu pomocou prihlasovacieho formulára a používateľ má možnosť zobraziť si svoj profil a nastavenia konta - zmena hesla, úprava atribútov, zrušenie konta.

4.1.3 Implementácia

Na správu používateľ sme použili gem ‘devise’, ktorý poskytuje registračný, prihlasovací a editačný formulár. Gem ‘cancancan’ rieši povolenia na základe rolí používateľa podľa nami definovaných pravidiel. Administrátor má nad používateľmi povolenie ‘manage’,

čo znamená kompletnú CRUD funkcionalitu, zatiaľ čo bežný používateľ si môže len zobraziť svoje vlastné konto. K administrácií ma používateľ prístup z hornej lišty aplikácie. Zobrazí sa mu zoznam vytvorených používateľov, spolu s dátumom registrácie a rolou. Odtiaľto si môže zobraziť profil alebo editáciu konkrétneho používateľa.

4.1.4 Testovanie

Pre správu používateľov sme vytvorili spolu 14 testov, ktoré pokrývajú všetky prípady použitia tohto modulu. V adresári spec/model sa nachádzajú unit testy pre vytvorenie objektu User, ktoré overujú prítomnosť povinných atribútov a unikátnosť emailovej adresy. V adresári spec/view sa nachádzajú testy pre prihlasovanie a registráciu, ktoré overujú správnosť a korektný formát hesla, potvrdenie emailovej adresy a základné prípady použitia prihlásenie a registrácia. Taktiež sa tu nachádzajú testy pre administráciu, ktoré overujú prístupové práva a zmenu roli používateľa.

4.2 Project management

4.2.1 Analýza

Princípom celej webovej extrakcie je sťahovanie dát z webu. Keďže chceme aby mali požívatelia pri určovaní dát na sťahovanie poriadok, bolo nutné vytvoriť miesto kde sa bude definovať každé extrahovanie z webu.

4.2.2 Návrh

Z výsledku analýzy sme dospeli k vytvoreniu 3 modelov a to projekt, dátová schéma a skript. Platí, že používateľ má mnoho projektov a jeden projekt patrí jednému používateľovi, projekt má mnoho skriptov a jeden skript patrí jednému projektu, projekt má mnoho dátových schém a jedna dátová schéma patrí jednému projektu. V tejto etape riešenie nebude limitovaný počet projektov skriptov a atribútov, ktoré si môže používateľ vytvoriť.

Najskôr si používateľ musí vytvoriť projekt, pre každý jeho prípad použitia. V prípade ak by chcel používateľ sťahovať rôzne typy dát, pre každý typ bude musieť vytvoriť samostatný projekt. K projektu teda logicky prislúcha dátová schéma. V dátovej schéme si používateľ definuje atribúty a ich typy. Po definovaní dátovej schémy si používateľ môže vytvoriť skripty pre všetky weby z ktorých chce extrahovať.

Používateľ môže mať napríklad 2 projekty pre extrahovanie dát z eshopu s notebookmi a pre extrahovanie dát z inzercie na bazári. V projekte extrahovania dát z elektronického bazára má používateľ definované atribúty cena, popis lokalita a telefónne číslo. Následne si vytvorí skripty pre každú bazárovú stránku o ktorú má záujem.

4.2.3 Implementácia

Pre projekty, skripty a dátové schémy sme najskôr vytvorili 3 modely spolu s prislúchajúcimi migráciami. V tejto fáze riešenia obsahovali projekty a skripty iba jeden atribút a to ich meno. Dátová schéma obsahovala okrem názvu aj typ, ktorý je

implementovaný ako enum (vymenovaný typ) s hodnotami integer (celé číslo), float (desatinné číslo) a string (reťazec znakov). Zabezpečenie projektov pred neoprávneným čítaním a úpravou má na starosti gem ‘cancancan’.

4.2.4 Testovanie

Pre účely testovania sme najskôr vytvorili factories (továrne - hodnoty v databáze určené iba pre testovanie). Následne sme napísal niekoľko testov, ktoré skontrolujú či sa správne zobrazujú všetky potrebné informácie o projektoch, skriptoch a dátových schémach.

4.3 Browser extension

4.3.1 Analýza

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

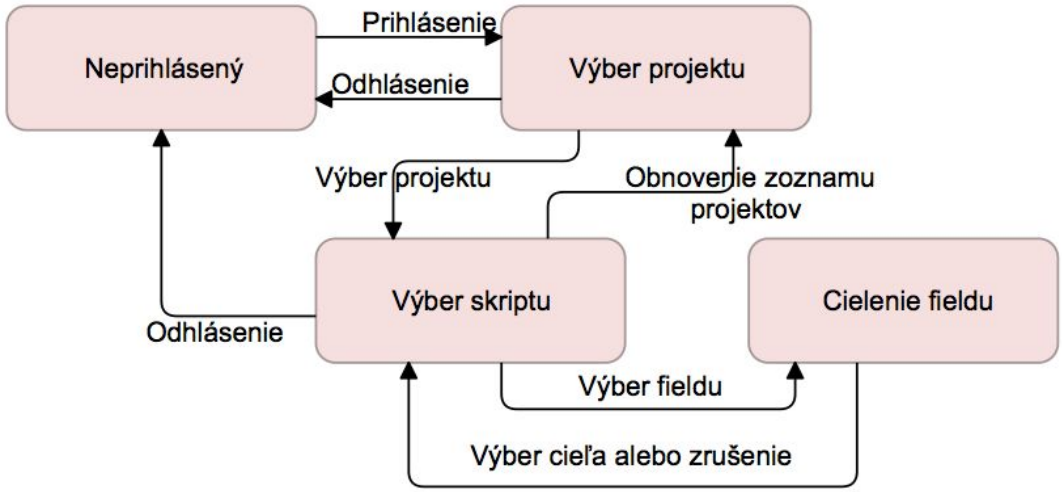
4.3.2 Návrh

Vytvorili sme jednoduché rozšírenie, ktoré umožňuje používateľovi prihlásiť sa a následne zobraziť svoje projekty, skripty a dátové schémy. Po kliknutí na prihlásenie sa zobrazí prihlasovacie okno do aplikácie. Po úspešnom prihlásení si môže používateľ rozkliknúť niektorý projekt a zvoliť si konkrétny skript.

4.3.3 Implementácia

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

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



4.3.4 Testovanie

Testovanie tohto modulu zatiaľ nebolo vyžadované (výnimka v DoD) a teda nebolo riešené.