29. Poprawność składniowa i strukturalna dokumentu XML

Dawid Gałecki

13 października 2015

Poprawność składniowa

Poprawność składniowa

Poprawny składniowo dokument XML powinien być tworzony zgodnie z poniżej przedstawionymi zasadami.

Poprawność składniowa

Deklaracja XML

Powinien zawierać deklarację XML, która musi być umieszczona na samym początku pliku (nie może być poprzedzona np. komentarzem) oraz musi posiadać atrybut version (dopuszczalne wartości to 1.0 albo 1.1) oraz opcjonalnie atrybuty:

- Encoding – deklaruje zestaw znaków używanych w dokumencie XML, wartością domyślną jest kodowanie UTF-8 w systemie Unicode.
- Standalone – określa tryb dokumentu XML, może przyjmować wartość **yes** lub **no**. Jeśli ustawimy wartość na **yes** będzie to oznaczało, że dokument nie zawiera innych plików, które muszą zostać przetworzone wraz z nim samym. Może to być np. zewnętrzny arkusz stylów lub definicja DTD.

Poprawność składniowa

Jeden element główny

Musi zawierać dokładnie jeden element główny.

Znaczniki początkowy i końcowy

Każdy element musi zaczynać się znacznikiem początku elementu, np. `<imie>`, oraz kończyć identycznym znacznikiem końca elementu, np. `</imie>`. Wyjątek stanowią elementy puste (np. `<element-pusty />`), czyli takie, które nie zawierają żadnych danych ani innych elementów, ale mogą zawierać atrybuty.

Poprawność składniowa

Jeden element główny

Musi zawierać dokładnie jeden element główny.

Znaczniki początkowy i końcowy

Każdy element musi zaczynać się znacznikiem początku elementu, np. `<imie>`, oraz kończyć identycznym znacznikiem końca elementu, np. `</imie>`. Wyjątek stanowią elementy puste (np. `<element-pusty />`), czyli takie, które nie zawierają żadnych danych ani innych elementów, ale mogą zawierać atrybuty.

Poprawność składniowa

Nazwy elementów

Nazwy elementów **mogą** zawierać znaki alfanumeryczne (litery a-z, A-Z oraz cyfry 0-9) oraz znaki interpunkcyjne: podkreślenie, myślnik i kropkę. **Nie mogą** natomiast zaczynać się od myślnika, kropki ani cyfry.

Poprawność składniowa

Zagnieżdżanie elementów

Elementy można zagnieżdżać w sobie i wtedy każdy element znajdujący się wewnątrz innego elementu jest nazywany „dzieckiem" tego elementu, a element, wewnątrz którego znajdują się inne elementy, zwany jest „rodzicem" tych elementów. Nie można stosować konstrukcji typu `<news><tresc> ... </news></tresc>`, ponieważ element `<tresc>` nie jest prawidłowo zagnieżdżony w elemencie `<news>`.

Atrybuty

Każdy element może zawierać atrybuty, które definiuje się w znaczniku początkowym elementu. `<usmiech szczery="tak">` – tu atrybutem elementu `usmiech` jest atrybut o nazwie `szczery` oraz wartości `tak`. Wartości atrybutów podaje się w cudzysłowach.

Poprawność składniowa

Zagnieżdżanie elementów

Elementy można zagnieżdżać w sobie i wtedy każdy element znajdujący się wewnątrz innego elementu jest nazywany „dzieckiem" tego elementu, a element, wewnątrz którego znajdują się inne elementy, zwany jest „rodzicem" tych elementów. Nie można stosować konstrukcji typu `<news><tresc> ... </news></tresc>`, ponieważ element `<tresc>` nie jest prawidłowo zagnieżdżony w elemencie `<news>`.

Atrybuty

Każdy element może zawierać atrybuty, które definiuje się w znaczniku początkowym elementu. `<usmiech szczery="tak">` – tu atrybutem elementu `usmiech` jest atrybut o nazwie `szczery` oraz wartości `tak`. Wartości atrybutów podaje się w cudzysłowach.

Poprawność składniowa

Zabronione znaki

W danych, atrybutach oraz nazwach elementów nie mogą pojawiać się znaki takie jak < albo & ponieważ parsery XML „widząc” np. znak mniejszości wewnątrz elementu stwierdzą, że jest to początek znacznika i dokument zostanie błędnie zinterpretowany.
Specyfikacja XML daje jednak możliwość używania takich znaków – jeśli chcemy wstawić znak < wpisujemy zamiast niego sekwencję `<`, a gdy chcemy wprowadzić znak & wpisujemy `&`.

Dane zawierające kod HTML

Część danych, które zawierają np. kod HTML lub XML, możemy zapisać w sekcji danych znakowych, która nie będzie przetwarzana przez analizator składni XML. Znacznik początku sekcji danych znakowych to `<![CDATA[`, a znacznik końca to `]]>`.

Poprawność składniowa

Zabronione znaki

W danych, atrybutach oraz nazwach elementów nie mogą pojawiać się znaki takie jak < albo & ponieważ parsery XML „widząc” np. znak mniejszości wewnątrz elementu stwierdzą, że jest to początek znacznika i dokument zostanie błędnie zinterpretowany.
Specyfikacja XML daje jednak możliwość używania takich znaków – jeśli chcemy wstawić znak < wpisujemy zamiast niego sekwencję `<`, a gdy chcemy wprowadzić znak & wpisujemy `&`.

Dane zawierające kod HTML

Część danych, które zawierają np. kod HTML lub XML, możemy zapisać w sekcji danych znakowych, która nie będzie przetwarzana przez analizator składni XML. Znacznik początku sekcji danych znakowych to `<![CDATA[`, a znacznik końca to `]]>`.

Poprawność składniowa

Komentarze

W dokumencie XML możemy wykorzystywać komentarze, które zaczynają się znakami <!-- a kończą znakami --> – dokładnie tak jak w HTML-u.

Instrukcje przetwarzania

Specyfikacja XML zezwala na wstawianie instrukcji przetwarzania, które są wykorzystywane do przeniesienia informacji do aplikacji. Instrukcje przetwarzania rozpoczynają się znakami <?, a kończą znakami ?>. Przykładem takiej instrukcji może być odniesienie do arkusza stylów, który jest powiązany z dokumentem XML: `<?xml-stylesheet type="text/css" href="style.css"?>`.

Poprawność składniowa

Komentarze

W dokumencie XML możemy wykorzystywać komentarze, które zaczynają się znakami <!-- a kończą znakami --> – dokładnie tak jak w HTML-u.

Instrukcje przetwarzania

Specyfikacja XML zezwala na wstawianie instrukcji przetwarzania, które są wykorzystywane do przeniesienia informacji do aplikacji. Instrukcje przetwarzania rozpoczynają się znakami <?, a kończą znakami ?>. Przykładem takiej instrukcji może być odniesienie do arkusza stylów, który jest powiązany z dokumentem XML: `<?xml-stylesheet type="text/css" href="style.css"?>`.

Poprawność strukturalna

Poprawność strukturalna

Poprawność strukturalna XML – poprawność konstrukcji dokumentu XML z punktu widzenia jego zgodności ze zdefiniowanym w DTD językiem. Porównanie dokumentu z językiem jest określane mianem walidacji i jest znacznie bardziej skomplikowanym procesem niż badanie poprawności składniowej. Mówimy że dokument jest poprawny strukturalnie jeżeli jest zgodny z definicją dokumentu, tzn. dodatkowymi regułami określonymi przez użytkownika. Do precyzowania tych reguł służą specjalne języki, np. bardzo popularny DTD.

Poprawność strukturalna

Poprawność strukturalna – przykład użycia DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE osoba [
   <!ELEMENT osoba (imie, nazwisko, zawod)>
   <!ELEMENT imie (#PCDATA)>
   <!ELEMENT nazwisko (#PCDATA)>
   <!ELEMENT zawod (#PCDATA)>
]>
<osoba>
   <imie>Robert</imie>
   <nazwisko>Lewandowski</nazwisko>
   <zawod>Piłkarz</zawod>
</osoba>
```