

Zadatak 2

Kreiranje CSV datoteka na osnovu XML baze podataka

Klijent je kompanija za prenos električne energije. Aplikacija treba da se bavi kreiranjem CSV datoteka sa podacima o prognozama i ostvarenjima potrošnje električne energije, na osnovu prosleđene XML datoteke.

Slede korisnički zahtevi.

1. Poslovna logika

Aplikacija se sastoji od servisa i od klijentske aplikacije. Servis i klijentska aplikacije komuniciraju putem WCF-a. Klijentska aplikacija ima u konfiguracionoj datoteci (App.config) definisanu putanju i naziv XML datoteke. U XML datoteci nalaze se podaci o prognoziranoj potrošnji i o ostvarenoj potrošnji po satima. Klijentska aplikacija osluškuje definisanu putanju. Ukoliko se na definisanoj putanji kreira datoteka sa definisanim nazivom, ili datoteka sa definisanim nazivom već postoji ali se promenio „Changed“ atribut datoteke, datoteka se šalje servisu.

Kada je datoteka pristigla servisu, servis izvršava sledeće zadatke:

- Na osnovu pristigle datoteke, servis kreira kolekciju objekata klase **Load**. Podaci za svaki sat iz XML datoteke reprezentovani su jednim **Load** objektom. **Load** objekti se kreiraju samo za datum i sat za koji već nisu prethodno kreirani. Objekti se upisuju u In-Memory bazu podataka.
- Ako se za neki sat uoči greška, kreira se novi **Audit** objekat koji se upisuje u bazu podataka. Greška se može desiti ako je podatak nevalidan (na primer nevalidno float ili DateTime polje), ako nedostaju neki podaci ili ako postoji greška u strukturi XML datoteke.
- Kada se kreirao i poslednji **Load** objekat iz pristigle XML datoteke, aktivira se događaj (*Event*) kojim kreira jedna ili više CSV datoteka. Ukoliko se kreira samo jedna CSV datoteka, podaci za sve datume iz XML datoteke nalaziće se u toj jednoj CSV datoteci. Ukoliko se kreira više CSV datoteka, za svaki datum iz XML datoteke kreira se po jedna CSV datoteka. Odluka da li će biti kreirana jedna ili više CSV datoteka donosi se na osnovu podešavanja u App.config datoteci servisa.
- Za svaku obrađenu XML datoteku kreira se objekat *ImportedFile* i upisuje se u bazu podataka.
- Servis šalje CSV datoteke (jednu ili više njih) kao rezultat klijentskoj aplikaciji.

Kada primi CSV datoteke, klijentska aplikacija čuva datoteke na putanji koja je posebno definisana u App.config datoteci klijentske aplikacije. Za svaku primljenu datoteku ispisuje se poruka u konzoli, koja sadrži pun naziv (sa putanjom) primljene datoteke.

2. Model podataka

Model podataka obuhvata sledeće klase:

- **Load** (polja: Id, Timestamp, ForecastValue, MeasuredValue)
- **ImportedFile** (polja: Id, FileName)
- **Audit** (polja: Id, Timestamp, MessageType, Message)
MessageType može da ima vrednosti Info, Warning i Error

3. Implementacija baze podataka

Baza podataka treba da bude implementirana kao In-Memory baza podataka.

In-Memory baza podataka implementirana je kroz strukturu podataka Dictionary ili ConcurrentDictionary. Svaka tabela je implementirana kroz jedan (Concurrent)Dictionary, pri čemu je Key ID reda u tabeli, a Value je objekat odgovarajuće klase (*Load*, *ImportedFile* i *Audit*). Podaci u In-Memory bazi podataka postoje samo dok je servis pokrenut.

4. Tehnički i implementacioni zahtevi

Aplikacija treba da bude u višeslojnoj arhitekturi. Aplikacija treba da sadrži najmanje sledeće komponente:

- baza podataka (In-Memory baza podataka)
- servisni sloj
- korisnički interfejs – konzolna aplikacija
- Common – projekat koji je zajednički za sve slojeve

Komunikacija između klijentske aplikacije i servisa obavlja se putem WCF-a

Rad sa datotekama treba da bude implementiran tako da se vodi računa o održavanju memorije, korišćenjem Dispose paterna

Aktiviranje kreiranja CSV datoteka izvršava se korišćenjem Event-a i Delegate-a. Delegat treba da pokazuje na odgovarajući metod – kreiranje jedne datoteke ili više datoteka.

Za aplikaciju treba da postoje sledeći dokumenti:

- User manual
- Dokument u kom je opisana arhitektura aplikacije