
Laboratorium 6 – Projektowanie języków XML

XML Schema – restrykcje, indykatory

Co poznamy na zajęciach i do czego się to przyda?

Kontynuujemy poznawanie XML Schema i na dzisiejszych laboratoriach zaznajomimy się z restrykcjami i indykatorami, które pozwalają na szerokie definiowanie własnych typów.

Zadanie 1

Zapoznaj się z materiałami na stronie W3C, zwłaszcza na podstronach:

- http://www.w3schools.com/xml/schema_facets.asp (Restrykcje)
- http://www.w3schools.com/xml/schema_dtypes_string.asp (Stringi)
- http://www.w3schools.com/xml/schema_dtypes_date.asp (Daty)
- http://www.w3schools.com/xml/schema_dtypes_numeric.asp (liczby)

Spróbuj znaleźć i inne ciekawe tutoriale.

Pobierz plik czekolada.xml (<https://inf.ug.edu.pl/~gmadejsk/xml/docs/czekolada.xml>) zawierający małą bazę danych wyrobów czekoladowych znajdujących się w magazynie pewnego sklepu. Zapoznaj się z jego treścią.

Stwórz plik czekolada.xsd dla powyższego pliku XML używając metody definiowania typów (metoda druga z poprzednich laboratoriów z zadania 2). Użyj odpowiednich typów bazowych i nałóż na nie odpowiednie restrykcje tak, aby:

- element `nazwa` był łańcuchem znaków o długości minimum 3 i maksimum 50
- element `rodzaj` był łańcuchem znaków przyjmującym tylko cztery wybrane wartości: `bombonierka`, `czekolada`, `cukierki`, `pozostałe`
- element `liczba` był liczbą całkowitą z zakresu od 0 do 10000
- element `cena` był liczbą zmiennoprzecinkową z dwoma miejscami po przecinku (czy `cena` ma typ prosty i złożony? Co z atrybutem `jednostki`?
http://www.w3schools.com/xml/schema_complex_text.asp)
- atrybut `jednostki` przyjmował wybrane wartości: `sztuki`, `kg`, `dg`, `g`
- element `kod` był łańcuchem znaków o określonej strukturze, który może składać się tylko z dużych liter, cyfr i myślników, ponadto
 - najpierw są 3 cyfry, potem myślnik, potem jedna litera, potem myślnik, potem dowolnie długi ciąg cyfr i liter poza literami A i B, lub:
 - najpierw są 3 cyfry, potem myślnik, potem dwie litery, potem myślnik, potem ciąg cyfr i liter o długości od 5 do 10
- element `data-spozycia` był datą w formacie YYYY-MM, w której rok to minimum 2016

Zwaliduj czekolada.xml względem czekolada.xsd. Sprawdź czy walidacja przebiegła pomyślnie.

Pobierz i zwaliduj również plik [czekolada2.xml](https://inf.ug.edu.pl/~gmadejsk/xml/docs/czekolada2.xml) (<https://inf.ug.edu.pl/~gmadejsk/xml/docs/czekolada2.xml>), w którym celowo umieszczono kilka błędów. Wyjaśnij jakie to błędy.

Zadanie 2

W zadaniu przećwiczmy korzystanie z indykatorów:

- http://www.w3schools.com/xml/schema_complex_indicators.asp

Pewna firma chce zarządzać kodami rabatowymi do pewnej loterii. Ściągnij plik z kodami [kody.xml](https://inf.ug.edu.pl/~gmadejsk/xml/docs/kody.xml) (<https://inf.ug.edu.pl/~gmadejsk/xml/docs/kody.xml>) i przejrzyj jego treść.

Napisz schemat [kody.xsd](#), który będzie walidował [kody.xml](#).

Wymagania:

- element-korzeń `kody` będzie zawierał dowolną (od zera do nieskończoności) liczbę elementów `grupa1`, po której znajduje się sekwencja dowolnej liczby elementów `grupa2`, a następnie w podobny sposób pojawiają się wielokrotnie `grupa3`, `grupa4`, `grupa5`.
- Każdy element `grupa1` zawiera tylko jeden kod: albo `kod1`, albo `kod2`, albo `kod3`. Użyj indykatora `choice`.
- Każdy element `grupa2` zawiera `kod1`, `kod2`, `kod3` (łącznie trzy kody), ale w dowolnej kolejności. Użyj indykatora `all`.
- Każdy element `grupa3` zawiera `kod1`, za którym znajduje się albo `kod2` albo `kod3`. Jakich indykatorów użyć?
- Każdy element `grupa4` zawiera najpierw sekwencję kodów `kod1`, przy czym jest ich minimalnie 2. Następnie pojawia się sekwencja kodów `kod2`, ale może być ich maksymalnie 4.
- Każdy element `grupa5` zawiera dowolną liczbę kodów `kod1`, `kod2`, `kod3` ustawionych w dowolnej kolejności. Jak sprytnie połączyć użycie indykatorów, żeby to było możliwe?
- Element `kod1`, `kod2`, `kod3` to proste elementy zawierające stringi.

Zadanie 3

Na zakończenie przećwiczmy jeszcze restrykcje oparte o wyrażenia regularne. Ściągnij plik xml i schemat do niego:

- <https://inf.ug.edu.pl/~gmadejsk/xml/docs/wzory.xsd>
- <https://inf.ug.edu.pl/~gmadejsk/xml/docs/wzory.xml>

a następnie uzupełnij schemat odpowiednimi wzorami pattern (regex).

Polecenia i wskazówki do rozwiązania (link) znajdują się w pliku xsd. Zauważ, że w komentarzu znajduje się polecenie, linki do bardzo przydatnych tutoriali (należy przejrzeć) i wykomentowany `<xsd:patter>`, który należy odkomentować i uzupełnić.

Należy starać się użyć technik zastosowanych w samouczkach z linków.

Po wstawieniu wyrażeń regularnych zwaliduj plik xml względem schematu i sprawdź, które z danych są poprawne, a które nie. W pliku XML umieściłem trochę błędnych danych, więc na pewno pojawią się błędy. Błędne linijki w XML można zaznaczyć umieszczając w nich komentarz z wyjaśnieniem błędu. Pliku XML nie trzeba poprawiać.

Sprawdzenie

Po wykonaniu zadań poproś prowadzącego o sprawdzenie plików:

- czekolada.xsd z pierwszego zadania. Należy zwalidować względem dobrego i błędnego pliku xml, wyjaśnić błędy.
- kody.xsd z drugiego zadania. Należy przedstawić schemat i zastosowane w nim rozwiązania, zwalidować kody.xml.
- wzory.xsd – uzupełniony, wzory.xml z komentarzami – objaśnij wprowadzone wzory.

Podsumowanie

Na dzisiejszy zajęciach rozszerzyliśmy wiedzę na temat schematów XML. Umiemy już definiować własne typy za pomocą wyliczeń, wyrażeń regularnych i innych restrykcji. To sprawia, że mamy już dużą kontrolę nad zawartością pliku XML.

Na kolejnych zajęciach...

Na kolejnych zajęciach kontynuujemy temat XSD.