

Virtuelizacija Procesa

Predmetni projekat:

Kreiranje CSV datoteka na osnovu Xml baze
podataka

Članovi tima:

Bojana Bratić PR 28/2020

Marko Ratić PR 4/2020

Vladimir Simić PR 26/2020

Ilija Spasić PR 34/2020

Opis projektnog zadatka

Klijent je kompanija za prenos električne energije. Aplikacija treba da se bavi kreiranjem (jedne ili više) CSV datoteka sa podacima o prognozama i ostvarenjima električne energije na osnovu prosleđene XML datoteke.

Aplikacija očekuje da je XML datoteka sledećeg formata:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<rows>
  <row>
    <TIME_STAMP>2023-01-17 00:00</TIME_STAMP>
    <FORECAST_VALUE>6297.66</FORECAST_VALUE>
    <MEASURED_VALUE>6234</MEASURED_VALUE>
  </row>
  . . .
</rows>
```

Potrebno je da sva polja zadovoljavaju gore navedeni format. Pritom:

- Datum unutar TIME_STAMP polja treba da bude u formatu "yyyy-MM-dd HH:mm"
- FORECAST_VALUE je realan broj (tipa **float**)
- MEASURED_VALUE je realan broj (tipa **float**)

U suprotnom, servisni deo aplikacije čuva sva neslaganja u tabeli **Audits** koja sadrži odgovarajuće poruke.

Svi uspešno parsirani podaci se čuvaju u tabeli **Loads**, dok se obrađene datoteke čuvaju u tabeli **ImportedFiles**.

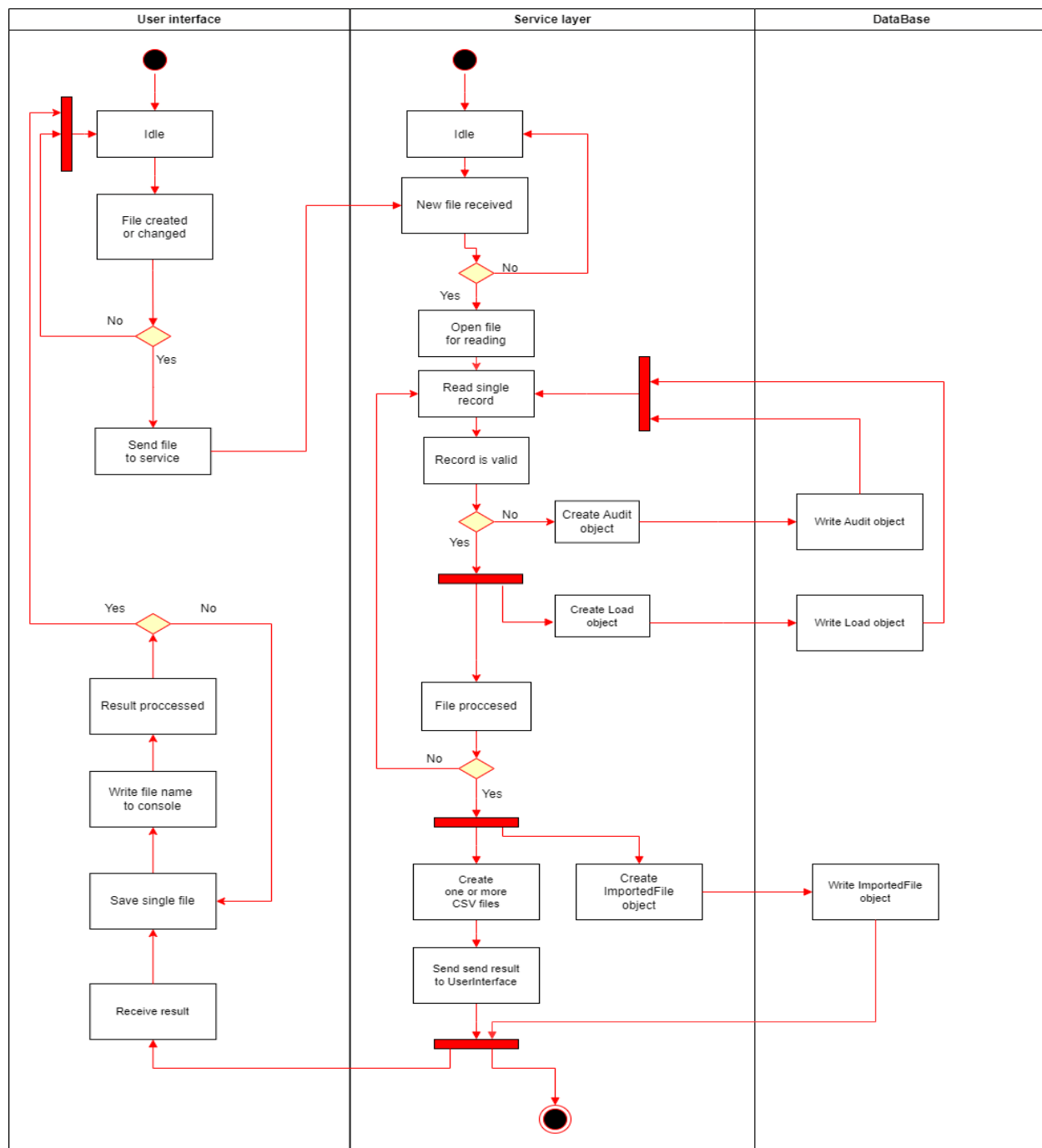
Aplikacija čuva podatke u **InMemory** bazi podataka.

Napomena: Obrađeni podaci su dostupni samo dok je **Service** aktivan.

Nakon obrade XML datoteke, pristigle od korisničkog interfejsa, servisni deo kreira i popunjava CSV datoteke koje se zatim prosleđuju nazad korisničkom interfejsu.

CSV datoteke se čuvaju na putanji iz **App.config** datoteke korisničkog interfejsa.

Arhitektura projektnog zadatka



Slika 1. - Dijagram kontrole toka aplikacije

Aplikacija je troslojna. Sadrži:

1. Korisnički interfejs
2. Servisni sloj
3. Baza podataka

Korisnički Interfejs

Potrebno je konfigurirati parametre Korisničkog Interfejsa u AppConfig.xml datoteci:

- **“xmlPath”** - predstavlja putanju XML datoteke, čije kreiranje ili promene prati Korisnički Interfejs
- **“savedPath”** - predstavlja putanju čuvanja pristiglih CSV datoteka od strane Servisa

Korisnički Interfejs čeka promene XML datoteke, nakon kojih istu šalje Servisu na obradu.

Servisni sloj

Potrebno je takođe konfigurirati parametar Servisa u AppConfig.xml datoteci:

- **“multipleCSV”** - predstavlja pokazatelj da li će se kreirati jedna ili više CSV datoteka, validne vrednosti su “true” ili “false”.

Pristiglu XML datoteku Servis interno prosleđuje modulu **XmlHandler**, čiji je zadatak učitavanje same datoteke, proveru njene strukture i parsiranje samih **Load** podataka. Svaki validan **Load** objekat se upisuje u bazu podataka, dok za neispravane upisuje **Audit** objekat sa odgovarajućom porukom.

Nakon obrade XML datoteke, servis dobija povratnu informaciju o uspešnosti obrade. Servis je dalje zadužen za kreiranje jedne ili više CSV datoteka i njihovo slanje korisničkom interfejsu.

Servis takođe sadrži pomoćni modul **DBPrinter**, čiji je zadatak ispis tabela baze podataka na izvršnu konzolu. Komande za sam ispis su:

- **“F5”** - komanda za štampanje tabele **Audits**
- **“F6”** - komanda za štampanje tabele **Loads**
- **“F7”** - komanda za štampanje tabele **ImportedFiles**

Baza podataka

Baza podataka je implementirana kao skup rečnika (**ConcurrentDictionary**) gde je ključ tekući indeks, a vrednost jedan od modela podataka (**Audit**, **Load**, **ImportedFile**). Baza se pristupa pomoću staticnog polja **Instance**, koje predstavlja instancu same baze. Ovim je baza sinhronizovana između niti, i upotrebom konkurentnih rečnika je osiguran bezbedan unos.

Opis interfejsa i osnovnih funkcionalnosti

Servis implementira interfejs **IFileHandling**, i koristi pomoćne klase **SendFileOptions** i **ReceiveFileOptions**.

```
namespace Common.FileHandling
{
    [ServiceContract]
    4 references
    public interface IFileHandling
    {
        [OperationContract]
        [FaultContract(typeof(FileHandlingException))]
        2 references
        ReceivedFileOptions SendData(SendFileOptions file);
    }
}
```

Osnovna i jedina metoda interfejsa je "SendData" koja kao parametar očekuje objekat **SendFileOptions** a kao rezultat vraća objekat **ReceivedFileOptions**.

SendFileOptions sadrži sledeća polja:

- "MS" - podaci datoteke koja se šalje
- "FileName" - naziv same datoteke

ReceivedFileOptions sadrži sledeća polja:

- "ReceivedFiles" - rečnik rezultujućih datoteka koje se šalju korisniku
- "NumOfFiles" - broj gore navedenih datoteka
- "ResultMessage" - tip povratne poruke
- "Message" - povratna poruka

Opis korišćenih tehnologija

Projekat je rađen pomoću Microsoft Windows Communication Protocol (WCF) protokola, pisan u programskom jeziku C#.

Korišćene tehnologije:

- .NET Framework verzija **4.7.2**
- Windows Communication Protocol
- C# 8

Zaključak

Rešenje je zadovoljavajuće za primene manjih razmera. Jedan od oblika unapređenja je uvođenje neke vrste ograničenja klijentskih zahteva, a samim tim i zaštita od “poplava”. Kako bi rešenje bilo skalabilno, neminovno je rekonstruisanje modula baze podataka kao nezavisnog servisa.

Takođe može biti razmatrano, ukoliko ima potrebe na osnovu saobraćaja, uvođenje neke vrste balansiranja zahteva. Uvođenjem novog sloja (load balancer / proxy) olakšavamo skaliranje samog sistema, ali je potrebno voditi računa o sinhronizaciji podataka između baze i samih servisa. Ovim je omogućeno uvođenje redundantnosti, kao i rezervnih kopija (hot / cold swap) bez većih poteškoća.