

## 02. Import metadanych z formatu XML



### Uwaga

Biblioteka programistyczna dla tego rozszerzenia to `dlibra-app-extension-mf-xml`.

Funkcjonalność aplikacji redaktora i administratora można rozszerzać przy pomocy tzw. rozszerzeń. Między innymi można dodawać rozszerzenia umożliwiające import metadanych do systemu dLibra z zewnętrznych źródeł. Takim właśnie rozszerzeniem jest rozszerzenie umożliwiające importowanie opisu bibliograficznego z pliku zapisanego w formacie XML (informacje na temat formatu XML można znaleźć [tutaj](#)).

Rozszerzenie umożliwiające import metadanych z formatu XML bazuje na zapytaniach XQuery (informacje na temat XQuery można znaleźć [tutaj](#)). Poniższy opis zakłada, że czytelnik jest zaznajomiony z standardem XQuery.

Aby była możliwość importowania opisu bibliograficznego, rozszerzenie XML musi być odpowiednio skonfigurowane. Domyślna konfiguracja tego rozszerzenia pozwala na importowanie opisu bibliograficznego z formatu RDF oraz z formatu MASTER.

Rozszerzenie XML konfiguruje się przy użyciu dwóch plików właściwości (pliki właściwości zawierają pary klucz=wartość):

- `tests.properties` - plik zawierający zapytania XQuery testujące rodzaj pliku z którego będzie importowany opis bibliograficzny,
- `conversion.properties` - plik zawiera reguły konwersji pliku XML do metadanych w systemie dLibra.

Pliki te należy utworzyć samodzielnie na dysku lokalnym redaktora lub udostępnić przez www, a następnie wskazać ich lokalizację w konfiguracji rozszerzenia do importu XML.

Powiązanie pomiędzy tymi plikami jest bardzo ściśle - dla każdego testu z pliku `tests.properties` istnieją reguły konwersji w pliku `conversion.properties`. Dla pliku XML z metadanymi mechanizm importu wykonuje po kolei zapytania testowe z pliku `tests.properties`. Jeśli natrafi na zapytanie, które zwraca więcej niż 0 wartości to importuje metadane z pliku XML przy użyciu reguł konwersji (z pliku `conversion.properties`) odpowiadających testowi.

W pliku `tests.properties` zamieszczone są zapytania XQuery, które sprawdzają czy dany plik z metadanymi można importować przy użyciu reguł konwersji skojarzonych z tym zapytaniem. Klucz identyfikuje reguły konwersji w pliku `conversion.properties`.

Przykładowo założmy, że mamy następujące pliki (przykład ten prezentuje domyślną konfigurację rozszerzenia):

plik `tests.properties`:

```
master=for $x in fn:doc({document})*[fn:compare(fn:name(), 'msDescription')=0] return $x
rdf_dc=for $x in fn:doc({document})*[fn:compare(fn:local-name(), 'RDF')=0] return $x
```

plik `conversion.properties`:

```
master.Title=for $x in fn:doc({document})//msHeading/title return $x
master.Creator=for $x in fn:doc({document})//msHeading/author return $x
master.Description=for $x in fn:doc({document})//msContents/overview return $x
master.Publisher=for $x in fn:doc({document})//msContents/respStmt/resp/name return $x
master.Contributor=for $x in fn:doc({document})//msDescription/msContents/respStmt//resp return $x
master.Date=for $x in fn:doc({document})//msHeading/origDate return $x
master.Type=for $x in fn:doc({document})//physDesc/form return $x
master.Identifier=for $x in fn:doc({document})//msIdentifier/country/settlement/repository/idno return $x
master.Source=for $x in fn:doc({document})//msPart//idno return $x
master.Language=for $x in fn:doc({document})//msContents/textLang return $x, for $x in fn:doc({document})
//msContents/textLang/@otherLangs return $x
master.Rights=for $x in fn:doc({document})//msIdentifier/repository return $x

rdf_dc=for $x in fn:doc({document})*[fn:local-name()='Description']/* return $x
```

Jak widać, plik `conversion.properties` zawiera reguły konwersji odpowiadające testom z pliku `tests.properties`. Reguły w pliku `conversion.properties` mogą być dwójakiego rodzaju:

1. Reguła dla konkretnego atrybutu - klucz reguły składa się z klucza identyfikującego test z pliku `tests.properties`, kropki, oraz identyfikatora atrybutu (nazwy RDF) w systemie dLibra. W takim przypadku wartości wszystkich węzłów zwróconych przez zapytanie XQuery zostaną przypisane do atrybutu o tym identyfikatorze.
2. Reguła uniwersalna - klucz reguły jest taki sam jak klucz identyfikujący test z pliku `tests.properties`. W takim przypadku każdy węzeł zwrócony przez zapytanie XQuery musi mieć nazwę taką samą jak nazwa RDF docelowego atrybutu. Przestrzenie nazw xmlns są ignorowane.

W przykładzie powyżej, konfiguracja dla formatu MASTER została zdefiniowana przy pomocy reguł pierwszego rodzaju, a dla formatu RDF - drugiego rodzaju. Nie ma przeszkód, aby w definicji jednego formatu wykorzystać oba rodzaje reguł jednocześnie - w takim przypadku, jeśli reguła uniwersalna wygeneruje wartości dla atrybutu, który ma również przypisaną dedykowaną regułę, to zostaną przypisane wartości uzyskane z obu reguł.

Dla danego formatu XML można zdefiniować tylko jedną regułę uniwersalną, jak i dla każdego konkretnego atrybutu można przypisać tylko jedną dedykowaną regułę. Reguła XQuery może jednak składać się z kilku reguł połączonych przecinkami, tak jak ma to miejsce w przypadku atrybutu `Language`.

Każde zapytanie XQuery powinno wykorzystywać ciąg znaków `{document}` do określenia dokumentu na którym wykonywane jest zapytanie. Rozszerzenie zamienia ten ciąg znaków ścieżką do pliku XML.

Założmy, że chcemy zaimportować plik A zawierający metadane w formacie XML. Mechanizm importu sprawdza kolejno testy znajdujące się w pliku `tests.properties`. Pierwszy test, który zwróci w wyniku listę wartości o rozmiarze większym od 0 decyduje o regułach konwersji. Założmy że był to test opatrzony kluczem master. Mechanizm importu wybiera reguły konwersji z pliku `conversion.properties` - wszystkie te które rozpoczynają się od słowa master. Następnie wartości z zapytań XQuery trafiają do odpowiedniego atrybutu, np. do atrybutu, którego nazwa RDF to `Title` trafiają wszystkie wartości z zapytania `for $x in fn:doc({document})//msHeading/title return $x`.