Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004
DOI: https://doi.org/10.24867/13BE26Besic

MIKROSERVIS ZA EKSTRAKCIJU TEKSTA IZ WORD I PDF DOKUMENATA

MICROSERVICE FOR TEXT EXTRACTION FROM WORD AND PDF DOCUMENTS

Dejan Bešić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *U ovom radu će biti opisano rešenje ekstakcije teksta iz dokumenata u Word i PDF formatu. Pored same implementacije rešenja, diskutovaće se biblioteke koje su potrebne za ekstrakciju teksta, kao i za konverziju jednog formata dokumenta u drugi. Opisaće se struktura PDF dokumenta, zbog čega su u upotrebi kao i koji su problemi prilikom ekstraktovanja teksta. Problem ekstraktovanja teksta iz Word i PDF dokumenata se svodi na problem ekstraktovanja teksta iz PDF dokumenata.*

Ključne reči: *Mikroservis, ekstrakcija, tekst, Word, PDF, konverzija*

Abstract – *This paper will describe the solution of text extraction from documents in Word and PDF format. The solution will be created as a microservice architecture. In addition to the implementation of the solution itself, the libraries required for extraction as well as conversion will be discussed. This paper will describe the structure of PDF documents, why they are in us and what are possible disadvantages of text extraction from this type of documents. The problem of extracting text from PDF and Word documents, will be reduced to the problem of extracting text from PDF documents.*

Keywords: *Microservice, extraction, text, Word, PDF, conversion*

1. UVOD

U ovom radu se prolazi kroz Servisno orijentisanu arhitekturu kao i Mikroservisnu arhitekturu. Zatim, nepohodno je objasniti šta su PDF dokumenti, kakva je struktura ovakvih dokumenata i kakvi problemi mogu da nastanu prilikom ekstrakcije teksta baš zbog takve strukture. Opisano je kako se problem ekstrakcije teksta iz Word i PDF dokumenata se prebacuje na problem ekstrakcije teksta iz samo PDF dokumenata. Prolazi se kroz biblioteke za konverziju datoteka i zbog čega neke od tih biblioteka ne zadovoljavaju potrebe ovog rada. Zatim, prolazi se kroz biblioteke za ekstrakciju teksta, za čime sledi implementacija rešenja. Na kraju rada se nalazi zaključak u kojem se spominju nedostatci ovog rešenja i moguća unapređenja.

2. MIKROSERVISI

Postoje mnogi dizajn šabloni za implementaciju i dizajniranje programskog rešenja, zavisno od toga koje poslovne zahteve treba programsko rešenje da pokrije i koliko se planira unapred i ostavlja mogućnost za dalja unapređenja.
Ovo je neizbežno pitanje pri razvoju programskog rešenja, koliko god malo ili veliko po pitanju domena i opsega.

2.1. Servisno orijentisana arhitektura (SOA)

Osnovna jedinica SOA sistema jeste servis [2], dakle SOA se može definisati kao skup servisa, najčešće u distribuiranom računarskom sistemu, povezanih servisnim interfejsima koji komuniciraju pomoću zajedničkog komunikacionog kanala po predefinisanim pravilima. Ako odemo korak dalje, servisno orijentisana arhitektura predstavlja oblik organizacije integrisanog informacionog okruženja jednog sistema. Njega karakteriše ponuda i korišćenje njegovih distribuiranih poslovnih funkcija, grupisanih u različitim vidovima servisa. [1] Svaki servis u ovakvom sistemu sadrži u sebi kod (implementaciju, nezavisnu od konkretnog programskog jezika) i integraciju podataka neophodnih da se izvrši jedna kompletna, diskretna poslovna funkcija. Npr: kalkulacija mesečne rate za kredit, ili provera kreditnog statusa klijenta.

Dalje apstraktnu definiciju ovakve arhitekture čini skup principa pomoću kojih se definiše novi koncept informacionih sistema, od kojih su najznačajniji:

1. granulacija poslovnih funkcija,
2. obezbeđenje jedinstvenog pogleda na poslovanje kroz izgradnju arhitekture poslovnog sistema,
3. nezavisnost od tehnologije implementacije (fleksibilnost).

2.2. Mikroservisna arhitektura

Mikroservisna arhitektura je arhitektonski šablon koji je takoreći „nikao“ iz sveta domenski-vođenog dizajna (domain-driven design), kontinualnog isporučivanja softverskog rešenja, automatizacije kako platformi na koji je hostovan, tako i infrastrukture korišćena za hostovanje, i skalabilnih sistema. Mikroservisna arhitektura je zasnovana na SRP principu tj na principu da komponenta (u ovom slučaju servis) treba da ima jednu odgovornost

Ključna jedinica mikroservisne arhitekture leži u njenom nazivu: niz malih servisa, zvanih mikroservisa, tj. slobodnih programskih komponenti. Ovi servisi pružaju uslugu drugim servisima, sa kojima su povezani i sarađuju. Cilj jeste da servisi sadrže isključivo one funkcije koje su neophodne za izvršavanje jedne odgovornosti, i da se servisi kao takvi mogu nezavisno testirati i kombinovati.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branko Milosavljević, red. prof.

Ovakva struktura servisa omogućava brži razvoj, i lakše delegiranje resursa tima ka održavanju i puštanju novijih verzija servisa u produkciju [3].

3. PDF DOKUMENTI

Kao najčešći tipovi dokumenata u sadašnjoj upotrebi računara, nije na odmetu reći da su PDF i Word dokumenta prva asocijacija kada korisnik treba da podeli neke informacije ili prebaci ručno napisane formulate u digitalan ekvivalent. U ovom poglavlju se zalazi u strukturu PDF dokumenta, pošto su oba zasnovana ispod haube na XML notaciji.

3.1. Struktura

PDF je format dokumenta koji omogućava prezentovanje dokumenata sa tekstualnim sadržajem, multimedijalnim elementima, slikama i slično, razvije od strane Adobe korporacije, 1993. godine. Ovaj format je napravljen da bude agnostičan u odnosu na konkretne operativne sisteme, aplikacije i hardver. PDF fajl sadrži opis konkretnog rasporeda sadržaja dokumenta, kao i opis svakog elementa neophodnog za njegovo prikazivanje, bilo to fontovi, vektorske slike ili čist tekst.

4. PROBLEM EKSTRAKCIJE TEKSTA IZ PDF DOKUMENATA

Kada se posmatra ekstrakcija teksta iz dokumenta u PDF formatu, često se postavlja pitanje šta je tu zaista problem jer je tekst lako vidljiv ljudskom oku. Rad sa dokumentima u PDF formatu je težak zbog same ekstremne fleksibilnosti PDF-a. Najveći problem je što PDF format nikada nije stvarno dizajniran kao format unosa podataka, već je dizajniran kao izlazni format koji daje fino zrnastu kontrolu nad rezultirajućim dokumentom. U osnovi, PDF format se sastoji od niza instrukcija koje opisuju kako se crta na stranici, što znači da se tekstualni podaci ne čuvaju kao pasusi ili reči, već kao posebni karakteri sa instrukcijama gde će biti nacrtani na stranici. Kao rezultat, većina semantike sadržaja se gubi kada je tekst konvertovan u dokument u PDF formatu.
Mogući uzroci problema u ekstrakciji teksta iz dokumenata u PDF formatu mogu biti:

1. Zaštita od čitanja
2. Karakteri van stranice
3. Mali i nevidljivi karakteri na stranici
4. Veliki broj razmaka
5. Mali broj razmaka
6. Detekcija pasusa i reči
7. Redosled teksta i pasusa
8. Slike

5. BIBLIOTEKE ZA KONVERZIJU DOKUMENATA U WORD FORMATU U DOKUMENTE U PDF FORMATU

Nekada PDF dokument sadrži dodatne razmake između karaktera. To se uglavnom dešava zbog Kerning-a. Kerning je proces prilagođavanja razmaka između karaktera. Jedan od načina za rekonstruisanje ovakvog teksta je tehnikom optičkog prepoznavanja karaktera (OCR).

5.1 Documents4j

Documents4j je Java biblioteka za konverziju dokumenata u drugi format dokumenta. To se postiže davanjem ovlašćenja konverzije bilo kojoj izvornoj aplikaciji koja razume konverziju date datoteke u željeni ciljni format. Dokument4j dolazi sa adaptacijama za MS Word i MS Excel za Windows operativni sistem, što omogućava, na primer, konverziju dokument sa .docx ekstenzijom u PDF dokument bez uobičajenih izobličenja u rezultujućem dokumentu koja se često primećuju kod konverzija izvršenih pomoću proizvoda koji nisu Microsoftovi. Drugim rečima, ova biblioteka je zavisna od softvera MS Word i MS Excel. U slučaju rešenja opisanog u ovom radu, postoji nekoliko mana ove biblioteke koje sprečavaju njeno korišćenje:

1. Da bi funkcionisala, potrebno je posedovati MS Word i MS Excel u okruženju
2. Mikroservis u ovom radu će biti pokrenut na Linux operativnom sistemu, samim tim dodavanje MS Word i MS Excel softvera, nije tako jednostavno
3. MS Word i MS Excel su licencirani proizvodi, nisu besplatni
4. Ova biblioteka ne podržava konverziju dokumenata sa .doc ekstenzijom (obe ekstenzije, .doc i .docx, pripadaju Word dokumentima, gde .doc predstavlja stariji format)

5.2 Apache POI

Apache POI je projekat vođen od strane neprofitabilne kompanije pod nazivom ,,Apache Software Foundation''. Apache POI pruža biblioteke u Java programskom jeziku, za čitanje i pisanje datoteka u Mikrosoft Office formatima, kao što su Word, PowerPoint i Excel [4]. Poredeći sa Documents4j bibliotekom, Apache POI podržava konverziju za dokumente sa .doc i .docx ekstenzijom. Nije potreban prateći softver iz grupe Microsoft Office softvera. Takođe, ova biblioteka je nezavisna od platforme na kojoj se nalazi.

Međutim, ova biblioteka takođe poseduje mane zbog kojih u rešenju ovog rada, nije najprikladnija za korišćenje. Prva mana je što implementacija rešenja nije ista za dokumente sa .doc i .docx ekstenzijama. Implementacija za .docx je kratka i bez većih komplikacija. Problem nastaje prilikom implementacije rešenja za dokumente sa .doc ekstenzijom. Konvertovanje dokumenata sa .doc ekstenzijom se vrši ručno, tako što se ekstraktuje tekst, stilovi, razmaci, margine, fontovi, itd. i zatim se pomoću ekstraktovanih osobina dokumenta, kreira PDF dokument. Implementacija ovakvog rešenja već postoji međutim ni jedno od rešenja ne daje dovoljno dobre rezultate i vrlo često se dobije neodgovarajući PDF dokument. Za rešenje u ovom radu, veoma je bitno da PDF dokument odgovara Word dokumentu, i iz tog razloga ova biblioteka nije dovoljno precizna.

5.3 LibreOffice

LibreOffice je nastao na bazi popularnog slobodnog kancelarijskog paketa OpenOffice.org iz želje većine učesnika da se dalji razvoj izmesti u okrilje nezavisne neprofitne fondacije. LibreOffice je besplatan softverski

paket za Windows, Mac OS i Linux sisteme koji radi na uobičajenim računarima. Ova biblioteka, kao i Documents4j, zavisi od eksternog LibreOffice softvera koji mora da se nalazi u okruženju. Prednosti LibreOffice softvera u odnosu na Microsoft Office grupu softvera su:

1. LibreOffice je besplatan
2. Podržan je na Linux operativnom sistemu
3. Podržava konverziju dokumenata sa .doc i .docx ekstenzijama

LibreOffice biblioteka, iako se pokazala kao najbolja biblioteka za potrebe rešenja u ovom radu, postoji jedna mana koja je zajednička kod svih softvera. Implementacija se menja sa verzijama biblioteke i vrlo je moguće da će u budućnosti biti potrebe za izmenama u implementaciji ukoliko dođe do promene verzije biblioteke.

5.4 Gotenberg

Gotenberg je API bez stanja za konverziju HTML, Markdown i Office dokumenata u dokumente u PDF formatu [5]. Gotenberg podržava konverziju .doc i .docx dokumenata koristeći LibreOffice biblioteku. Gotenberg predstavlja rešenje problema sa verzijama u LibreOffice biblioteci, jer na ovaj način, taj problem se problem verzija prebacuje na Gotenberg. Gotenberg je, takođe, podržan na bilo kojoj platformi. Implementacija konverzije Word dokumenata u PDF dokumente se prebacuje na ovu biblioteku.

6. BIBLIOTEKE ZA EKSTRAKCIJU TEKSTA IY DOKUMENATA U WORD FORMATU

U prethodnoj sekciji je opisan problem konverzije iz dokumenata u Word formatu u dokumente u PDF formatu, dok u ovoj sekciji će se opisati biblioteke za ekstrakciju teksta kao i zašto je potrebna konverzija dokumenata.

6.1 Apache POI

U prethodnoj sekciji su opisane mane Apache POI biblioteke zbog čega se ne koristi u ovom radu za konverziju iz dokumenata u Word formatu u dokumente u PDF formatu. Konverzija dokumenata nije jedino svojstvo Apache POI biblioteke. Ova biblioteka takođe podržava ekstrakciju teksta iz dokumenata u Word formatu. Kao rezultat ekstrakcije se dobija celoukupni tekst iz datog dokumenta. Međutim, za potrebe rešenja u ovom radu, to nije dovoljno. Potrebe ovog rada je da se ekstraktuje tekst tačno na kojoj se stranici nalazi. Celoukupni tekst ne sadrži informaciju o stranicama. Parsiranjem teksta po količini novih redova je moguće, međutim nije u svakom slučaju tačno i dobijaju se nedovoljno precizni podaci.

6.2 Apache PDFBox

Apache PDFBox je Java PDF biblioteka otvorenog koda za rad sa PDF dokumentima. Ovaj projekat omogućava stvaranje novih PDF dokumenata, manipulaciju postojećim dokumentima i mogućnost izdvajanja sadržaja iz dokumenata. [6]

Neke od glavnih karakteristika ove biblioteke su:

- Ekstrakcija teksta iz PDF dokumenata
- Spajanje PDF dokumenata
- Enkripcija i dekripcija PDF dokumenata
- Lucene pretraživač
- Popujnavanje podataka obrasca FDF (Forms data format) i XFDF (XML Forms data format)
- Kreiranje PDF dokumenata od tekstualnih datoteka
- Kreiranje slika od stranica u PDF dokumentu
- Štampanje PDF dokumenata

Za razliku od biblioteka za ekstrakciju teksta iz Word dokumenata ova biblioteka podržava ekstrakciju teksta po stranici i iz tog razloga, Apache PDFBox će se koristiti u implementaciji rešenja problema ekstrakcije teksta.

7. IMPLEMENTACIJA REŠENJA

Implementacija mikroservisa urađena je kao Spring Boot aplikacija, u Java programskom jeziku. Portabilnost i nezavisnost servisa je rešeno kreiranjem mikroservisa kao Docker slike. Komunikacija sa ovim mikroservisom se vrši putem HTTP-a. Aplikacija sadrži jednu krajnju tačku preko koje je dostupna celom sistemu. Krajnja tačka se nalazi na relativnoj putanji pod nazivom /extract. Ova kranja tačka prima jedan parametar pod nazivom file. Parametar file je tipa MultipartFile što označava da ova metoda može da primi bilo koju datoteku. Zbog ovog parametra, metoda HTTP zahteva je POST. Ukoliko je file parametar izostavljen, rezultat ove metode će biti greška sa HTTP status kodom 400 i porukom koja govori da je file polje obavezno. Ovaj mikroservis treba da ekstraktuje tekst iz dokumenata u Word i PDF pa se iz tog razloga dokument filtrira koristeći ekstenziju. Ukoliko se naziv dokumenta završava sa doc ili docx, znači da je dokument u Word formatu. Ukoliko je naziv dokumenta završava sa pdf, znači da je dokument u PDF formatu. Ukoliko nije ni jedan ova dva formata, rezultat metode će biti greška sa HTTP status kodom 400 i porukom koja govori da dokument mora biti u Word ili PDF formatu. . U zavisnosti od formata, poziva se odgovarajući servis. Ukoliko je dokument u Word formatu, poziva se metoda extract iz WordExtractionService klase. Ukoliko je dokument u PDF formatu, poziva se metoda extract iz PdfExtractionService klase.

Rezultat, nakon uspešne ekstrakcije teksta, je objekat opisan klasom PDF. Atributi ove klase su:

- author – autor dokumenta
- title – naslov dokumenta
- subject – tema dokumenta
- keywords – ključne reči koje opisuju dokument
- numberOfPages – broj stranica
- creationDate – datum kreiranja kao tekstualni podatak u ISO 8601 formatu
- modificationDate – datum poslednje izmene kao tekstualni podatak u ISO 8601 formatu
- pages – lista stranica koje su opisane klasom Page

Pored atributa, ova klasa poseduje pomoćne metode radi lakšeg rukovanja atributima, kao što su:

- addPage – služi za dodavanje stranice

- setCreationDate – služi za postavljanje atributa creationDate u ISO 8601 format
- setModificationDate – služi za postavljanje atributa modificationDate u ISO 8601 formatu

WordExtractionService je klasa koja služi kao servis za ekstrakciju teksta iz dokumenata u Word formatu. Ova klasa sadrži jednu glavnu metodu, extract koja je vidljiva ostatku aplikacije. Extract metoda služi za ekstrakciju teksta iz dokumenata u Word formatu kao i dodavanje podataka o samom dokumentu. Ulazni parametar je dokument, dok je rezultat kreiran PDF objekat. Da bi se tekst ekstraktovao, potrebno je konvertovati dokument iz Word formata u dokument u PDF formatu. Konvezija se vrši pomoću metode wordToPDF iz ConvertService klase. Posle konverzije, sledi ekstrakcija teksta i novonastalog PDF dokumenta. Ekstrakcija teksta se vrši pomoću metode addText iz PdfExtractionService klase. Na kraju, preostaje da se dodaju podaci o dokumentu.

ConvertService klasa služi za konverziju dokumenata u Word formatu u dokumente u PDF format. Ova klasa poseduje jednu glavnu metodu, wordToPDF. Ova metoda koristi Gotenberg API koji će konvertovati Word dokument u PDF dokument. Gotenberg se pokreće u istom okruženju kao i mikroservis. Koristi se krajnja tačka na putanji /convert/office. Komunikacija između mikroservisa i Gotenberg API-a se odvija putem HTTP-a. Šalje se HTTP zahtev sa POST metodom. Bitno je naglasiti da ContentType u zaglavlju HTTP zahteva mora da bude multipart/form-data. Telo zahteva treba da sadrži dva polja, files i waitTimeout, gde je files niz dokumenata koji treba da se konvertuju a waitTimeout maksimalna dužina čekanja konverzije dokumenata. Rezultat ove konverzije jeste niz bajtova koji predstavljaju dokument u PDF formatu.

PdfExtractionService je klasa koja služi za ekstrakciju teksta i meta podataka iz dokumenata u PDF formatu.

Dodavanje teksta se vrši pomoću Apache PDFBox biblioteke, koja služi za izvlačenje teksta po stranici u dokumentu. Kao rezultat rada ove biblioteke, dobija se lista stranica u PDDocument formatu. Iz svake stranice je potrebno ekstraktovati tekst posebno. Takav ekstraktovani tekst je potrebno srediti tako što se uklanjaju suvišni razmaci između teksta. Svaku stranicu je potrebno numerisati i na kraju dodati u PDF objekat

Za kreiranje i pokretanje Docker slike su bitna dva fajla, Dockerfile i docker-compose.yml. Dockerfile se koristi za kreiranje okruženja koje je potrebno da bi mikroservis funkcionisao. Da bi ovaj mikroservis mogao da funkcioniše, potreban je Java Development Kit. Zatim, potrebno je prebaciti već u napred pripremljene .jar datoteke koje predstavljaju ovaj mikroservis. Docker-compose.yml služi da bi povezali Gotenberg API sa mikroservisom za ekstrakciju teksta. U ovom fajlu je bitno definisati na kojem portu će biti dostupan mikroservis i Gotenberg API. Takođe je bitno da se definiše da mikroservis za ekstrakciju teksta zavisi od Gotenberg API-a. Ukoliko pokretanje Gotenberg API ne uspe da se pokrene, ceo proces će biti zaustavljen. Za Gotenberg je bitno da se navede tačan naziv i verzija kao i port na kojem će da „sluša''.

8. ZAKLJUČAK

Ovaj rad objašnjava problem ekstrakcije teksta iz dokumenata u Word i PDF formatu. Diskutuje koja od biblioteka najbolje odgovara problemu konverzije dokumenata iz dokumenata u Word formatu u dokumente u PDF formatu. Opisana je struktura mikroservisa, razlika između SOA i mikroservisa kao i prednosti ovakve arhitekture sistema. Dat je pregled strukture PDF dokumenata, zašto predstavlja jedan od najzastupljenijih formata za pregled dokumenata. Objašnjeno je koji problemi postoje korišćenjem PDF dokumenata i kako se rešavaju ti problemi u praksi, iako mnogi od njih nemaju idealno rešenje. Takođe, objašnjeno je kako se problem Word i PDF dokumenata, svelo na problem ekstrakcije teksta samo iz PDF dokumenata.

Kao poboljšanje ovog rada može biti povećani broj tipova dokumenata koje ovaj mikroservis podržava. Kao jedno rešenje, može biti takođe Gotenberg API. Ovaj API podržava veliki broj dokumenata, pa kao rešenje može biti konverzija tih dokumenata u PDF format, pa zatim ekstraktovati tekst iz tih dokumenata. Takođe, kao jedan vid unapređenja može biti bolji način detekcije vrsta dokumenta. Većina dokumenata ima „potpis'' po kojem se prepoznaje da li dati dokument odgovara datom formatu. Na taj način se može smanjiti broj grešaka koji se mogu dogoditi unutar mikroservisa i učiniti ceo sistem robusnijim. Kao još jedan vid unapređenja bi moglo da bude ekstrakcija multimedijalnog sadržaja iz dokumenata.

Cilj ovog rada je bio da se kreira mikroservis koji će za svaku stranicu PDF ili Word dokumenta, posebno ekstraktovati tekst i takav ekstraktovani tekst numerisati stranicom na kojoj se nalazi.

9. LITERATURA

[1] Servisno orijentisana arhitektura i integrisanje poslovnih aplikacija. Preuzeto sa https://www2.masfak.ni.ac.rs/uploads/articles/www2_5._soa_skraceno.pdf

[2] Servisno-orijentisana arhitektura, IBM, https://www.ibm.com/cloud/learn/soa

[3] Microservice Architecture, https://microservices.io/patterns/microservices.html

[4] Apache POI, https://en.wikipedia.org/wiki/Apache_POI

[5] Gotenberg, https://thecodingmachine.github.io/gotenberg

[6] Apache PDFBox, https://en.wikipedia.org/wiki/Apache_PDFBox

Kratka biografija:

Dejan Bešić rođen je u Novom Sadu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika odbranio je 2021.god. kontakt: dejanbesic31@gmail.com