Dokumentácia k inžinierskemu dielu

Tím 19 Nautilus - Vyhľadávanie so sémantikou

FIIT STU v Bratislave, 2016/2017

Autori: Bc. Jakub Hagara
Bc. Adam Rafajdus
Bc. Martina Redajová
Bc. Tomáš Repiský
Bc. Jozef Sitarčík
Bc. Martin Vaško

Obsah

Úvod

Sémantické vyhľadávanie, či inak povedané vyhľadávanie na základe významu použitého termínu, sa v súvislosti s nárastom počtu informácií, ale často aj potrebou nájsť rýchlo tie správne informácie, stáva nevyhnutnosťou. Úspešnosť vyhľadávania však závisí od dostupnosti zdrojov. Periodické dokumenty predstavujú bohatý a nenahraditeľný zdroj informácií. V súčasnosti, keď ich príprava, či samotné publikovanie je realizované v elektronickej podobe to nie je problém, čo sa však nedá povedať o tých starších. Digitalizácia nám umožňuje ich prevod do elektronickej podoby, ale ani to nie je dostatočné. Pre používateľa je dôležité, aby ho vyhľadávanie naviedlo na konkrétny článok a nie na celé číslo časopisu. Preto sme sa rozhodli v našom projekte zautomatizovať proces spracovania takto zdigitalizovaných periodík až na úroveň analytického rozpisu článkov.

Súčasťou procesu digitalizácie je rozpoznávanie znakov, k čomu sa využíva Adobe Recognition Software a digitalizované obrazy transformuje do špeciálnych XML dokumentov. Takéto súbory, ale aj bežné txt dokumenty, či word dokumenty sú tie, na ktoré sa zameriavame a ktoré sú na vstupe do nášho procesu. Tieto špeciálne XML súbory síce obsahujú množstvo informácií, ale sú to zväčša informácie týkajúce sa formátovania a úpravy textu, čo pre účely vyhľadávania nemá dostatočný význam. Našim hlavným cieľom v tomto projekte je, ako sme už uviedli, identifikovať tie informácie, ktoré nám umožnia správne rozpoznať názvy a k nim prislúchajúce texty konkrétnych článkov. Následne tieto texty spracovať a identifikovať v nich kľúčové slová, neskôr aj význam, v akom boli tieto slová v danom texte použité. Na základe takto získaných údajov generovať bibliografické záznamy pre jednotlivé čísla periodík, ako aj pre konkrétne články, vo formáte MARC21 podľa platných katalogizačných pravidiel, aby s nimi mohli priamo pracovať aj knižnično-informačné systémy. K jednotlivým bibliografickým záznamom tiež v našom digitálnom repozitári ukladáme aj plné texty a výrez zo zdigitalizovaného obrazu strany, ktorý zachytáva daný článok.

Keďže kvalita zdrojov a schopnosti OCR nástrojov rozpoznávať znaky nemusí byť stopercentná, je súčasťou nášho projektu tiež návrh a realizácia aplikácie, ktorá umožňuje zamestnancom inštitúcií pre archiváciu takýchto zdrojov editovať výsledky OCR spracovania, teda napríklad upravovať nepresnosti pri rozpoznávaní textu. Rovnako pomocou tejto aplikácie umožňujeme týmto správcom upravovať výsledky nášho algoritmu pre rozdelenie dokumentu na samostatné dokumenty - články, pretože vplyvom nekonzistencie tlače nie je možné garantovať 100%-nú úspešnosť rozpoznania článkov z týchto údajov.

Ciele projektu

Cieľ projektu:

- Vytvoriť nástroj pre kontrolu a editáciu spracovaných zdrojov na úrovni rozdelenia do článkov

Ciele zimného semestra:

- Rozpoznanie elementov v XML súborch
- Extrahovanie jednotlivých článkov v skenoch periodík
- Vytvorenie jednoduchej databázy článkov a zdrojových periodík tak, aby bolo možné vykonať kontrolu úspešnosti algoritmov extrakcie článkov

Ciele letného semestra:

- Vytvorenie používateľského prostredia pre nahratie a spustenie spracovania zdrojov
- Vytvorenie používateľského prostredia pre kontrolu a editáciu rozdelenia čísla do článkov a kontrolu bibliografických záznamov
- Vytvorenie aplikácie pre extrakciu MARC21 a obrazových záznamov pre jednotlivé články
- Identifikácia kľúčových slov pre jednotlivé články

Celkový pohľad na systém

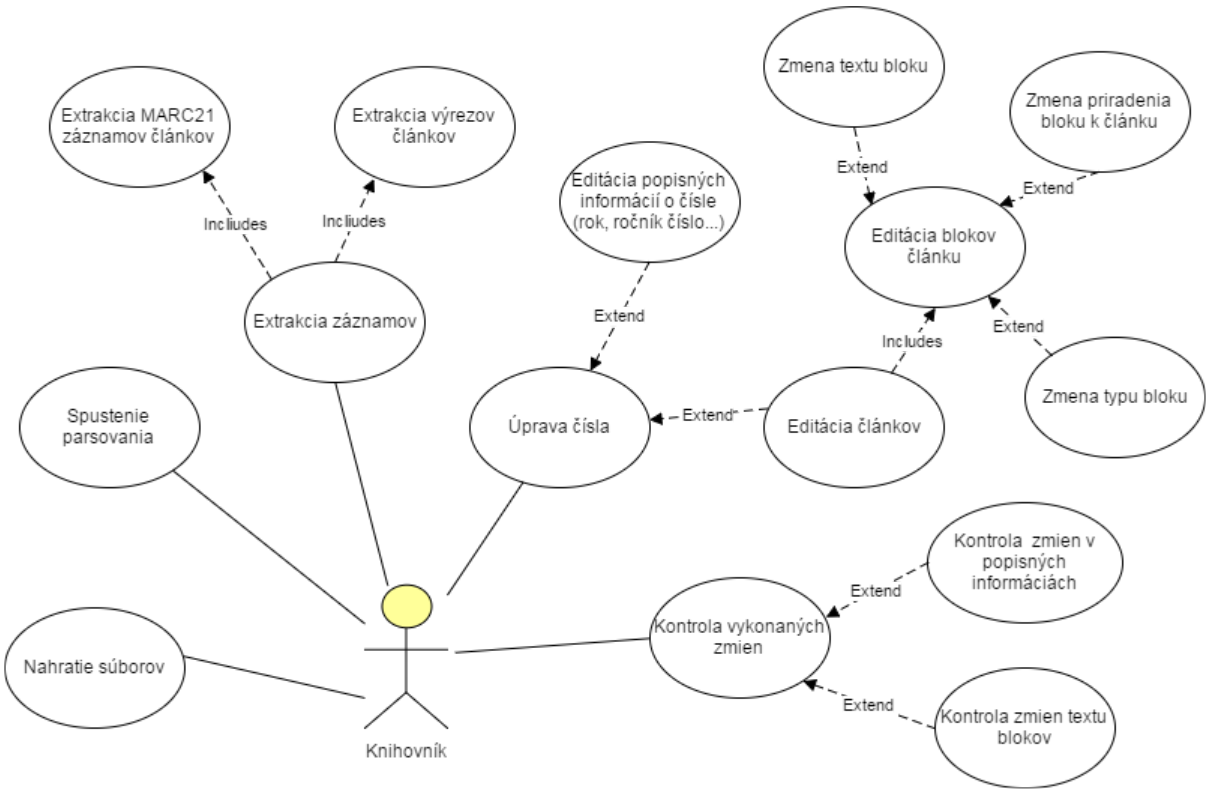
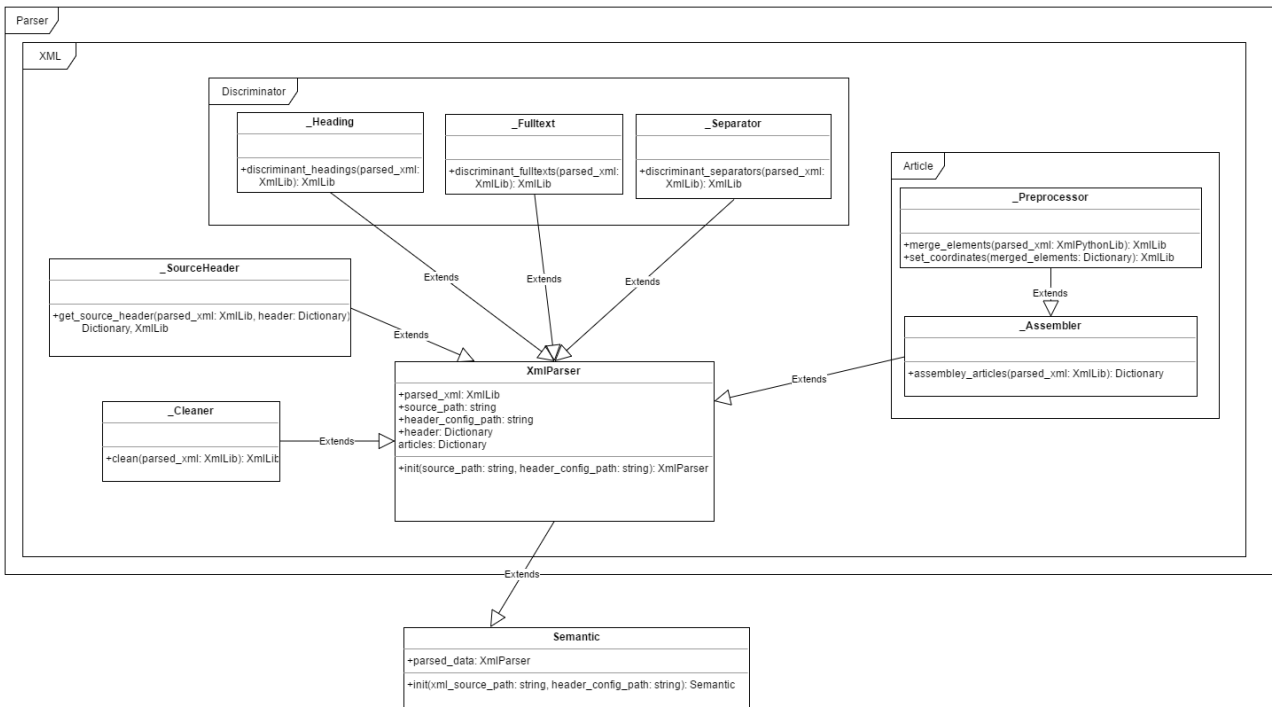


Diagram 1: Diagram prípadov použitia pre navrhnutý systém



Kostru systému tvorí parser, zatiaľ spracúvajúci vstupné XML súbory. Skladá sa z častí pre prečisťovanie súborov XML _Cleaner, získavanie údajov z hlavičky časopisu _SourceHeader, identifikácia častí xml (nadpisy, fulltexty a separátory - horizontálne grafické čiary) prostredníctvom moduu Discriminant a skladanie častí do článkov - spájanie nadpisov a fulltextov prostredníctvom modulu Article.
Link: *analyza.pdf, Navrhnutie štruktúry class diagramov*

Moduly systému

Analýza

Počas prvých šprintov sme sa zamerali na analyzovanie spôsobov ako získavať potrebné údaje o článkoch, ako nadpisy, či fulltexty zo vstupných XML súborov. Jednotlivé poznatky o pravidlách zadeľovania, prečisťovania, či spájania článkov sú opísané v súhrnnom PDF dokumente z prvých 3 šprintov.
Link: *analyza.pdf*

Návrh

Navrhli sme si algoritmus vychádzajúci z analýzy, ktorý extrahuje články z časopisov.
Referencia na dokument *analyza.pdf, Navrhnutie pseudokódu na prepájanie skupín elementov rôzneho typu.*
Navrhli sme class diagram programu a identifikovali sme jeho moduly. Navrhnutý class diagram programu je v predchádzajúcej časti Celkový pohľad na systém.
Výstup algoritmu uchovávame v databáze, pričom sme navrhli entito-relačný model pre referenciu databázy.

Implementácia

Pre implementáciu sme si vybrali jazyk Python, verzia 3.5 alebo novšie. Repozitár a základy verziovania sme vytvorili podľa preddefinovaných metodík. Vytvorili sme si virtuálne prostredie pre optimálnu prácu podľa metodiky. Pre vývoj používame prostredia IDE Pycharm alebo textového editoru Vim. Celý kód je kontrolovaný a dodržiava sa pri písaní metodika Code Quality - konkrétne sa dodržiava PEP8 (Viac v metodike).
Aplikácia je rozdelená na 3 hlavné časti - spracovanie základných XML súboru pomocou jednoduchého parsera pre získanie blokov textu, následne identifikácia daných blokov na typy Nadpis, Fulltext a Separátor. Posledná časť pomocou do reálneho kódu preloženého pseudokódu pospája

určené bloky do článkov definovaných ako nadpis a ich text. Tieto entity sú zapísané do NoSQL databázy Elastisearch.
S databázou elasticsearch pracuje naša webová aplikácia, ktorá je vytvorená vo frameworku RubyOnRails. Tá ponúka prostredie pre načítavanie, spracovávanie a zmeny a ukladanie článkov. Webová aplikácia prepája backend Python funkcionalitu pomocou skriptov, vytvorených v Python jazyku a faktu, že sa neustále komunikuje s databázou ElasticSearch.
Funkcionalita webovej aplikácie tak isto aplikuje istý fileserver, ktorý sa vytvára ako hierarchická postupnosť zložiek, do ktorej si vieme extrahovať záznamy MARC21 a obrázkové výrezy konkrétnych článkov. Viac o tejto funkcionalite v technickej dokumentácií.
Link: TP-DeepSearch-develpment.zip (export z Github repozitáru)
Link: TP-DeepSearch-rails-develpment.zip (export z Github repozitáru)
Link: Dokumentacia k riadeniu/Prílohy/B_Metodiky.pdf:Metodika verziovania
Link: Dokumentacia k riadeniu/Prílohy/B_Metodiky.pdf:Metodika code quality
Link: Technická dokumentácia

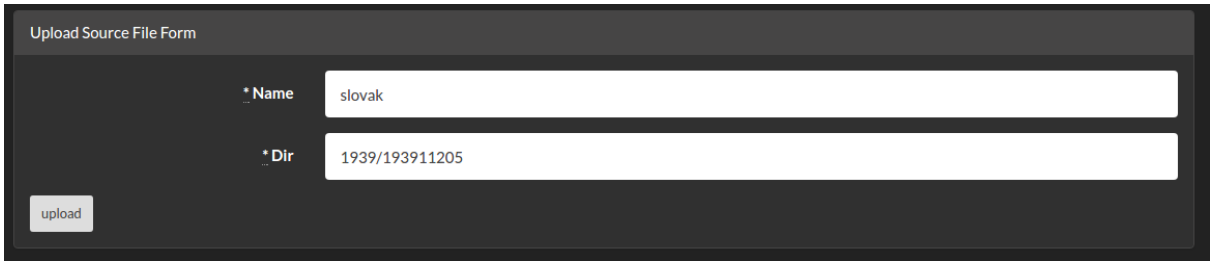
Testovanie

Písanie testov sme uskutočnovali pomocou metodiky písania testov. Testy sú teda písané pre každú funkciu a metódu. Snažili sme sa aj o TDD, no nie je to nevyhnutný spôsob vývoja a písania testov. Následne máme implementovaný nástroj pre kontinuálnu integráciu (CI), ktorý sa stará o spúšťanie testov v ideálnom testovacom prostredí, čo odľahčuje nároky na programátora tímu ešte viac.
Link: Dokumentacia k riadeniu/Prílohy/B_Metodiky.pdf:Metodika písania testov a testovania

Používateľská príručka

Kapitola 1: Parsovanie a výber čísla periodika

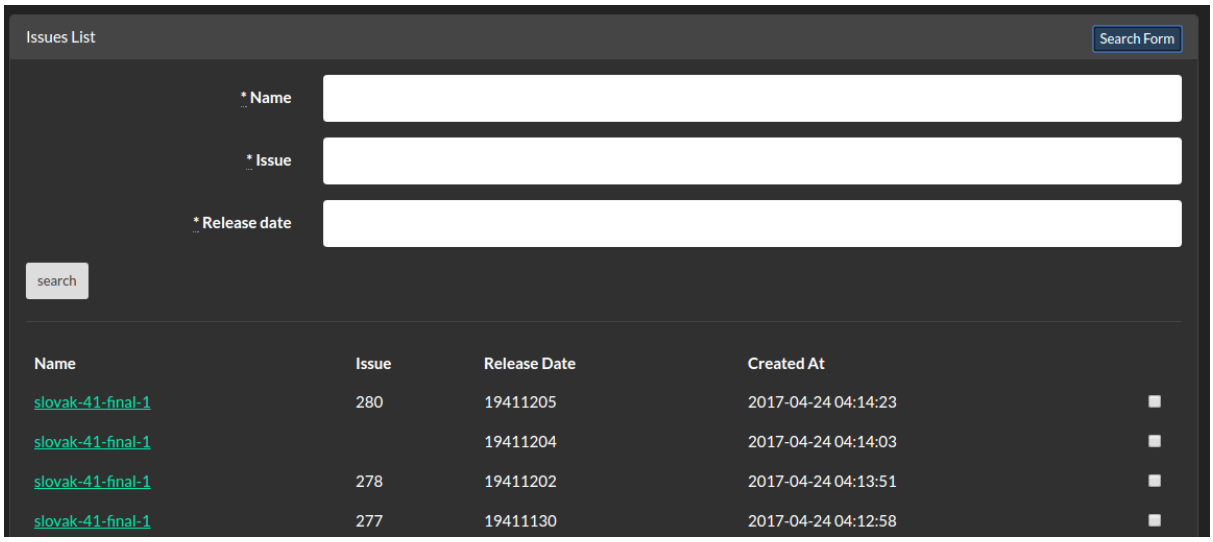
Parsovanie nového čísla časopisu



1. Prejdeme na domovskú stránku webovej aplikície
2. v hornej časti mame tento formulár pre upload nových čísiel “Issues” časopisu
3. Vyplníme meno časopisu (Nie je smerodajne, skôr slúži pre pomoc knihovníkovi)
4. Vyplníme cestu k cieľového issue
    a. Ak je napríklad pôvodná cesta k adresáru kde sú umiestnené všetky raw čísla časopisov napr. /var/lib/issues
    b. potom by na základe hore vyplneného formulára systém hľadal číslo v na tejto ceste “/var/lib/issues/1939/193911205” (teda spája pôvodnú cestu k adresáru so všetkými issues s cestou, ktorú zadáte vo formulári
    c. Ak by sme chceli parsovať napríklad všetky issues v roku 1939, potom nám stačí do formulára zadať cestu 1939. systém prehľadáva rekurzívne do hĺbky podľa priečinkov, takže takto by sa sparsovali všetky čísla v priečinku “/var/lib/issues/1939”

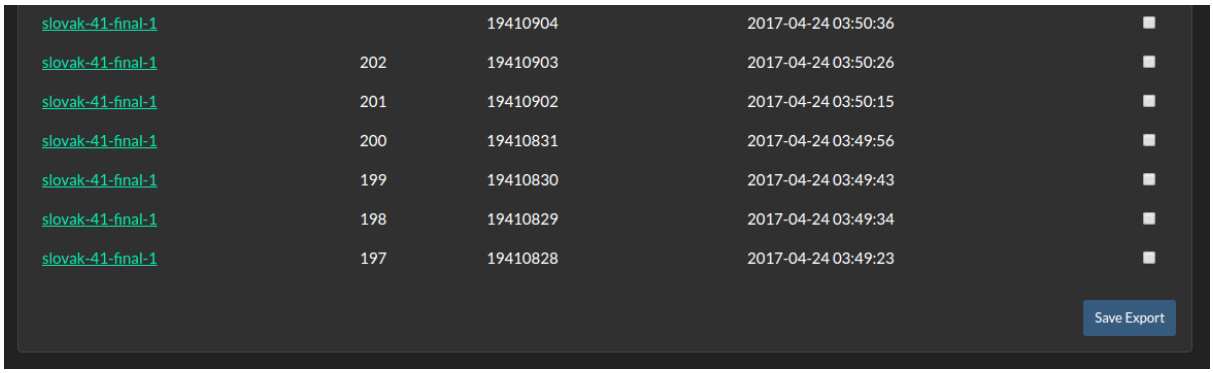
Kliknem na tlačidlo “upload”, parsovanie čo to trvá, záleží hlavne od množstva a veľkosti parsovaných issues

Vyhľadávanie čísiel periodika



1. Pre tieto účely sme vytvorili jednoduchý vyhľadávaci formulár taktiež na hlavnej stránke aplikácie
2. Kliknutím na tlačidlo “Search Form” sa nám otvorí inak skrytý formulár pre vyhľadávania
3. Vidíme v ňom rôzne polia, všetky sú nepovinné, vyhľadávanie funguje na jednoduchom princípe AND, teda s každým políčkom sa systému posiela príkaz napríklad kde “číslo issue je 45” A “dátum vydania je 19411205”
4. Kliknutím na tlačidlo “search” sa vyhľadajú žiadané čísla časopisov

Export vybraných čísiel periodika



1. Na domovskej stránke aplikácie môžme vidieť zoznam zobrazených issues
2. zakliknutím niektorých z nich pomocou na pravo umiestneného checkboxu si vyberieme tie, ktoré chceme exportovať
3. Následnej klikneme na tlačidlo “Save Export”
4. Na serveri, presne na miestach, kde sa fyzicky nachádzajú konkrétne čísla časopisov, tak tu sa vytvoria priečinky a súbory.

a. Priečinok Articles, ktorý obsahuje priečinky podľa čísiel článkov. V každom takomto priečinku sú umiestnené obrázky vystrihnutých textov priamo z originálnych obrúzkov raw issue. Ďalej je tu aj súbor predstavujúci MARC21 záznam tohto článku.

b. Ďalší súbor sa vytvorí priamo v ceste daného issue, predstavuje MARC21 záznam daného issue

<u>Príklad exportu a vzniknutých priečinkov a súborov je zobrazený nižšie</u>

issue
drwxr-xr-x 68 vasko vasko 4096 Apr 24 15:45 articles/
-rw-r--r-- 1 vasko vasko 1479 Apr 24 15:45 issue_marc21.xml
drwxrwxr-x 2 vasko vasko 4096 Apr 24 15:45 STR/
drwxrwxr-x 2 vasko vasko 4096 Apr 24 15:45 XML/

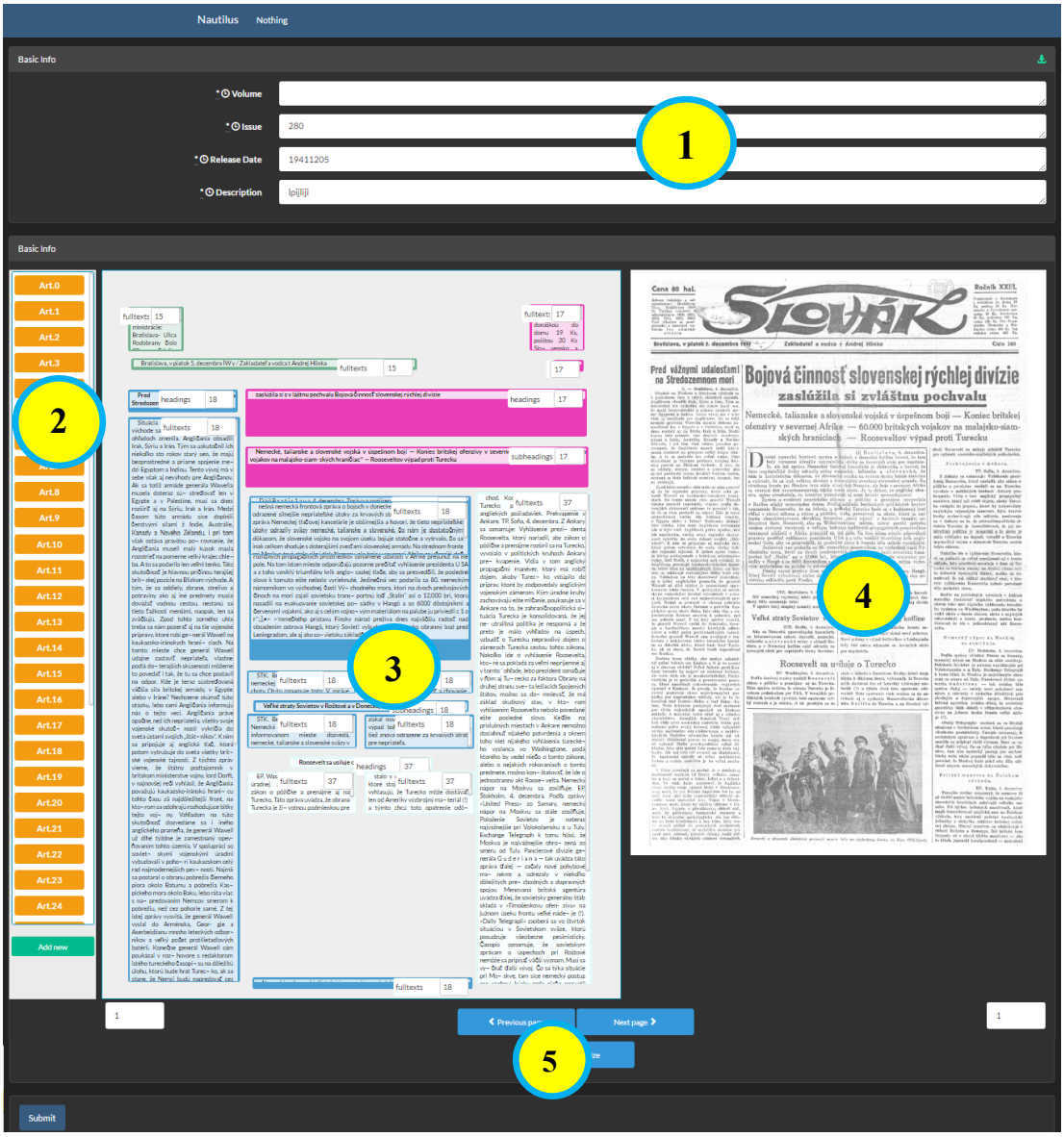
issue/articles
drwxr-xr-x 3 vasko vasko 4096 Apr 24 15:45 1/
drwxr-xr-x 3 vasko vasko 4096 Apr 24 15:45 10/
drwxr-xr-x 3 vasko vasko 4096 Apr 24 15:45 11/
drwxr-xr-x 3 vasko vasko 4096 Apr 24 15:45 12/
drwxr-xr-x 3 vasko vasko 4096 Apr 24 15:45 13/

```
drwxr-xr-x  3 vasko vasko 4096 Apr 24 15:45 14/
drwxr-xr-x  3 vasko vasko 4096 Apr 24 15:45 15/


# issue/articles/1
-rw-r--r--  1 vasko vasko 2051 Apr 24 15:45 1_marc21.xml
drwxr-xr-x  2 vasko vasko 4096 Apr 24 15:45 pictures/


# issue/articles/1/pictures
-rw-r--r-- 1 vasko vasko 421597 Apr 24 15:45 article_extract_page2.jpg
```

Kapitola 2: Kontrola a editácia konkrétneho čísla

Vitajte v časti aplikácii určenej pre kontrolu a editáciu bibliografických záznamov a rozdelenia časopisov do článkov. Aplikácia sa skladá z niekoľkých častí.

Časť 1 je určená pre zobrazenie a editáciu informácií o aktuálnom čísle. V časti 2 sa nachádza panel s tlačidlami odkazujúcimi sa na jednotlivé články nachádzajúce sa v čísle časopisu. Časť 3 je určená pre grafické zobrazenie článkov na jednotlivých stranách. Každý článok predstavuje niekoľko zafarbených blokov. V časti 4 sa nachádza náhľad do skutočnej podoby aktuálnej strany čísla. V dolnej časti aplikácie, v časti 5 sa nachádzajú tlačidlá na prepínanie medzi stranami čísla a uloženie zmien.



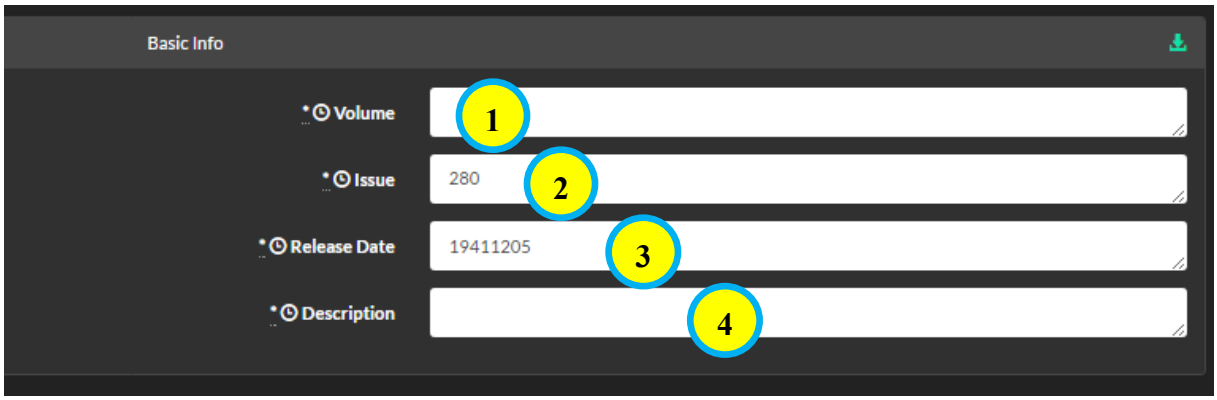
Kontrola a editácia bibliografického záznamu čísla

V hornej časti aplikácie ***Basic Info*** sa nachádza formulár, v ktorom sú zobrazené informácie o aktuálnom čísle. K týmto informáciám patrí:

- Informácia o ročníku v časti ***Volume*** 1

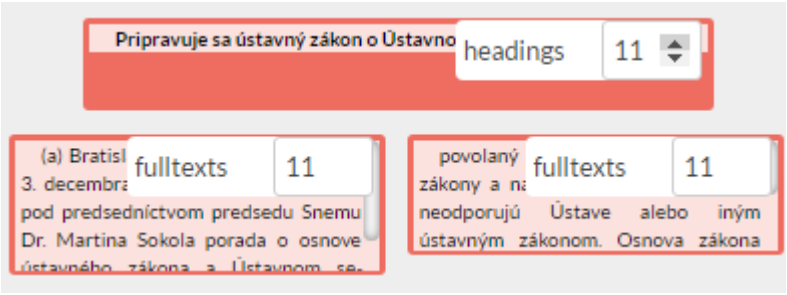
- Informácia o čísle v časti ***Issue*** 2
- Informácia o dátume vydania v časti *Release* ***Date*** 3
- Popis čísla v časti ***Description*** 4

Obsah týchto políčok je automaticky doplnený z MARC21 záznamu časopisu a hlavičky daného čísla. Avšak, ak by niektorý z týchto údajov nebol doplnený, každé z políčok je editovateľné.



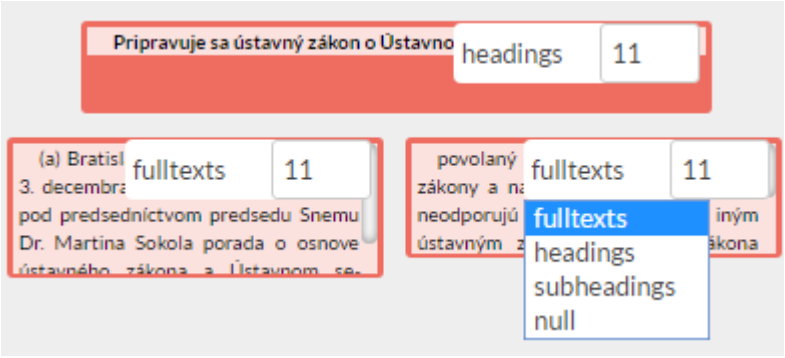
Editácia článkov

Pre zmenu príslušnosti bloku textu k článku použite číselný vstup nachádzajúci sa v pravom hornom rohu bloku. Vo vstup nastavte číslo článku, ku ktorému chcete daný blok priradiť. Po tejto zmene sa farebne zmení podfarbenie daného bloku. Toto číslo je možné zmeniť prepínaním šípok v pravej časti vstupu alebo prepísaním aktuálnej hodnoty.



Editácia typu bloku

Pre zmenu typu bloku použite rozbaľovacie menu v pravom hornom rohu. Hodnota, ktorá je menu zobrazená automaticky, je hodnota, ktorá znamená typ bloku. Hodnota ***headings*** znamená skupinu nadpisov. Hodnota ***fulltexts*** znamená text článku. Hodnota ***subheadings*** naznačuje skupinu podnadpisov. Ak chcete daný blok vymazať zo zoznamu blokov pre daný článok, nastavte hodnotu tohto poľa na ***null***. Takto označené bloky nebudú pri ďalšom načítaní čísla zobrazené.

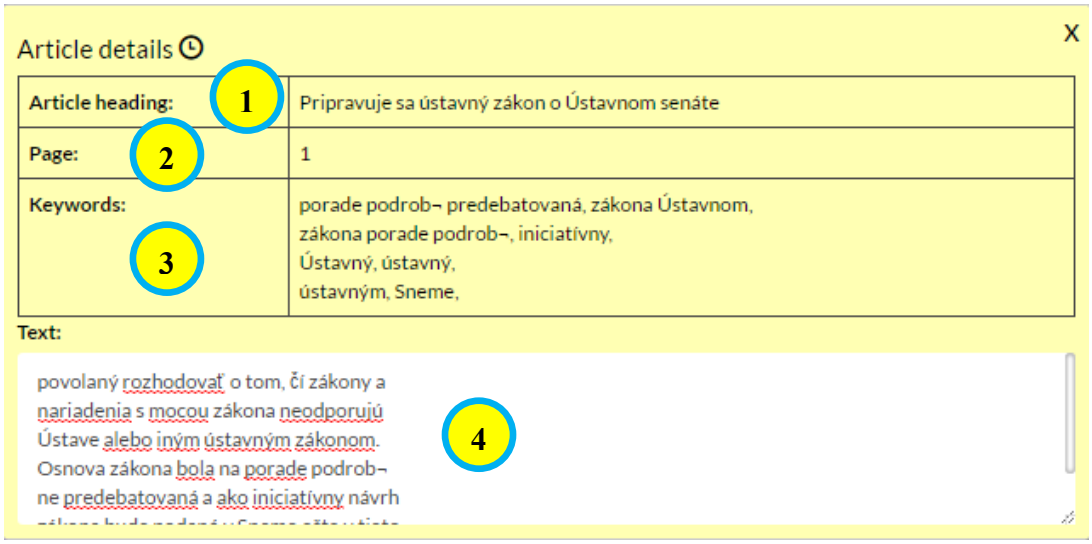


Editácia textu bloku

Pre editáciu textu bloku dvakrát kliknite na blok článku. Po týchto kliknutiach sa Vám zobrazí okno obsahujúce

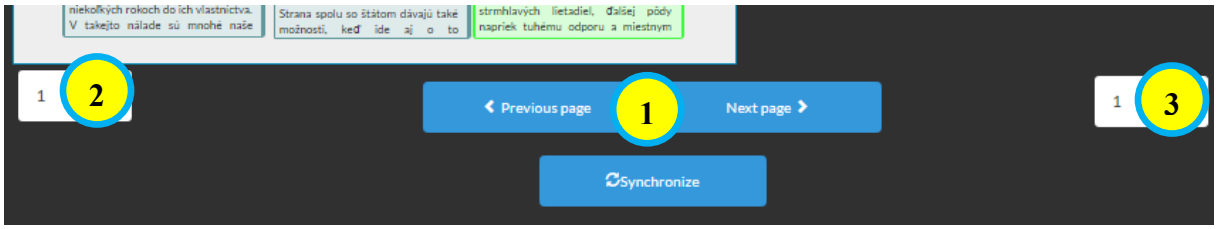
detaily článku, ku ktorému daný blok patrí. V riadku ***Article heading*** 1 je zobrazená skupina nadpisov patriacich k danému článku. V riadku ***Pages*** 2 sa nachádza číslo strany, na ktorej daný článok začína. V časti ***Keywords*** 3 sa nachádza desať kľúčových slov patriacich k danému článku.

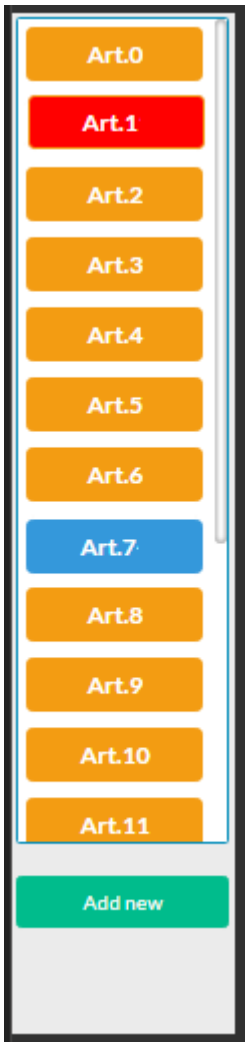
V textovom poli v spodnej časti okna 4 sa nachádza text nachádzajúci sa v danom bloku článku. Tento text je možné editovať. Zmeny vykonané v texte sa ukladajú automaticky. Zavretie okna kliknite na tlačidlo *X* v pravom hornom rohu.



Prepínanie strán

Na prepínanie strán sú určené polia a tlačidlá v spodnej časti aplikácie. Pre prepínanie oboch častí aplikácie, teda aj obrázku aj náhľadu do rozdelenia článkov, použite tlačidlá *Previous page* a *Next page* nachádzajúce sa v strede 1. Na prepínanie len ľavej (pravej) strany použite číselné pole nachádzajúce sa na ľavej (pravej) strane 2 3





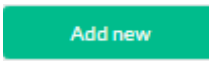
Hľadanie článkov a pridávanie článkov

Na ľavej strane aplikácie sa nachádza panel, v ktorom sa nachádzajú tlačidlá s číslami jednotlivých článkov. Tieto tlačidlá slúžia na prepínanie medzi článkami a ich rýchle vyhľadávanie. Pre rýchle prepnutie na hľadaný článok, kliknite na tlačidlo s jeho číslom. Do časti s grafickým zobrazením sa načíta strana, na ktorej článok začína. Rovnako sa zmení aj náhľad do skutočnej podoby časopisu.

V prípade, že sa momentálne v danom článku nenachádzajú žiadne bloky, zafarbí sa toto tlačidlo na červeno.

Po stlačení tlačidla pre uloženie zmien budú všetky články, ktoré neobsahujú žiadne bloky vymazané.

Pre pridanie nového článku použite tlačidlo:



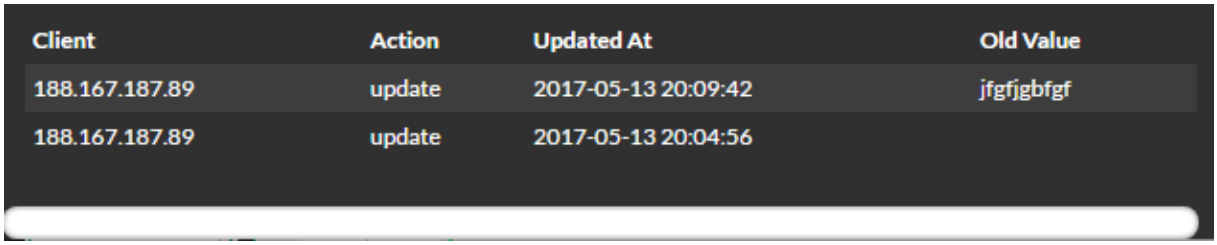
Po kliknutí na toto tlačidlo sa na konci zoznamu čísel článkov zobrazní nové tlačidlo zafarbené domodra.

K tomtu číslu môžete priradiť Vami vybrané články.

Kontrola zmien

Pre kontrolu zmien ktoré vykonali jednotlivý používatelia v systéme kliknite sa symbol:

Tento symbol sa nachádza napríklad pri názvoch polí v informáciách o danom čísle alebo v hlavičke okna, v ktorom sú zobrazené informácie o článku. Po kliknutí na tento symbol sa zobrazí tabuľka. V tejto tabuľke sa nachádzajú záznamy o zmenách, ktoré boli v daných poliach vykonané, pričom v stĺpci ***Client*** sa nachádza IP adresa používateľa, v stĺpci ***Action*** sa nachádza typ činnosti, akú vykonal, v stĺpco ***Updated At*** sa nachádza dátum a čas vykonania zmeny a v stĺpci ***Old Value*** sa nachádza hodnota, ktorá sa tam nachádzala pred zmenou.



Client	Action	Updated At	Old Value
188.167.187.89	update	2017-05-13 20:09:42	jfgfjgbfgf
188.167.187.89	update	2017-05-13 20:04:56	

Technická dokumentácia

V tomto dokumente sa nachádza hlavná technická dokumentácia pre 3 hlavné časti nášho projektu a to:

1. Python backend
2. Databáza ElasticSearch
3. Ruby on Rails frontend

Sú tu zhrnuté len základné informácie o fungovaní, zvyšok sa dá vyčítať z kódu, ktorý je z väčšej časti vhodne okomentovaný, ako aj písaní podľa metodík, takže by mal byť rozumne čitateľný aj pre človeka, ktorý s projektom ešte nepracoval.
Ďalšie infomácie sa dajú nájsť aj na README stránkach oboch repozitárov ako aj v inštalačnej príručke.

Python Backend

Organizácia repozitára

Nachádza sa tu 5 zložiek:

- tests
- parser
- helper
- misc
- other

Funkcionalita hlavného hrubého parsera sa nachádza v zložke parser.

Dodatočná funkcionalita pre triedu Semantic sa nachádza v zložke helper.

V zložke misc sú uložené requirements.txt, v ktorých sú uložené všetky knižnice, ktoré sme používali.

V zložke tests sú uložené všetky testy, ktoré replikujú hierarchiu zložiek hlavného repozitára.

Other slúži ako vedľajši priečinok od projektu, sú tu uložené vedľajšie kódy, ktoré sa nepoužili v programe, boli pri trénovacích user story a podobne.

Trieda Semantic

Trieda Sematic je hlavná trieda. Dedí tieto triedy:

- Elastic
- Analyzer
- Marc
- ImageExtractor

Obsahuje funkcie: __init__(self, **args), print_articles(self), print_correct_articles(self)

Funkcionalita: Používa sa na volanie funkcií ostatných tried. Spúšťa parsovanie xml.
Parsovanie xml postupne menia tieto triedy:

1. Cleaner - vyčistenie xml
2. SourceHeader - extrakcia hlavičky z dokumentu
3. Heading - určenie nadpisov, ostatné sú fulltexty
4. SeparatorId - určenie separátorov
5. Preprocessor - spájanie blokov do skupín
6. Assembler - spájanie skupín do článkov

Počas tohto procesu sa pracuje s objektom z knižnice lxml, ktorý reprezentuje parsované xml.

Modul parser

Modul parser poskytuje základnú funkcionalitu pre Semantic. Pre prácu s xml súborom používa knižnicu lxml. Skladá z týchto častí:

- Cleaner
- SourceHeader
- Discriminator
    - Heading
    - Fulltext
    - SeparatorId
- Article
    - Preprocessor
    - Assembler
- XmlParser

Trieda Cleaner

Obsahuje funkcie: clean(cls, parsed_xml)

clean(cls, parsed_xml)
Čistí vstupné xml. Najprv odstráni prefixy. Potom odstráni nepotrebné atribúty pre elementy. Ostanú len tieto parametre pre elementy:

- Document: 'producer', 'version', 'pagesCount', 'mainLanguage', 'languages'
- Page: 'resolution', 'width', 'height'
- Block: 'blockType', 'l', 't', 'r', 'b'
- Line: 'baseline', 'l', 't', 'r', 'b'
- Formatting: 'lang', 'ff', 'fs', 'spacing'

Ďalším korkom je odstránenie elementov charParams. Znaky, teda nie sú v samostatných elementoch, ale pre každý element Formatting je spojený do celého textu. Tiež mení štruktúru xml z block -> text -> par -> line na block -> par -> line.

Triede SourceHeader

Obsahuje funkcie: get_source_header(cls, parsed_xml, header)

get_source_header(cls, parsed_xml, header)
Vyberá základné informácie o dokumente podľa konfiguračného súboru. Vyberajú sa informácie ako číslo časopisu, miesto vydania.

Modul Discriminator

Discriminator je modul pre určenie nadpisov, fulltextu a separátorov. Tieto elementy sú potrebné pre určovanie článkov. Nadpisy a fulltexty sa určujú podľa veľkosti textu. Ako hranica pre určenie

nadpisu sa používa najčastejšie používaná veľkosť na strane. Všetky väčšie texty sa považujú za nadpis a rovné a menšie za fulltext. Trieda Separator určuje separátory v texte.

Trieda Heading

Obsahuje funkcie: discriminate_headings(cls, parsed_xml)

discriminate_headings(cls, parsed_xml)
Určí nadpisy v zadanom xml.

Trieda Fulltext

Táto trieda sa už nepoužíva.
Obsahuje funkcie: discriminate_fulltexts(cls, parsed_xml)

discriminate_fulltexts(cls, parsed_xml)
Určí fulltexty.

Trieda SeparatorId

Obsahuje funkcie: discriminant_separators(cls, parsed_xml)

discriminant_separators(cls, parsed_xml)
Určí separátory v xml.

Trieda XmlParser

Trieda XmlParser len spája funkcionalitu všetkých predošlých modulov.

Modul Article

Modul Article na základe určených nadpisov, fulltextov a separátorov vytvára články.

Trieda Preprocessor

Obsahuje funkcie: preprocess(cls, parsed_xml)

preprocess(cls, parsed_xml)
Spája všetky za sebou idúce nadpisy, alebo fulltexty do väčších celkov(group).

Trieda Assembler

Obsahuje funkcie: assembly_articles(self)

assembly_articles(self)
Podľa zadefinovaných pravidiel(v dokumente analyza.pdf) spája jednotlivé články.

Modul helper

Modul helper je nami vytvorený Python modul, ktorý obsahuje viacero pomocných funkcionalít, ktoré sa pridávajú ku triede Semantic a obohacujú tak jej funkcionalitu. Skladá sa z týchto aktívnych častí:

- elastic_filler.py -> trieda Elastic
- image_extractor.py -> trieda ImageExtractor
- marc.py -> trieda Marc
- semantic_analyzer.py -> trieda Analyzer
- infohandler.py -> trieda InfoHandler

Okrem týchto sa tu nachádzajú staré, neudržované funkcie, ktoré sa nepoužívajú, ale slúžia ako referencie progresu a v prípade pivotu funkcionality sa môžu použiť.

Trieda Elastic

Trieda Elastic pridáva funkcionalitu komunikácie s našou databázou ElasticSearch.
Obsahuje funkcie: save_to_elastic

save_to_elastic(self, issue_name, dirname, paths)
Nadviaže komunikáciu s ElasticSearchom na základe vopred definovaných informácií, ktoré sú úložené v konfiguračnom súbore. Následne vyextrahuje informácie z názvu súbora, z argumentov, pomocou, ktorých sa funkcia volala a iteratívnym spôsobom (vnorené cykly po stranách, po článkoch, po groupách a tie sa následne spracúvajú) vytvorí objekty, podľa definície, naplní ich údajmi a takto spracované číslo (lxml element, ktorý vznikne po spracovaní čísla) vloží do databázy.
Zápisy sú vkladané do loggera, ktorý by mal mapovať aktivitu vkladania.

Trieda ImageExtractor

Trieda ImageExtractor pridáva funkcionalitu pridávania výrezov článkov z pôvodných obrázkov do nášho filesystému, pričom informácie o článkoch berie z databázy ElasticSearch, pričom sa predpokladá prvotné spustenie generovania Marc záznamov, ktoré ako prvé predpripravia filesystém na úroveň článkov.
Obsahuje funkcie: export_article_image, export_image_for_issue

export_image_for_issue(self, issue_id)
Nadviaže komunikáciu s elasticsearchom, s konkrétnym id časopisu-čísla-issue, ktorý mu je zadaný a vyextrahuje z neho objekty článkov. Tak isto nájde zdrojové obrázky v zložkách STR v základnej zložke čísla.
Tieto údaje následne pošle do druhej funkcie export_article_image

export_article_image(cls, article, pages_paths)
Podľa zadaného článku sa nájdu jeho groupy a ich súradnice a tie sa vyextrahujú zo zadaných strán a vložia sa na čierne pozadie.
Doextrahujú sa do zložiek, ktoré sa vytvoria pri vytváraní Marc súborov pri funkcionalite triedy Marc.

Trieda Marc

Trieda Marc pridáva funkcionalitu generovania xml MARC21 záznamov, pre číslo časopisu a všetky jeho články, ktoré sú načítané v databáze ElasticSearch.
Obsahuje funkcie: __export_issue, __export_article, __get_time, __get_008_issue, __heading_subheading, __save_marc, export_marc_for_issue

Funkcionalita: Zabezpecuje vytvorenie nových zložiek do filesystému, do zložky čísla, kde sa vytvorí zložka articles v ktorej budú číselne pomenované zložky ku každému článku v ElasticSearchi, a do nich sa vytvorí MARC21.xml záznam o každom článku. Tak isto sa vytvorí MARC21.xml záznam na úrovni čísla.

Trieda Analyzer

Trieda Analyzer pridáva funkcionalitu extrahovania kľúčových slov z článkov, ktoré sú uložené v databáze ElasticSearch.
Obsahuje funkcie: insert_key_words, key_words_from_json, key_words, __tfidf, __idf, __n_containing, __tf
Funkcionalita: Využíva štatistickú metódu TF-IDF na vyhľadanie a extrahovanie kľúčových slov z textov článkov, ktoré sú úložené v databáze ElasticSearch. Odstraňujeme stop slová, ktoré mám definované v liste stops na začiatku súbora semantic_analyzer.py.

Trieda InfoHandler

Trieda InfoHandle pridáva pomocné funkcie pre jednoduchšie iniciálizácie logovanie v iných funkciách.
Obsahuje funkcie: __init__, emit
Funkcionalita: Jednoduchá trieda pre vytváranie logov, na základe klasickej dokumentácie Python3 logging. Pomocná trieda.

Neudržované

Mysql filler, drop_tables, init_tables

Opis: Triedy a funkcie pre komunikáciu s databázou MySQL, boli použité pre uspokojenie požiadaviek zákazníka, kým sme nenastavili a neprešli na plné využívanie databázy ElasticSearch.

Graphic

Opis: Pomocná trieda pre vykreslovanie objektov, podobne ako v ImageExtractore. Stretli sme sa tu ale s problémami, takže táto trieda funguje iba ako referenčná, keďže vidíme čo a ako sme skúšali. Slúžila pre naplnenie požiadaviek zákazníka

Testy a Continuos Integration (CI)

Testy majú štruktúru, ktorá replikuje celkovú štruktúru súboru, len sú uložené v zložke tests. Išlo nám o zlepšenie prehľadnosti ciest k funkciám a ich testom.

Ručné spúšťatie testov funguje pomocou príkazu:

```
python3 -m pytest tests
```

Takéto spúšťanie ale nie je nevyhnutné, keďže máme nastavenú plnú synchronizáciu s travisom, ktorý testy spustí v 100% oddelenom prostredí.

CI je implementované pomocou nástroja TravisCI.
V hlavnej zložke je vytvorený súvor travis.yml, v ktorom sú nastavené základne príkazy, aby sa nám replikovalo produkčné prostredie repozitára, nainicializovala sa nám databáza ElasticSearch a vykonali sa všetky testy. Máme tu aj prepojenie so Slackom, našim hlavným komunikačným kanálom, do ktorého nám chodia správy o prejdení alebo spadnutí testov, podľa čoho vieme ako pokračovať.

ElasticSearch

No-SQL databáza, ktorú sme si vybrali pre jej rýchlosť pri práci s textami, čo zahŕňa všetky naši zdielané dáta.

Verzia: 5.2.2

Nastavenia: Žiadne špeciálne

Reset a inicializácia

Pred efektívnym používaním ElasticSearchu s naším back a frontendom treba spustiť skript /helper/reset_elastic_indices.py, ktorý definuje indexy a ostatné nastavenia nevyhnutné pre komunikáciu s aplikáciou a ktoré zabezpečujú konzistentné ukladanie dát v databáze
!! POZOR !! Spustenie tohto skriptu spôsobí resetovanie všetkých dát, ktoré sú aktuálne uložené v databáze. Tento skript sa spúšťa iba jediný krát v produkčnom prostredí a následne ostavá len pre testovacie prostredie alebo prípade veľkých výpadkov, kedy sú dáta aj tak stratené a chcem dostať databázu do čistého neutrálneho stavu.

Ruby on Rails frontend

Základné informácie sa dajú nájsť v repozitári a k návodu na inštaláciu.

JS Funkcie

function initArticles(images, issue, articles)

Funkcia pre načítanie článkov
Atribúty:

- images – pole prvkov typu String, obsahujúce zoznam linkov na obrázky strán čísla
- issue – JSON záznam obsahujúci informácie o čísle
- articles – pole JSON prvkov, z ktorých každý záznam predstavuje jeden článok

function getArticles()

funkcia volaná pri ukladaní čísla
Výstup:

- Pole článkov pre dané číslo

function ReloadPage()

Funckia pre načítanie článkov po zmene alebo prepnutí na ďalšiu stranu

function nextPage()
function previousPage()

Funkcie volané pri navigácii medzi stranami

function imagePageSwitch()

Funkcia volaná pri prepínaní strán v časti náhľadu do skenu strany

function synchronizePages()

Funkcia volaná pri synchronizácii čísel strán

function changeType(art_num, group_num, selectObj)

Funkcia volaná pri zmene typu bloku
Vstup:

- art_num – číslo článku
- group_num – číslo bloku
- selectObj – rodičovský element

function changeArticleNumber(art_num, group_num,selectObj)

Funkcia volaná pri čísla článku ku ktorému je blok priradený
Vstup:

- art_num – číslo článku
- group_num – číslo bloku
- selectObj – rodičovský element

function findArticle(art_num, selectObj)

Funkcia volaná pri použití tlačidiel pre hľadanie článku
Vstup:

- art_num – číslo článku
- selectObj – rodičovský element

function newArticle()

Funkcia pre Pridanie nového článku

function loadPopover(art_num, group_num, selectObj)

Funkcia pre načítanie popoveru pre daný blok
Vstup:

- art_num – číslo článku
- group_num – číslo bloku
- selectObj – rodičovský element

function changeText(art_num, group_num, selectObj)

Funkcia pre zmenu textu v danom bloku
Vstup:

- art_num – číslo článku
- group_num – číslo bloku
- selectObj – rodičovský element

function getRandColor()

funkcia pre vygenerovanie podfarbenia článku

Inštalačná príručka – Ruby on Rails

This README would normally document whatever steps are necessary to get the application up and running.

BASICALLY for production, follow steps on Deployment instructions, follow them in connection with information about used versions in this README.

* Ruby version (installed via rvm)
  * 2.3.0
* Rails version (installed via Gemfile - for newbies, just folov Deployment instructions)
  * 5.0.1

* System dependencies
  * Ubuntu 16.04 LTS
  * Python project
    * just follow istructions on python project "DeepSearch"
  * rvm (installed as user, NO root)
  * elasticsearch 5.3.0 (installed from elasticsearch web through dpkg as downloaded .deb file)

* Configuration
  * change all local paths in files below
    * parser.dir to dir location with stored raw issue files)
    * semantic.project_path to root dir of Python project
  * create file nautilus.rb in config/initializers, add these lines into that file
  ```
  # parser documents dir name
  Rails.application.config.x.parser.dir = '/home/vasko/Documents/skola/TP/parser_dir'
  Rails.application.config.x.semantic.project_path = '/home/vasko/PycharmProjects/TP-DeepSeach'
  Rails.application.config.x.semantic.filler_path = 'elastic_filler.py'
  Rails.application.config.x.parser.pictures_path = 'STR'
  Rails.application.config.x.semantic.updater_path = 'elastic_updater.py'
  Rails.application.config.x.semantic.marc_exporter_path = 'marc_exporter.py'
  ```
  * AGAIN, it is important to CHANGE all local paths in these files

* Database creation

* Database initialization (Note - service elasticsearch must be running)
  * execute script from python project TP-DeepSearch, with path helper/reset_elastic_indices.py (You need to check documentation for proper understanting)

* Services that needs to bu runned(job queues, cache servers, search engines, etc.)
  * restart service: "sudo service elasticsearch restart"
  * restart service: "sudo apache2ctl restart"

* Deployment instructions
  * We use Phusion Passenger application server, whole tutorial with all necessary steps is written at adress: https://www.phusionpassenger.com/library/walkthroughs/deploy/ruby/

* In Production
  * when project is updated (cd rails_roject_root)
    ```
    rake assets:precompile
    sudo apache2ctl restart
    ```
  * ADDITIONAL (elasticsearch database structure is modified)
    * execute Database initialization (ATTENTION, will erase all data from elasticsearch database)

* TESTS - just run "rake test:all" or run them with IDE, for example with RubyMine

Inštalačná príručka python

Virtualenv

Virtualenv preparation

$ virtualenv ENV
To activate newly created virtualenv, you can run

$ source ENV/bin/activate
Install all requirements

$ pip3 install -r misc/requirements.txt
Install Python3 external dependency

$ sudo apt-get install python3-tk
Install key words processing dependency

$ pip3 install -m textblob.download_corpora
More information about virtualenv can be found on documentation_.

REQUISITES

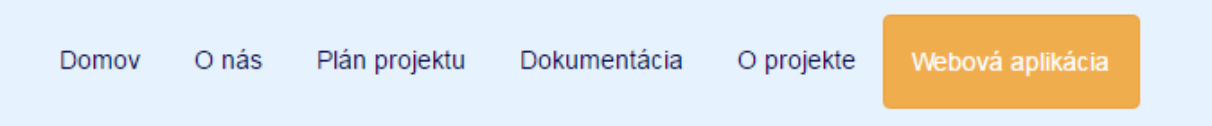
Database initialization * execute script from python project TP-DeepSearch
python3 helper/reset_elastic_indices.py

WARNING - WILL CLEAR ALL DATA OFF DATABASE
(You need to check documentation for proper understanting)

Webové sídlo tímu

Header

Súčasťou každej stránky webového sídla je navigačný header, ktorý obsahuje okrem loga tímu, ktoré je presmerovaním na úvodnú stránku, tiež odkazy na ostatné stránky a odkaz na webovú aplikáciu. Tento element zabezpečuje ľahkú navigáciu na všetkých stránkach ako aj informuje o všetkých častiach sídla, ktoré sú k dispozícii.

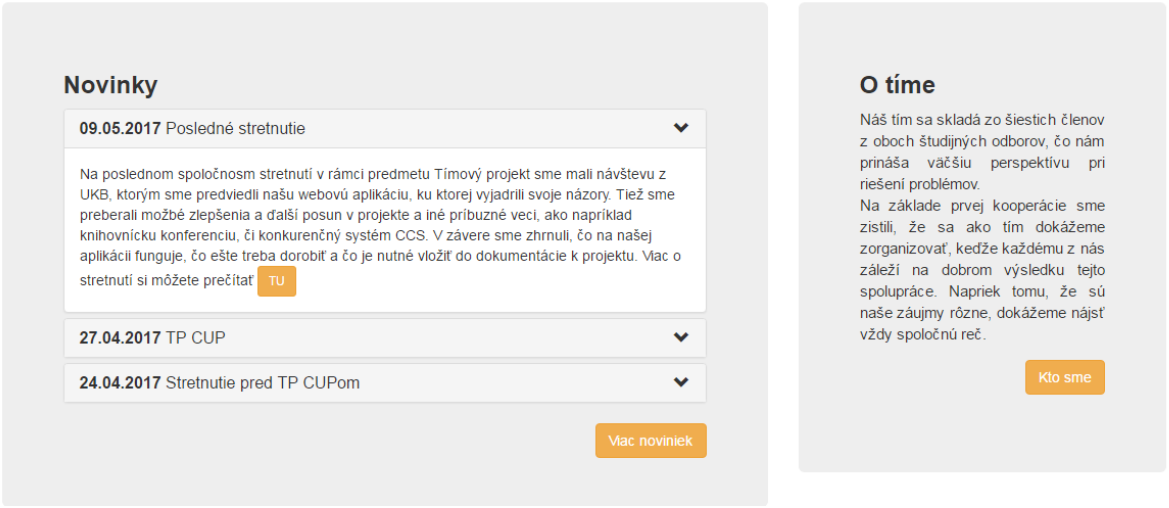


Úvodná stránka

Úvodná stránka je rozdelená na tri časti. V prvej časti je umiestnený prvok, v ktorom sa striedajú štyri obrazovky, odkazujúce na ostatné stránky sídla. Takto približujeme návštevníkovi stránky obsah ostatných stránok a zvyšujeme jeho záujem prejsť na ďalší obsah.

V strednej časti sa nachádzajú dve položky. Ľavá časť je naplnená aktuálnymi informáciami o práci v tíme, pokroku na projekte a iných udalostiach týkajúcich sa tímového projektu. Na zobrazenie je použitý otvárací element, ktorý umožňuje kompaktne zobraziť nadpisy udalostí a zároveň umožňuje pridať dlhší popis v prípade záujmu o danú tému. Dokumenty spomínané v texte sú prístupné cez tlačidlo v texte. V prípade väčšieho záujmu je v dolnej časti tlačidlo na zobrazenie ďalšieho obsahu.

Pravá položke je krátky text s informáciami o členoch nášho tímu spolu s prepojením na stránku s podrobnými informáciami o našich členoch.



V spodnej časti stránky sa nachádza footer, v ktorom sú jednoducho zobrazené kontaktné údaje v prípade záujmu návštevníka o projekt.

Dokumentácia

Stránka s dokumentáciou slúži na uchovávanie informácii o práci na projekte. Je rozdelená do dvoch častí, prvá je informačná, druhá slúži ako prístup k vypracovaným materiálom. V prvej časti s názvom Novinky o projekte sú uvedené všetky udalosti, ktoré boli zdokumentované a zverejnené na úvodnej stránke v časti Novinky. Texty jednotlivých noviniek sú označené nadpismi, ktoré informujú o dátume, kedy bola novinka pridaná a obsahu záznamu. Samotný text je bližšie popísaná udalosť, ktorá nastala pri práci na projekte. Odkazy na spomínané dokumenty sú priamo v texte ako hypertextový odkaz. Správy sú zhlukované do mesiacov, v ktorých vznikli pre lepšiu prehľadnosť.

Dokumenty

Zápis zo stretnutí

- Zápis zo stretnutia 27.09.2016
- Zápis zo stretnutia 06.10.2016
- Zápis zo stretnutia 13.10.2016
- Zápis zo stretnutia 20.10.2016
- Zápis zo stretnutia 27.10.2016
- Zápis zo stretnutia 03.11.2016
- Zápis zo stretnutia 10.11.2016
- Zápis zo stretnutia 15.11.2016
- Zápis zo stretnutia 25.11.2016
- Zápis zo stretnutia 01.12.2016
- Zápis zo stretnutia 08.12.2016
- Zápis zo stretnutia 15.12.2016
- Zápis zo stretnutia 27.02.2017
- Zápis zo stretnutia 06.03.2017
- Zápis zo stretnutia 13.03.2017
- Zápis zo stretnutia 20.03.2017

Novinky o projekte

Máj

09.05.2017 Posledné stretnutie

Na poslednom spoločnosm stretnutí v rámci predmetu Tímový projekt sme mali návštevu z UKB, ktorým sme predviedli našu webovú aplikáciu, ku ktorej vyjadrili svoje názory. Tiež sme preberali možbé zlepšenia a ďalší posun v projekte a iné príbuzné veci, ako napríklad knihovnícku konferenciu, či konkurenčný systém CCS. V závere sme zhrnuli, čo na našej aplikácii funguje, čo ešte treba dorobiť a čo je nutné vložiť do dokumentácie k projektu. Zápis o stretnutí je dostupný v časti Zápis zo stretnutí.

Apríl

27.04.2017 TP CUP

Dnes sme sa zúčastnili konferencie IIT.SRC, ktorou súčasťou bol aj TP CUP. Našu prácu sme preznentovali mnohým, vrátane hlavného hosťa Cesareho Putassa. Pri našej prezentácii sme použili poster, prezentáciu a letáčiky, všetky dostupné v časti Ostatné Aby to však nebolo všetko, podarilo sa nám postúpiť do semifinále súťaže, ktoré sa koná 7.6

24.04.2017 Stretnutie pred TP CUPom

Na dnešnom stretnutí sme sa snažili skompletizovať prípravu na TP CUP, dokončiť návrh posteru a letáčikov, a tiež prebrať obsah prezentácie, ktorá má byť umiestená na monitore. Na stretnutí sa zúčastili aj zástupkyne knižnice, ktorým sme predviedli náš produkt, webovú aplikáciu. Zápis o stretnutí je dostupný v časti Zápis zo stretnutí.

Druhá časť, obsahujúca linky na dokumenty je rozdelená na časť so zápismi o stretnutí a ostatnými dokumentmi vypracovanými počas práce na projekte. Dokumenty sú označené jasným popisným názvom v jednoduchom zozname, ktorý tiež umožňuje prístup k dokumentu kliknutím na názov alebo ikonu v pravej časti.

O projekte

Informačná stránka o projekte ponúka vizuálny prehľad práce na projekte. Stránka obsahuje motiváciu na vypracovanie projektu spolu s krokmi podstúpenými pri práci. Každý krok je detailne opísaný a názorne demonštrovaný vizuálne. Takýto prístup sme zvolili pre uvedenie návštevníka stránky do témy tak, aby pochopil ciele projektu a postupy, ktoré boli využité na dosiahnutie. Obrazová ukážka je použitá pre lepšiu predstavu návštevníka o procesoch vypracovania projektu , ktoré sa snažia vysvetliť tému aj osobe nezainteresovanej v danej téme.

Plán projektu

Stránka s plánom projektu bola vytvorená na začiatku projektu s položkami, ktoré sme predpokladali, že budú naplňou práce projektu. Plán projektu slúži ako možnosť porovnania vypracovaných úloh s predpokladaným scenárom šprintov a tiež ako možný spôsob ako pokračovať v práci na projekte v prípade, že na ňom bude chcieť pracovať nový tím. Za zobrazenie plánu sme zvolili opäť raz kompaktné elementy s jednoduchými nadpismi a rozbaliteľným obsahom pre skrátenie obsahu viditeľného na stránke.