

Dokumentacja projektu: Przetwarzanie dokumentów XML

Opis problemu i funkcjonalności API

Celem projektu jest stworzenie biblioteki umożliwiającej zarządzanie dokumentami XML w bazie danych SQL Server. Biblioteka udostępnia zestaw funkcji (API), które umożliwiają:

- dodawanie dokumentu XML,
- usuwanie dokumentu,
- pobieranie wszystkich dokumentów,
- wyszukiwanie dokumentów po wyrażeniu XPath,
- modyfikację zawartości węzłów i atrybutów,
- dodawanie/usuwanie atrybutów i węzłów,
- podmianę zawartości.

Typy danych i metody API

Typy danych:

- Id: INT, klucz główny
- Name: NVARCHAR(100)
- Content: XML
- CreatedAt: DATETIME

Udostępnione metody:

- GET /api/xml/all
- POST /api/xml/save
- DELETE /api/xml/delete/:id
- GET /api/xml/searchByXPath
- POST /api/xml/insert-node
- POST /api/xml/delete-node
- POST /api/xml/replace-value
- POST /api/xml/insert-attribute
- POST /api/xml/delete-attribute

Implementacja API w bibliotece

Logika aplikacji znajduje się w klasie XmlService. Metody tej klasy wykorzystują bibliotekę `mssql` do połączenia z bazą danych SQL Server i wykonania zapytań T-SQL, głównie z użyciem metody `.modify()` na typie danych XML. Operacje są wykonywane dynamicznie na podstawie danych wejściowych od klienta.

Testy jednostkowe

Przygotowano skrypt testowy testXmlApi.js, który testuje:

- dodanie dokumentu XML
- modyfikację atrybutu
- usunięcie węzła
- dodanie nowego węzła
- usunięcie atrybutu
- usunięcie dokumentu

Testy wykonują żądania HTTP z wykorzystaniem biblioteki axios i zawierają instrukcje assert, weryfikujące poprawność odpowiedzi.

Przykładowa aplikacja

Frontend aplikacji został zrealizowany w Angularze 18. Umożliwia on:

- dodawanie dokumentu XML,
- podgląd listy dokumentów,
- wyszukiwanie po XPath,
- wykonanie operacji na węzłach i atrybutach za pomocą formularza modalnego.

Podsumowanie i wnioski

Projekt zapewnia pełną funkcjonalność zarządzania dokumentami XML w bazie danych, a zaproponowane API może być z powodzeniem wykorzystane w innych aplikacjach.

Literatura

- Microsoft Docs: <https://learn.microsoft.com/en-us/sql/relational-databases/xml/xml-data-sql-server>
- Dokumentacja biblioteki `mssql`: <https://www.npmjs.com/package/mssql>
- Dokumentacja Angular: <https://angular.io>

Skrypty T-SQL wykorzystane w projekcie

```
CREATE DATABASE XmlDocuments;

USE XmlDocuments;
GO;

CREATE TABLE XMLDocuments (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    Name NVARCHAR(100),
    Content XML NOT NULL,
    CreatedAt DATETIME DEFAULT GETDATE()
);

-- Wylistowanie wszystkich dokumentów
SELECT * FROM XmlDocuments

-- Zapisanie nowego
INSERT INTO XmlDocuments (Name, Content)
    OUTPUT INSERTED.Id, INSERTED.Name, INSERTED.Content, INSERTED.CreatedAt
VALUES ({name}, {xmlContent})

-- Usuwanie dokumentu
DELETE FROM XmlDocuments WHERE Id = {id}

-- Wyszukanie po XPath
SELECT
    Id,
    Name,
    Content,
    CreatedAt,
    Content.query('{safeXPath}') AS MatchedFragment
FROM XmlDocuments
WHERE Content.exist('{safeXPath}') = 1

-- Wstawianie węzła
UPDATE XmlDocuments
    SET Content.modify('insert {safeNode} {position} {safeXPath}')
    OUTPUT INSERTED.*
    WHERE Id = @id

-- Usuwanie węzła
UPDATE XmlDocuments
    SET Content.modify('delete {safeXPath}')
    OUTPUT INSERTED.*
    WHERE Id = @id

-- Zmiana węzła/atributu
UPDATE XmlDocuments
    SET Content.modify('replace value of {safeXPath} with ("{safeNewValue}")')
    OUTPUT INSERTED.*
    WHERE Id = @id

-- dodawanie atrybutu
UPDATE XmlDocuments
    SET Content.modify('insert attribute {safeAttr} "{safeValue}" into {safeXPath}')
    OUTPUT INSERTED.*
    WHERE Id = @id

-- Usuwanie węzła
UPDATE XmlDocuments
    SET Content.modify('delete {safeXPath}/@{safeAttr}')
    OUTPUT INSERTED.*
    WHERE Id = @id
```