

Dokumentacja Projektu

Przetwarzanie dokumentów XML

Mateusz Nizio

20 czerwca 2022

1. Opis projektu

Celem projektu było opracowanie oraz implementacja API umożliwiającego przetwarzanie dokumentów XML. Opracowane API udostępnia opcję zapisu oraz usuwania dokumentu w bazie. Posiada również możliwość wyszukania wybranych dokumentów po określonych węzłach i/lub atrybutach oraz modyfikację wybranych węzłów i/lub atrybutów. Projekt składa się z aplikacji konsolowej która korzysta z przygotowanego API oraz testów jednostkowych, które sprawdzają poprawność działania programu.

2. Opis aplikacji

Aplikacja konsolowa została napisana w języku C# z wykorzystaniem aplikacji Visual Studio 2019. Została ona przetestowana na lokalnym serwerze stworzonym przy użyciu programu Microsoft SQL Server Management Studio 2018.

Na samą aplikację składa się kilka klas (Menu, Connection, DatabaseFunc oraz main). Po uruchomieniu program automatycznie uruchamia metodę z klasy Connection i następuje próba połączenia się ze wskazaną bazą danych. Następnie tworzone jest menu. Teraz użytkownik może korzystać z funkcjonalności jakie oferuję API. Aby sprawdzić dostępne funkcje wystarczy wpisać w konsolę polecenie „help”. Menu składa się ze stringów do których „zbindowane” są konkretne funkcje. Specjalnie napisane handlers rozpoznają wejście z konsoli i przypisują je do odpowiedniej funkcji z klasy DatabaseFunc, która następnie jest

uruchamiana ze wskazanymi parametrami. Każda funkcja zwraca rekurencyjnie status, który informuję użytkownika o tym czy jego polecenie się udało. Wszystkie zapytania do bazy zostały również zabezpieczone blokami try/catch, i w przypadku niepoprawnego polecenia użytkownik zostanie o tym powiadomiony.

3. Baza danych

Na potrzeby projektu w bazie danych została utworzona jedynie jedna tabela zawierają dwie kolumny:

-**friendly_name** (VARCHAR 255) – czyli nazwa, która użytkownik może nadać dokumentowi oraz identyfikować dokument po tej nazwie

-**XMLColumn (xml)** – kolumna zawierająca dokumenty XML dodane przez użytkownika

4. Funkcję udostępnione w API

1. **Help** – jest to polecenie zwracające wszystkie dostępne funkcje w API. Jej działanie polega na użyciu Console.WriteLine() z języka C#. Aby go użyć należy w konsoli wpisać „help”

```
Polaczono z baza danych.

Dostepne komendy:
help - wypisanie dostepnych komend
show - wypisanie wszystkich przechowywanych dokumentów
insert - dodanie nowego pliku xml
delete - usuniecie pliku xml
print_node - wypisanie wezla w danym pliku
modify - zmiana wartosci wskazanego atrybutu
search - zwraca wszystkie dokumenty w których wystepuje wskazany wezel
exit - wyjscie z programu
```

2. **Show** – funkcja która zwraca wszystkie dokumenty wraz z ich nazwami z bazy danych i wypisuje je na ekran. Jej działanie polega na wywołaniu SqlCommand która wysyła zapytanie i zwraca całą tabelę z danymi użytkownika. Aby go użyć należy wpisać „show”

```
Wpisz nowa komende:
show
1.adsasd      <node>test</node>

2.testowy     <node />

3.nowy        <user><user_id>1</user_id><fname>Josef</fname><lname>Brown</lname><email_>j00@adventure-works.com</email_></user>

4.t1          <Names><Name><FirstName>Jan</FirstName><LastName>Smith</LastName></Name><Name><FirstName>James</FirstName><LastN
ame>White</LastName></Name></Names>

5.sh          <Names><Name><FirstName>paluch</FirstName><LastName>Smith</LastName></Name></Names>

6.am          <email><email_>to</email_></to><from>Fence</from></from><heading>Hello</heading><body>Hello, brother, how are you?</body>
```

3. **Insert** – funkcja dzięki której użytkownik może dodać do bazy nowy dokument. Przed dodaniem dokumentu następuje walidacji, czy nazwa dokumentu nie jest pusta oraz czy sam dokument ma w sobie zawartość. Działanie polega na użyciu SqlCommand która podejmuje próbę insertu nowego elementu do bazy danych. Aby go użyć należy w konsoli wpisać: „insert [friendly_name] [bezwzględna ścieżka do pliku]/[zawartość pliku]”, gdzie friendly_name to nazwa jaką chcemy nadać naszemu plikowi np: insert nowy_plik <node>test</node>

```
Wpisz komende:
insert nowy_dok <ksiazka>tolkien</ksiazka>
Poprawnie dodano do bazy
Wpisz nowa komende:
```

4. **Delete** – funkcja umożliwiająca usunięcie pliku o podanej nazwie (friendly_name) z bazy danych. Działanie polega na użyciu SqlCommand która podejmuje próbę usunięcia elementu z bazy danych. Użycie funkcji z konsoli: „delete [friendly_name]” np. delete nowy_plik

```
Wpisz nowa komende:
delete nowy_dok
Poprawnie usunięto wiersz
Wpisz nowa komende:
```

5. **Print_doc** – funkcja wypisująca na konsolę cały plik XML o podanej nazwie (friendly_name). Funkcja korzysta z SqlCommand i zwraca wiersz z dokumentem XML gdzie podana nazwa to nazwa wskazana przez użytkownika. Użycie funkcji z konsoli: „print_doc [friendly_name]”

```
Wpisz nowa komende:  
print_doc n4  
<?xml version="1.0" encoding="ibm850"?>  
<bookstore>  
  <book category="cooking">  
    <title lang="en">Everyday Italian</title>  
    <author>Giada De Laurentiis</author>  
    <year>2005</year>  
    <price>30.00</price>  
  </book>  
  <book category="children">  
    <title lang="en">Harry Potter</title>  
    <author>J K. Rowling</author>  
    <year>2005</year>  
    <price>29.99</price>  
  </book>  
  <book category="web">  
    <title lang="en">XQuery Kick Start</title>  
    <author>James McGovern</author>  
    <author>Per Bothner</author>  
    <author>Kurt Cagle</author>  
    <author>James Linn</author>  
    <author>Vaidyanathan Nagarajan</author>  
    <year>2003</year>
```

6. **Print_node**- funkcja wypisująca wszystkie znalezione węzły danego typu w dokumencie XML o podanej nazwie. Użycie funkcji z konsoli: „print_node [friendly_name] [węzeł]”
np. print_node nowy_plik /Names/Name/FirstName

```
print_node n4 /bookstore  
<bookstore>  
  <book category = cooking>  
    <title lang = en>  
      <Everyday Italian>  
    <author>  
      <Giada De Laurentiis>  
    <year>  
      <2005>  
    <price>
```

7. **Search** – funkcja zwracająca wszystkie znalezione dokumenty w których występuje węzeł o podanej nazwie. Funkcja iterując po całej tabeli sprawdza czy w danym dokumencie występuje wskazany przez użytkownika wiersz (użycie operatora EXISTS). Użycie funkcji z konsoli: „search [węzeł]”

```
Wpisz nowa komende:  
search /Imiona/Test  
Wzly znalezione w dokumentach o nazwach:  
n14
```

8. **Modify** – funkcja modyfikująca w pliku o podanej nazwie dany element na inny podany przez użytkownika. Funkcja korzysta z instrukcji UPDATE oraz ‘replace value of’. Użycie z konsoli: „modify [friendly_name] [atr1] [atr2]”

```
Wpisz nowa komende:  
modify t1 (/Names/Name/FirstName[1]/text())[1] "pawel"  
  
<Jan>  
Poprawnie zmodyfikowano wartosc  
Wpisz nowa komende:  
print_doc t1  
<?xml version="1.0" encoding="ibm850"?>  
<Names>  
  <Name>  
    <FirstName>pawel</FirstName>  
    <LastName>Smith</LastName>  
  </Name>  
  <Name>  
    <FirstName>James</FirstName>  
    <LastName>White</LastName>
```

5. Testy Jednostkowe

Dla projektu zostały przeprowadzone testy jednostkowe które sprawdziły poprawność testowania funkcji udostępnionych w API. Pokrycie testów ,według narzędzia Test Explorer w Visual Studio, to około 73%. Testy sprawdzają zarówno przypadki oczywiste(poprawny dokument, poprawna nazwa itp.) jak i skrajne(brak dokumentu, brak nazwy, pusty dokument, błędy itp.).

Test	Duration	Traits	Error...
▲ ✓ ProjektBDTests (6)	538 ms		
▲ ✓ ProjektBDTests (6)	538 ms		
▲ ✓ DodajDokumentTest (1)	423 ms		
✓ DodanieDoBazy	423 ms		
▲ ✓ ModyfikacjaTest (1)	55 ms		
✓ Modyfikuj	55 ms		
▲ ✓ PolaczenieTest (1)	2 ms		
✓ Polaczenie	2 ms		
▲ ✓ SzukanieDokumentowTest (1)	43 ms		
✓ Szukanie	43 ms		
▲ ✓ UsunDokumentTest (1)	8 ms		
✓ UsuniecieZBazy	8 ms		
▲ ✓ WypiszWezlyTest (1)	7 ms		
✓ WypisanieWezlow	7 ms		

6. Wnioski

Napisana aplikacja spełnia wymagania przedstawione przez prowadzącego przedmiot. Dalszy rozwój aplikacji mógłby polegać na implementacji większej liczby funkcji dostępnych dla użytkownika oraz samym zabezpieczeniu pliku użytkownika przed odczytem ze strony innych. Sama aplikacja mogłaby być także bardziej „rozciągliwa” tzn. dobierać odpowiednie tabele itp. w zależności od tego jaki użytkownik się do niej zaloguje.

7. Literatura

https://www.w3schools.com/xml/dom_nodetype.asp

<https://docs.microsoft.com/pl-pl/azure/azure-sql/database/connect-query-dotnet-visual-studio?view=azuresql>

<https://docs.microsoft.com/en-us/dotnet/api/system.xml.xmldocument?view=net-6.0>

<https://www.c-sharpcorner.com/article/reading-and-writing-xml-in-C-Sharp/>

<https://zetcode.com/csharp/xmldocument/>