

Jedan pristup ekstrakciji, transformaciji i punjenju – Rešenje za povećanje dostupnosti poslovnih podataka KonvertIz

Đorđe Ilijevski, dr Dušan Vujošević

Sadržaj — U okviru ovog rada fokus je postavljen na povećavanje dostupnosti podataka iz poslovnih informacionih sistema. Analiziraju se tehnologije ekstrakcije, transformacije i punjenja. Prikazuje se razvoj konkretnog softverskog rešenja, KonvertIza, implementiranog korišćenjem biblioteka *Apache POI* i *iText*. Ovo rešenje na jednostavan način konvertuje podatke iz *Excel* fajla i pravi *PDF* dokument u koji će ga smestiti. Potreba za ovakvim i sličnim renderovanjem prikaza podataka proizilazi iz sve većeg značaja podataka, koji su u dinamici savremenog poslovanja jedan od najznačajnijih resursa postizanja konkurentске prednosti i poslovnog uspeha. Softveri za rad sa transakcionim podacima, sa mogućnostima prikupljanja, upravljanja, skladištenja i transformacije podataka, u velikoj meri doprinose ubrzanju i automatizaciji poslovnog procesa, ali i efikasnosti donošenja poslovnih odluka, zbog čega imaju sve veću primenu u poslovnom svetu. Predloženo rešenje KonvertIz prikazano je kroz objašnjenje logike programa, diskusiju programskog koda, primere interfejsa i renderovanih dokumenata. Studija slučaja prati izveštaj pod nazivom Dokument o obračunu carinskih dažbina po osnovu zabrane povraćaja ili oslobodenja, kao primer izveštaja koji su kreirani KovertIzom.

Ključne reči: poslovni softver, softver za povećanje dostupnosti podataka, konverzija podataka, automatizacija, KonvertIz

I. UVOD

Veliki doprinos postizanju standardizacije u okviru radnih procesa, kao i efikasnosti u donošenju poslovnih odluka, pružili su različiti oblici poslovnih softvera, čija je primarna funkcionalnost rad sa transakcionim

Đorđe Ilijevski, Računarski fakultet, Srbija (telefon: 381-63-8376700; e-mail: djilijevski2323m@raf.rs).

podacima iliti evidentiranje detalja poslovanja. Softver za upravljanje poslovnim podacima je skup alata i aplikacija koji omogućavaju organizacijama da efikasno prikupljaju, čuvaju, analiziraju i vizuelizuju podatke, kako bi imali relevantne informacije i dobre uvide.

Ovakvi softveri imaju veoma značajnu ulogu u automatizaciji poslovnih procesa, smanjenju grešaka u podacima, poboljšanju poslovne efikasnosti, digitalnoj transformaciji i podršci informisanom donošenju odluka. Upravo stoga, postoji potreba za njihovom dubljom analizom, razumevanjem mesta i uloge ovih softverskih rešenja, načina na koji se oni pojavljuju na tržištu softvera, ali i analizom njihove primenljivosti u poslovnom okruženju.

Iz tog razloga, u okviru ovog rada, biće sagledana tematika softvera za povećanje dostupnosti podataka. I to, najpre, kroz analizu primene informaciono-komunikacionih tehnologija u savremenom poslovnom okruženju. A, zatim, i kroz konkretne primene ovog softvera, kao i pregled najpopularnijih softvera za rad sa podacima, na osnovu kojih je moguće utvrditi koja je vrednost koju ova softverska rešenja pružaju korisnicima.

Pored toga, analizu specifičnosti i značaja softvera za povećanje dostupnosti podataka moguće je proširiti analizom načina na koji može doći do kreiranja, implementacije i primene softvera za rad sa podacima. U radu će se opisati praktična realizacija jednog softvera za učitavanje, konverziju i punjenje podacima, pod nazivom KonvertIz. Prikazaće se njegova praktična primena.

II. INFORMACIONO – KOMUNIKACIONE TEHNOLOGIJE I SOFTVER ZA RAD SA POSLOVNIM PODACIMA

Tehnološki razvoj, posebno u oblasti telekomunikacija i informacionih tehnologija, značajno je uticao na oblikovanje informatičkog društva. Razvoj interneta, mobilnih uređaja i digitalnih platformi omogućio je jednostavniju i bržu komunikaciju, pristup informacijama i povezivanje širom sveta. [1]

Softver za rad sa poslovnim podacima omogućava organizacijama prikupljanje, obradu, analizu i vizuelizaciju podataka, kako bi se poboljšale poslovne operacije i donela informisana poslovna odluka. Čine ga alati koji pomažu u upravljanju podacima iz različitih izvora. Alati kao što su *Talend*, *Tableau*, *Qlik Sense* i *Looker* omogućavaju automatizaciju i efikasnost u procesu integracije i analize podataka.

A. Namena i značaj softvera za rad sa poslovnim podacima

Softveri za obradu podataka zauzimaju značajno mesto u savremenom poslovanju, pružajući organizacijama sredstva za efikasno upravljanje, analizu i tumačenje velikih količina podataka. Sam softver za rad sa

podacima, moguće je definisati kao integrisane platforme i alate koji su dizajnirani da olakšaju:

- prikupljanje,
- upravljanje,
- skladištenje i
- transformaciju podataka. [2]

B. Softver za konverziju podataka

Softveri za konverziju podataka, kao značajan segment šire kategorije softvera za povećanje dostupnosti podataka, predstavljaju ključnu komponentu u procesu transformacije podataka iz jednog formata u drugi. Ovaj proces je od suštinskog značaja kako bi podaci postali kompatibilni sa ciljanim sistemom, aplikacijom ili metodom skladištenja. Često se sprovodi kao deo većih projekata, poput migracije ili integracije podataka.

Naime, u domenu analize ovakvih tipova softvera, veoma je značajno sagledati elemente njihove primarne namene, koja se prevashodno odnosi na standardizaciju, različite operacije konverzije, ali i mogućnosti automatizacije radnih procesa. Prema tome, moguće je istaći da konverzija podataka ima značajnu ulogu u standardizaciji informacija, omogućavajući preduzećima da stvore „jedinствeni izvor istine“. Ova homogenizacija podataka postaje esencijalna tokom procesa promena u preduzeću, kao što su spajanja, akvizicije ili zajednički poduhvati, gde se integrišu ljudi, procesi i podaci iz različitih organizacija. [3]

C. Popularni softveri za rad sa podacima i elementi njihove primene

Neka poznata rešenja ističu se sveobuhvatnim funkcionalnostima koje služe da organizacijama i kompanijama omoguće izvlačenje maksimalne vrednosti iz njihovih podataka. Rešenja koja omogućavaju prikupljanje, upravljanje, skladištenje i transformaciju podataka, razvijala su se na različite načine, tako da danas savremena rešenja zadovoljavaju širok spektar potreba savremenih poslovnih sistema. U skladu sa tim, moguće je izdvojiti skup najznačajnijih softvera za rad sa podacima, koji nalaze veliku primenu u savremenim poslovnim sistemima. Među njima su:

- *IBM InfoSphere DataStage*
- *Talend*
- *Informatica PowerCenter*
- *Microsoft Power BI*
- *Salesforce Tableau*
- *Qlik Sense*

- *Looker*
- *Klipfolio*
- *Zoho Analytics*
- *Domo*, ali i drugi.

Ovi popularni softveri za povećanje dostupnosti podataka vremenom su postali stub modernog poslovanja, pružajući organizacijama alate za efikasno upravljanje podacima. Od integracije organizacionih podataka, do vizualizacije i dubinskih analiza, ovi alati doprinose postizanju ciljeva organizacija. Analizom njihovih funkcionalnosti i karakteristika, moguće je sagledati, kako izazove u domenu razvoja ovakvih softvera, tako i zahteve koje je potrebno postaviti, kako bi novi softver za povećanje dostupnosti podataka pružio adekvatnu vrednost korisnicima, ali i tržišnu prednost proizvođaču u odnosu na postojeće alternative. [4]

III. RAZVOJ SOFTVERA ZA UČITAVANJE, KONVERZIJU I PUNJENJE PODACIMA

Softver *KonvertIz*, koji je izrađen i predstavljen u nastavku, samo je jedan primer na osnovu koga mogu da se iz *Excel* dokumenata pročitaju podaci i da se od njih napravi *PDF* dokument. Kod je napisan u programskom jeziku *Java*. Dve biblioteke koje su korišćene su *Apache POI* za *Excel* fajlove i *ItxtPDF* za *PDF* fajlove. Biblioteke mogu da se implementiraju na sledeći način:

```
import org.apache.poi.ss.usermodel.*;  
import com.itextpdf.text.*;
```

Apache POI je projekat otvorenog koda koji pruža biblioteku za rad sa *Microsoft Office* formatima dokumenata, kao što su *Word*, *Excel* i *PowerPoint*, koristeći programski jezik *Java*. Naziv "*POI*" potiče od "*Poor Obfuscation Implementation*". U paketu *org.apache.poi.ss.usermodel* u *Javi* nalazi se deo *Apache POI* projekta, koji je biblioteka za rad sa *Microsoft Office* dokumentima (kao što je *Excel*) u *Javi*. Ovaj paket omogućava programerima da manipulišu *Excel* radnim listovima (*spreadsheet*-ovima) iz svojih *Java* aplikacija. Neki od osnovnih zadataka koji mogu da se obavljaju sa ovim paketom uključuju čitanje, pisanje i izmenu *Excel* datoteka, kao i rad sa ćelijama, redovima i kolonama unutar radnih listova. [5]

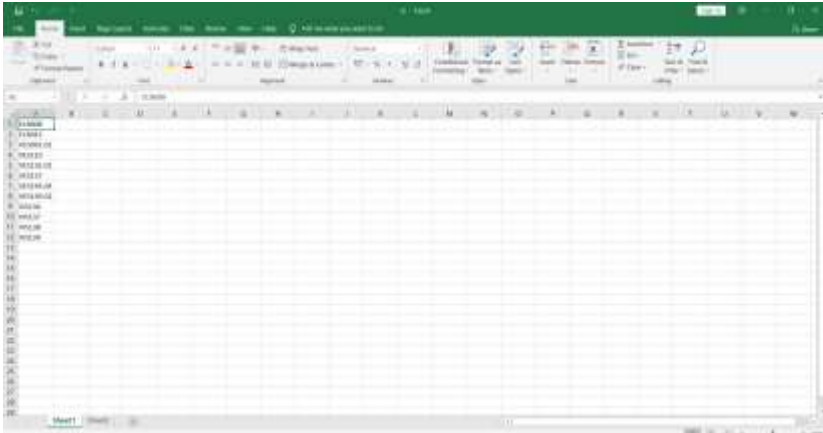
Biblioteka *iText* sadrži funkcionalnosti koje omogućavaju manipulaciju *PDF*-om. Omogućava kreiranje novih *PDF* dokumenata, kao i uređivanje postojećih, poput dodavanja teksta, slika, tabela, linkova, formi i drugih

elemenata. Takođe, ova biblioteka podržava i složenije funkcionalnosti poput digitalnog potpisivanja dokumenata. Korišćenje paketa *com.itextpdf.text* u *Java* projektima omogućava programerima da automatski generišu ili manipulišu *PDF* dokumentima iz svojih aplikacija, što je korisno u mnogim scenarijima, uključujući generisanje mnogih vrsta dokumenata. *Com.itextpdf.text.pdf.PdfPTable* deo je biblioteke *iText* koja omogućava programerima da generišu tabele u *PDF* dokumentima. Ova klasa omogućava kreiranje tabela sa više redova i kolona, te postavljanje različitih svojstava tabele, poput širine kolona, pozicioniranja elemenata u ćelijama, postavljanja granica tabele i ćelija, stilizovanja teksta, i još mnogo toga. *PdfPTable* omogućava fleksibilnu kontrolu izgleda *PDF* tabele i kreiranje *PDF* tabela. [6]

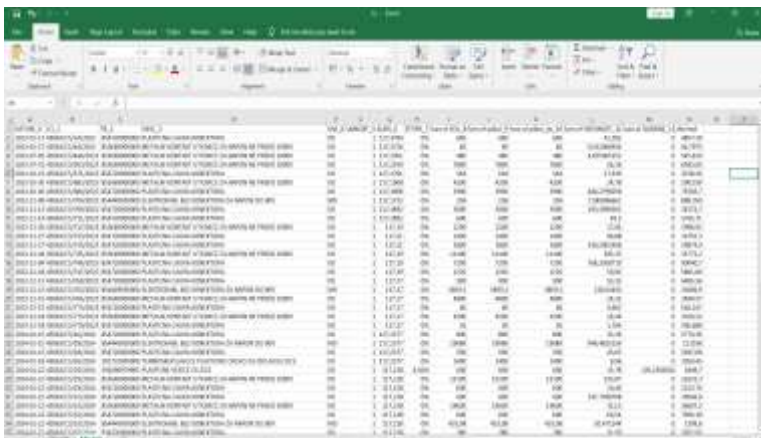
Org.apache.poi.ss.usermodel.Workbook predstavlja *Excel* radnu knjigu (*workbook*) u okviru *Apache POI* projekta. Ova radna knjiga predstavlja *Excel* dokument koji može sadržati više listova (*spreadsheet*-ova). Kroz *Workbook*, može se pristupiti svim listovima unutar radne knjige. Mogu se obavljati različite operacije, poput čitanja, pisanja i izmene podataka u ćelijama, te dodavanja novih listova i postavljanja svojstava radne knjige (kao što su naslovi, autor, itd.). Moguće su i mnoge druge operacije koje su potrebne prilikom manipulacije *Excel* dokumentima (Slike 1, 2, 3, 4, 5 i 6). *Row* i *Cell* su, takodje, i klase *Apache POI*-a koje su korišćene za čitanje redova i ćelija. [5]



Slika 1. Prikaz radne površine na kojoj se nalazi document - Na slici je prikazan dokument „ts.xlsx“ koji će biti učitán korišćenjem aplikacije i pretvoren u *PDF* tabelu.



Slika 2. Prikaz radnog lista 1 u kome se nalaze podaci - U dokumentu se nalaze dva radna lista. U prvom postoji samo jedna kolona.



Slika 3. Prikaz radnog lista 2 u kome se nalaze preostali podaci za obradu - U radnom listu 2 postoji više kolona koje će biti pročitane i konvertovane u PDF. Na drugoj adresi, napraviće se novi PDF dokument koji je kreiran od zadatog Excel dokumenta.

FINI		FINI		FINI		FINI	
naziv operacije		datum		datum		datum	
Tipovacki iznos prihod	Tarifa osuda	Kolostno	Vrednost valuta	Vrednost dinarima	Stopa osuda	Davanje jedn. dijela	Iznos dijela
1500,00	9120007900	1200,00	51,44	1000,15	0,20%		254,00
Ukupno							254,00

FINI		FINI		FINI		FINI	
naziv operacije		datum		datum		datum	
Tipovacki iznos prihod	Tarifa osuda	Kolostno	Vrednost valuta	Vrednost dinarima	Stopa osuda	Davanje jedn. dijela	Iznos dijela
281,40	9911015000	281,40	11,34	170,55	0,00%		70,20
Ukupno							70,20

FINI		FINI		FINI		FINI	
naziv operacije		datum		datum		datum	
Tipovacki iznos prihod	Tarifa osuda	Kolostno	Vrednost valuta	Vrednost dinarima	Stopa osuda	Davanje jedn. dijela	Iznos dijela
1400,00	8542000000	1400,00	08,20	700,55	0,00%		0,00
Ukupno							0,00

Ukupno vrednos-osuda novost _____ miliona.

Ukupno iznos osuda sa _____ miliona-ejina

100770,54	1549,07
Iznos ostalih davanja ostalog dijela	

Odgovorna lica

Slika 6. Prikaz poslednje strane PDF dokumenta sa podacima

A. Objašnjenje koda

Ključna metoda za preuzimanje podataka iz ćelija je `getCellContent()`. Preko nje podešava se vrednost svih promenljivih koje su na početku definisane (`getCellContent` se poziva tokom iteriranja). Preko ove metode možemo da pristupimo svakoj ćeliji u *Excel*-u, da je pročitamo i da tu vrednost sačuvamo u svoju promenljivu koju ćemo dalje koristiti (Slika 7).

```
public static void getCellContent(Cell cell){  
    switch (cell.getColumnIndex() {  
        case 0:  
            if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC) {  
                SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy");  
                datum? = sdf.format(cell.getDateCellValue());  
            } else if (cell.getCellType() == Cell.CELL_TYPE_STRING) {  
                datum? = cell.getStringCellValue();  
            }  
            break;  
        case 1:  
            if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC)  
                jc1Str = cell.getNumericCellValue()+"";  
            else jc1Str = cell.getStringCellValue();  
            if (jc1Str.isEmpty())  
                jc1Str = " ";  
            break;  
        case 2:  
            th = String.format("%.0f", cell.getNumericCellValue());  
            if (th.isEmpty()) {  
                th = " ";  
            }  
            break;  
        case 3:  
            desc = cell.getStringCellValue();  
            break;  
        case 4:  
            us = cell.getStringCellValue();  
            break;  
    }  
}
```

Slika 7. Prikaz dela implementiranog koda – Metoda *getCellContent()*

B. Slični primeri i rešenja

Pomenuto rešenje samo je jedan od načina na koje KonvertIz može da se koristi. U nastavku su prikazani još neki primeri koji mogu da se koriste u čitanju *Excel* dokumenata, pravljenju *PDF* tabelarnih izveštaja, kao i izradi izveštaja (Slike 8 i 9).

KOMENTAR ZA KOMPENZIRAJUĆI PROIZVOD

EXB C ID broj:
 00087703RW00 Naziv: PROGETTI BOSNA
 Tarifa broj: 040010100000 Broj (rešenja): LPII-1453/17
 Količina kompenziraćeg proizvoda: 1000 par Komentar broj:

Naziv vezane robe	Tarifa broj	Jed. m.	Utrošak u jedinici kompenziraćeg proizvoda	Ukupan utrošak
KOŽA ZA KOŽU	32099000000	LT	0.0175	3.24
TRAKA TAG RFIID	391010120000	PZ	2.0990	2199.00
SAMOLEPLJIVA GLAČANJA	39196000000	MO	0.0325	210.00
SAMOLEPLJIVA GLAČANJA OD NAJLONA	39196000000	NR	0.0040	4.32
OPREMA, JUNEĆA I TELEĆA KOŽA I POSTAVE OD KOŽE	410190100000	MO	0.2017	971.38
KOVANCI ZA ŠIVENJE	590810100000	CG	0.1429	61960.00
TEKSTILNO GLAČANJE ODCCI	590314000000	MO	0.0285	611.58
TEKSTILNE TRAKE ZA GLAČANJA	590010000000	MT	0.4104	4846.40
VEŽENE ETIKETE	591099000000	PZ	1.0000	2190.00
SAMOLEPLJIVA GLAČANJA	590320000000	MO	0.0199	127.44

Gore navedeni podaci su prikazani u odobrenju od nadležne carinarnice (javnika uposljeni)
 Datum: _____ Počet carinarnice (javnika uposljeni): _____
 Mjesto: _____

Slika 8. Prikaz još jednog PDF rešenja koje može da se napravi – izveštaj Komentar za kompenzirajući proizvod

```

// Pisanje u tekstualni fajl
private static void printToFile(String xlsxBuff, boolean pl) throws IOException {
    String fileName;
    if(pl)
        fileName = "packlist.txt";
    else fileName = "other.txt";
    try {
        PrintWriter outputStream = new PrintWriter(new FileWriter(fileName, true));
        outputStream.println(xlsxBuff);
        outputStream.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}

private static String getDateFromName(File f) {
    String dateString = f.getParentFile().getName();
    dateString = dateString.substring(0, 10);
    dateString.replace('_', '-');
    return dateString;
}

private static void printID(String line) throws IOException {
    String fileName = "packlistID.txt";
    try {
        PrintWriter outputStream = new PrintWriter(new FileWriter(fileName, true));
        outputStream.println(line);
        outputStream.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}

```

Slika 9. Prikaz koda koji omogućava pisanje u tekstualni dokument – Metoda *printToFile(String xlsxBuff, boolean pl)*

Pored ovakvih rešenja, *Apache POI* i *Itext* omogućavaju pravljenje jednostavnog *PDF* dokumenta, dodavanje slike u *PDF* dokument, čitanje sadržaja iz *PDF* dokumenta, te dodavanje metapodataka u *PDF* dokument. Naročitu primenu ima klasa *PrintWriter*, koja se koristi za pisanje podataka u tekstualne datoteke, konzolu ili druge izlazne tokove. To je korisna klasa za generiranje tekstualnih datoteka jer omogućava lako pisanje različitih tipova podataka (npr. stringova, brojeva).

IV ANALIZA REŠENJA KONVERTIZ

Konvertiz je samo jedan od načina na koji mogu da se implementiraju podaci iz *Excel* fajlova i od njih se kreira *PDF* tabela. *Apache POI* i *TextPDF* su biblioteke koje omogućavaju lako manipulisanje podacima. Korišćenjem ovih biblioteka mogu da se kreiraju razna rešenja koja mogu da postanu jedan od standarda za sve vrste čitanja i konverzije sličnih dokumenata i podataka. Obe biblioteke imaju dobru dokumentaciju i sadrže uputstvo za početnike što olakšava programerima da koriste ove biblioteke. Programeri mogu na taj način da ih prilagode svojim potrebama kako bi rukovali sa mnogim vrstama podataka. Biblioteke su veoma fleksibilne,

budući da nude veliki niz metoda za korišćenje. Smatraju se lako upotrebljivim za rad s *Office* dokumentima.

A. Kritički osvrt

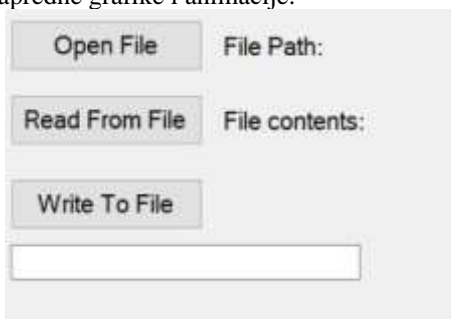
Izrađeno rešenje koristi biblioteku *Apache POI* za čitanje *Excel* dokumenata sa podacima i biblioteku *Itext.pdf* za pravljenje *PDF* dokumenata od tih podataka. Ovo rešenje je jednostavno programerima za upotrebu, ali mu nedostaje grafičko korisničko rešenje (*GUI* na engleskom), kako bi korisnik na intuitivan način mogao da koristi aplikaciju.

Nedostaje da bude izrađeno više rešenja za mnoge verzije izveštaja kako bi se poboljšala fleksibilnost. Neophodno je da se korisnicima prilagodi aplikacija prema njihovim potrebama, koji bi uključili opcije postavki i prilagođavanje rada aplikacije.

U radu nije pomenuto licenciranje. Zatim, neophodno je da podaci korisnika budu sigurni. To podrazumeva da se omogući autentifikacija – provera identiteta korisnika pre nego što mu dozvoli pristup aplikaciji, kao i autorizacija – provera prava korisnika i koje akcije može da preduzme unutar aplikacije. Nedostaje ažuriranje aplikacije – treba novije potrebe firme zadovoljiti ukoliko postoji potreba za izmenom.

B. Pravci daljeg rešenja

Rešenje se može proširiti novim funkcionalnostima koje bi služile za veći broj izveštaja. Da bi korisnicima bilo lakše za upotrebu, napravio bi se grafički korisnički interfejs (Slika 10). On može da se izradi korišćenjem biblioteke *JavaFx* – koja omogućava veliki izbor funkcionalnosti, pruža i rad s multimedijom, napredne grafike i animacije.



Slika 10. Primer jednog dela grafičkog korisničkog interfejsa u razradi

Pošto se često zahteva da se koristi velika količina podataka koja se obrađuje, dato rešenje će se integrisati sa bazama podataka. Može da se koristi *JDBC* (*Java Database Connectivity*), *API* za povezivanje i rad sa raznim bazama podataka. *JDBC* omogućava povezivanje sa različitim vrstama *SQL* baza podataka.

Da bi se povezali sa određenom bazom podataka, potrebno je dodati *JDBC driver* za tu bazu. To se obično postiže dodavanjem odgovarajuće zavisnosti u *pom.xml* dokument (bar ukoliko se koristi *Maven*).

Primer povezivanja na *MySQL* dat je na slici (Slika 11).

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.33</version>
</dependency>
```

Slika 11. Dodavanje zavisnosti korišćenjem *Maven*-a

Da bismo se povezali sa bazom podataka, potrebno je da se koristi metoda *DriverManager.getConnection()*. Takođe je potrebno da se navede *URL* baze podataka, *username*, te *password*, ako je potreban (Slika 12).

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {

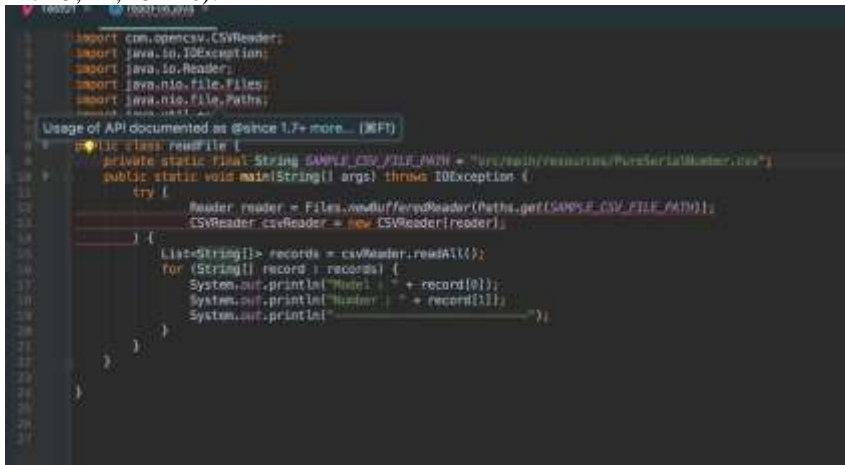
    // Podaci o bazi podataka
    private static final String URL = "jdbc:mysql://localhost:3306/baza_podataka"; // Promeniti na svoj URL baze podataka
    private static final String USER = "korisnicko_ime"; // Promeniti na svoje korisničko ime
    private static final String PASSWORD = "lozinka"; // Promeniti na svoju lozinku

    public Connection connect() {
        Connection conn = null;
        try {
            // Povezivanje sa bazom podataka
            conn = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("Povezan sa bazom podataka.");
        } catch (SQLException e) {
            System.err.println("Povezivanje sa bazom podataka nije uspelo: " + e.getMessage());
        }
        return conn;
    }
}
```

Slika 12. Primer povezivanja na bazu podataka u programskom jeziku *Java*

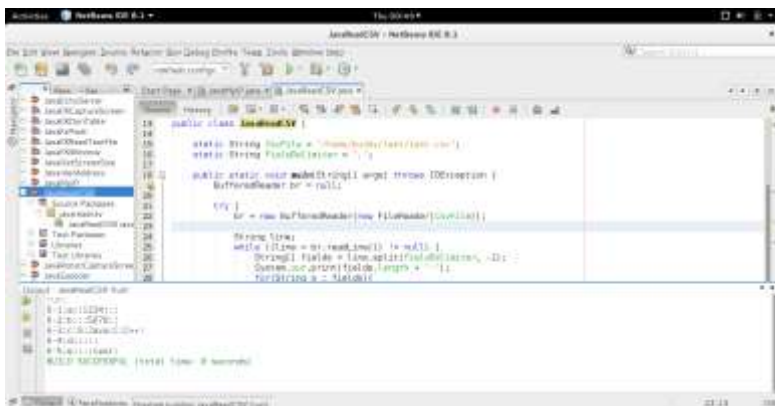
Kako bi se povećala fleksibilnost rešenja, može se uvesti podrška različitim formatima kao što su *JSON*, *XML* i *CSV*. Neke od biblioteka koje bi mogle da budu upotrebljene su: *Jackson*, *Gson*, *OpenCSV*, *JAXB*. Neki

primeri radnih okruženja pomenutih biblioteka dati su na slikama u nastavku (Slike 13, 14, 15 i 16).



```
1 import com.opencsv.CSVReader;
2 import java.io.IOException;
3 import java.io.Reader;
4 import java.nio.file.Files;
5 import java.nio.file.Paths;
6
7 Usage of API documented at @since 1.7+ more... [API]
8
9 public class readerFile {
10     private static final String SAMPLE_CSV_FILE_PATH = "src/main/resources/PurifierListMaster.csv";
11     public static void main(String[] args) throws IOException {
12         try {
13             Reader reader = Files.newBufferedReader(Paths.get(SAMPLE_CSV_FILE_PATH));
14             CSVReader csvReader = new CSVReader(reader);
15
16         }
17         List<String[]> records = csvReader.readAll();
18         for (String[] record : records) {
19             System.out.println("Name: " + record[0]);
20             System.out.println("Header: " + record[1]);
21             System.out.println("-----");
22         }
23     }
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Slika 13. Primer biblioteke *OpenCSV* u programskom jeziku *Java*



Slika 14. Drugi primer korišćenja biblioteke *OpenCSV* u programskom jeziku *Java*

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.x</version>
</dependency>
```

Slika 15. Dodavanje zavisnosti za biblioteku *Jackson* (*Maven*)

```
import com.fasterxml.jackson.databind.ObjectMapper;

public class JacksonExample {
    public static void main(String[] Args) {
        ObjectMapper mapper = new ObjectMapper();
        String json = "{\"name\":\"John\", \"age\":30}";

        try {
            Person person = mapper.readValue(json, Person.class);
            System.out.println(person.getName());
            System.out.println(person.getAge());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

class Person {
    private String name;
    private int age;
    // Getters and setters
}
```

Slika 16. Čitanje formata *JSON* korišćenjem biblioteka *Jackson*

LITERATURA

- [1] Vujaklija, D. (2008) Uvod u informacione tehnologije, Beograd: Fakultet organizacionih nauka
- [2] Mathur, V. (2022). What is Data Processing and Why is it Important?, AnalyticSteps, Dostupno na: <https://www.analyticsteps.com/blogs/what-data-processing-and-why-it-important>
- [3] Anwar, M. (2023). What is Data Conversion: Techniques, Tools, and Best Practices, Astera, Dostupno na: <https://www.astera.com/type/blog/data-conversion/>
- [4] IBM DataStage, Dostupno na: <https://www.ibm.com/products/datastage>
- [5] <https://poi.apache.org/>
- [6] <https://itextpdf.com/>

An approach to extraction, transformation and loading – The ConvertEase solution for the improvement of business data availability

Dorđe Ilijevski, dr Dušan Vujošević

Abstract — This paper focuses on enhancing the availability of data from business information systems. It analyzes technologies for extraction, transformation, and loading. The development of a software solution named *ConvertEase*, implemented using the *Apache POI* and *iText* libraries, is presented. This solution easily converts data from an *Excel* file into a *PDF* document. The need for such rendering of data arises from the increasing importance of data, which is one of the most significant resources for achieving competitive advantage and company success in the dynamics of modern business. Software for transactional data management, with features for data collection, management, storage, and transformation, greatly contributes to speeding up and automating business processes, as well as to the efficiency of decision-making. This leads towards its increasing application in the business world. The proposed solution, *ConvertEase*, is presented through an explanation of program logic, discussion of program code, interface examples, and examples of rendered documents. A case study presents The Prohibition of Refund or Exemption Based Customs Duty Assessment Report, as an example of reports generated by *ConvertEase*.