

Przetwarzanie danych XML

Dokumentacja projektu

Marcin Urbanowicz 20 kwietnia 2022

1. Założenia projektu i zakres funkcjonalności

Stworzenie API umożliwiającego przetwarzanie dokumentów XML. API udostępnia opcje modyfikacji wybranego dokumentu. Poprzez modyfikację mam tu na myśli operacje takie jak wyszukiwanie określonych węzłów, ich usuwanie, dodawanie nowych i zmiana zawartości już istniejących. Udostępniona jest też możliwość tworzenia nowych plików.

2. Wykorzystane technologie

Aplikacja powstała w oparciu o technologię Windows Forms, która pozwala na stworzenie desktopowych aplikacji w oparciu o język C#. Pliki xml były przechowywane i przetwarzane w oparciu o metody pochodzące z biblioteki Linq. Aplikacja wczytuje pliki xml poprzez fildialog.

3. Sposób uruchomienia

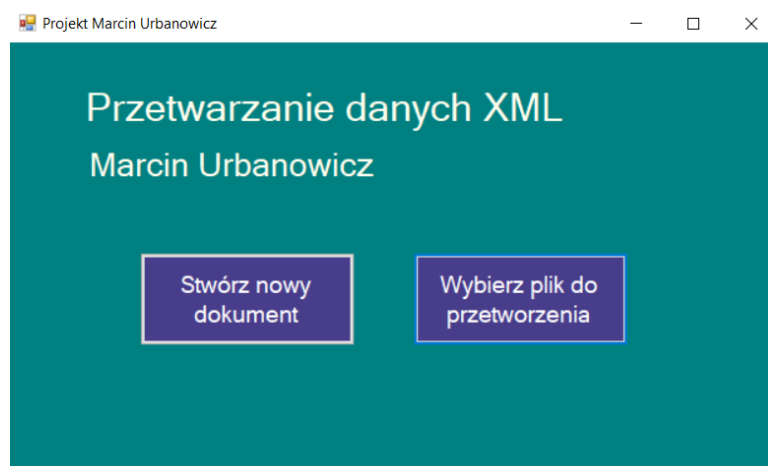
Aplikacja powstała w programie Visual Studio 2019 i też za jej pomocą sugeruję jej uruchomienie (Dostarczony jest plik .sln). Dodatkowo dostarczam dwa przykładowe pliki xml (studenci.xml, pracownicy.xml), które mogą posłużyć do sprawdzenia działania.

4. Opis implementacji / Podręcznik użytkownika

Przetwarzanie danych odbywa się w oparciu o GUI i do odpowiednich przycisków w odpowiednich formularzach przypisane są poszczególne funkcjonalności, przez które teraz przeprowadzę.

4.1. Start aplikacji

Po starcie aplikacji pojawia się nam następujące okno:



Rysunek 1: Okno startowe aplikacji

Z tego poziomu dostępne są dwie opcje (rysunek 1):

1. **Stwórz nowy dokument** (Punkt 4.2)
2. **Wybierz plik do przetworzenia** (Punkt 4.3)

4.2. Tworzenie nowego dokumentu

Po naciśnięciu opcji stwórz nowy dokument pojawia się nam następujący interfejs:

Rysunek 2: Formularz tworzenia nowych dokumentów xml

Mamy dwa przyciski:

1. **Stwórz dokument** - aby stworzyć dokument trzeba podać jego nazwę i treść. Tytuł, aby być poprawnym musi zawierać końcówkę .xml, natomiast zawartość musi być w pełni poprawnym dokumentem xml (Wszystkie znaczniki muszą być zamknięte). Jeżeli któreś z opisanych wyżej wprowadzeń jest niepoprawne program pokaże stosowny komunikat o błędzie oraz nie przejdzie dalej.
2. **Powrót** – wraca do poprzedniego okna (Punkt 4.1)

Nowoutworzone pliki są domyślnie zapisywane w folderze z projektem: **PrzetwarzanieDanychXML\bin\Debug**

4.3. Przetwarzanie dokumentów

Jeżeli w punkcie 4.1 wybraliśmy opcję by wybrać plik do przetworzenia (wymaga wybrania istniejącego pliku z rozszerzeniem xml). Wtedy pojawia się nam poniższe okno.



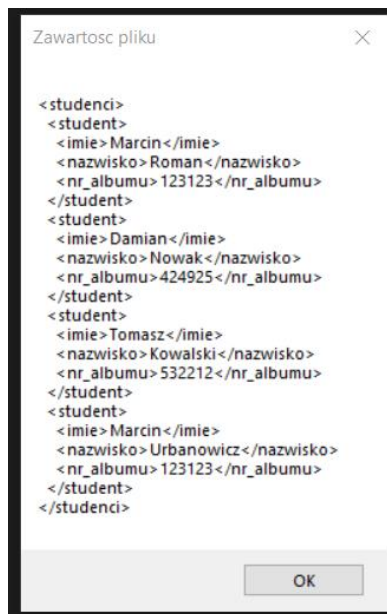
Rysunek 3: Okno z operacjami do przetwarzania plików.

Dostępne jest 7 operacji, które każde po kolei omówię (na podstawie pliku studenci.xml):

1. Sprawdź zawartość pliku
2. Dodaj węzeł
3. Modyfikuj węzeł
4. Usuń węzeł
5. Zapisz zmiany w obecnym pliku
6. Wyszukiwanie węzłów
7. Wybierz inny plik

4.3.1. Sprawdź zawartość dokumentu

Po wybraniu tej opcji pojawi nam się obecna zawartość wybranego pliku:



Rysunek 4: Wyświetlenie obecnej treści pliku

4.3.2. Dodaj węzeł

Po wybraniu tej opcji pojawia się nam następujący formularz :

A screenshot of a window titled "Marcin Urbanowicz Projekt" with standard window controls. The main content area has a teal background and is titled "Dodawanie węzła" (Adding node). It contains three input fields with labels "imie", "nazwisko", and "nr_albumu" to their left. At the bottom of the form are two buttons: "Dodaj węzeł" (Add node) and "Powrót" (Return).

Rysunek 5: Formularz do dodawania

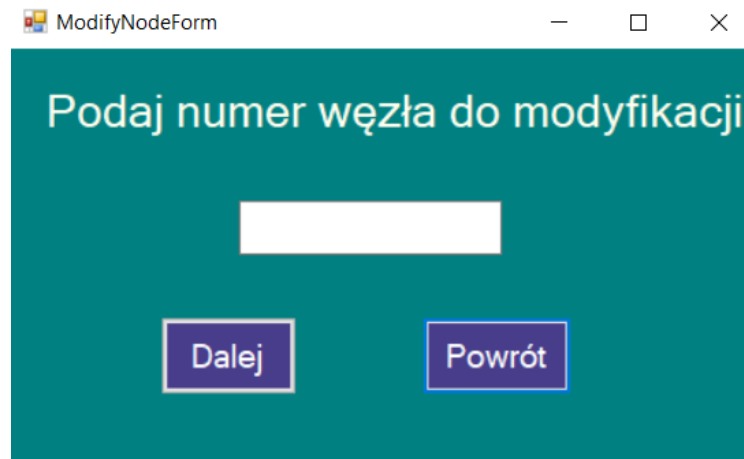
Formularz ma ograniczenie w postaci tego, że maksymalna liczba elementów, jakie mogą być w węźle to 5 (na więcej nie ma miejsca). Nazwy są zgodne z tymi pochodzącymi z węzłów. Nie jest dopuszczone aby pola pozostały puste, co jest walidowane.

Mamy dostępne dwie operacje:

1. **Dodaj węzeł** – dodaje nowy na koniec dokumentu
2. **Powrót** – wraca do poprzedniego okna (*Punkt 4.3*)

4.3.3. Modyfikuj węzeł

Po wybraniu tej opcji pojawia się nam następujący formularz :



Rysunek 6: Wybranie węzła do modyfikacji

W tym formularzu musimy podać numer węzła, który chcemy zmodyfikować, przy czym niestety nie ma walidacji sposobu jaki został wprowadzony (poza tym, że nie może być puste i musi być liczbą całkowitą), więc trzeba podać poprawną liczbę.

Mamy do dyspozycji dwie opcje:

1. **Dalej** – przechodzi do następnego formularza
2. **Powrót** – wraca do poprzedniego okna (*Punkt 4.3*)

Po wybraniu opcji **Dalej** przechodzimy do następnego formularza:



Rysunek 7: Modyfikacja wybranego węzła

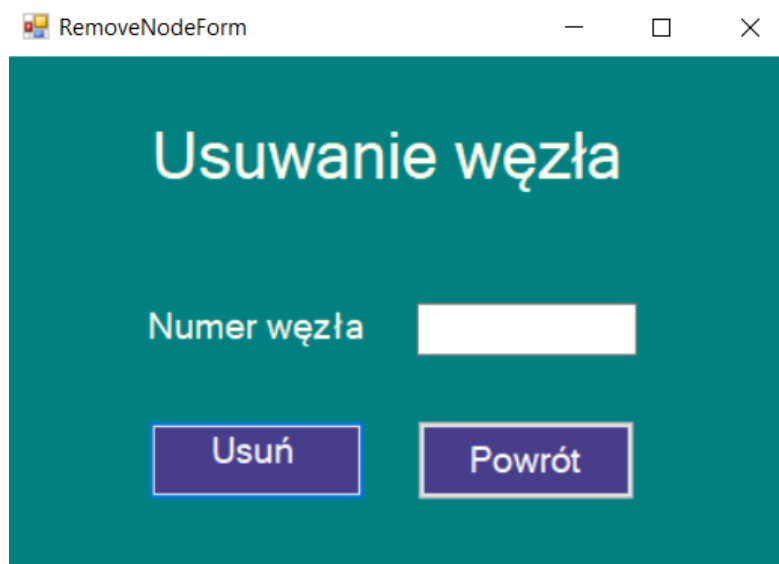
Podobnie jak przy formularzu dodawania węzłów (punkt 4.3.2) ma on ograniczenie w postaci tego, że maksymalna liczba elementów, jakie mogą być w węźle to 5 (na więcej nie ma miejsca). Nazwy są zgodne z tymi pochodzącymi z węzłów. Nie jest dopuszczone aby pola pozostały puste, co jest sprawdzane.

Mamy na oknie do wyboru trzy opcje:

1. **Modyfikuj** – w przypadku poprawnego wprowadzenia danych w pole tekstowe dokonuje operacji modyfikacji węzła i pokazuje się komunikat o powodzeniu informacji. W przeciwnym wypadku wyskakuje komunikat o błędzie
2. **Zobacz wybrany węzeł** – pozwala sprawdzić wybrany do modyfikacji węzeł, po udanej modyfikacji wybranie tej opcji pokaże zmieniony węzeł
3. **Zmień węzeł** – jest to operacja powrotu do poprzedniego formularza (*rysunek 6*)

4.3.4. Usuń węzeł

Po naciśnięciu przycisku pojawia się następujący formularz:



Rysunek 8: Usuwanie węzła

Formularz ten służy do usuwania węzłów na podstawie Id. Podobnie jak przy modyfikowaniu podawany numer węzła nie jest w żaden sposób walidowany, więc trzeba podać poprawny.

Dostępne są dwie operacje:

1. **Usuń** – usuwa węzeł o danym id
2. **Powrót** – wraca do poprzedniego formularza (*Punkt 4.3*)

4.3.5. Zapisz zmiany w obecnym pliku.

Domyślnie operacje w aplikacji nie modyfikują wybranego do przetwarzania pliku. Jeśli chcemy zapisać zmiany w pliku musimy nacisnąć ten przycisk.

4.3.6. Zapisz zmiany w obecnym pliku.

Po naciśnięciu wyświetla się nam następujący formularz:



Rysunek 9: Formularz do wyszukiwania

Formularz ten służy do wyszukiwania węzłów po poszczególnych zawartościach elementu. W pole tekstowe wpisujemy element którego szukamy.

Na formularzu są dwa przyciski:

1. **Wyszukaj** – Dokonuje operacji wyszukania i pokazuje jego wynik
2. **Powrót** – wraca do poprzedniego formularza (*Punkt 4.3*)

4.3.7. Wybierz inny plik

Ten przycisk cofa nas do pierwszego okna aplikacji (*Punkt 4.3*)

5. Podsumowanie

Biblioteka Linq dostarcza wiele bardzo przydatnych i bardzo intuicyjnych narzędzi do przetwarzania danych xml. Pomimo pewnych początkowych trudności z implementacją, czytanie dokumentacji pozwoliło mi rozwiązywać powstające na przestrzeni pisania problemy.

6. Literatura

Przy implementacji aplikacji posługiwałem się przede wszystkim dokumentacją dostarczoną przez Microsoft. Poza tym posługiwałem się również poradnikami jak tworzyć aplikacje z użyciem Microsoft Forms.

- <https://docs.microsoft.com/en-us/dotnet/csharp/>
- <https://docs.microsoft.com/pl-pl/dotnet/api/system.xml.linq?view=net-6.0>
- <https://docs.microsoft.com/pl-pl/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>

