# XPath – Zadania

## Przykłady wykorzystania XPath

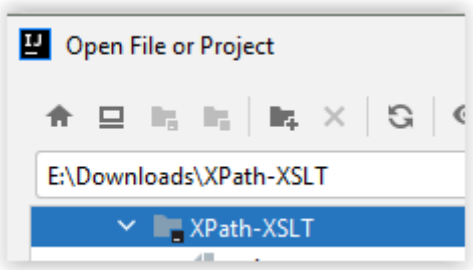
Na początku przyjrzymy się przykładom wykorzystania wyrażeń XPath w dwóch różnych standardach:

- w transformacjach XSLT
- w zapytaniach XQuery.

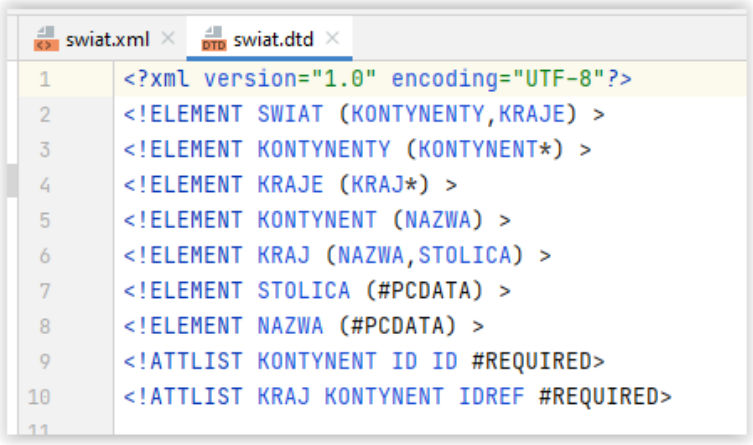
Oba standardy wykorzystywane są do przetwarzania dokumentów XML.

### XPath w XSLT

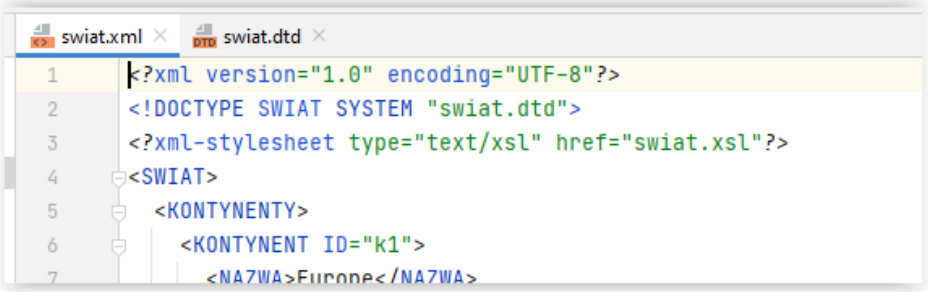
1. Ze strony kursu pobierz plik `XPath-XSLT.zip`. a następnie go rozpakuj.
2. Otwórz *IntelliJ IDEA*, a następnie otwórz katalog `XPath-XSLT`




3. Przyglądnij się budowie pliku `swiat.xml` zawartej w jego definicji DTD. W tym celu otwórz plik `swiat.dtd`

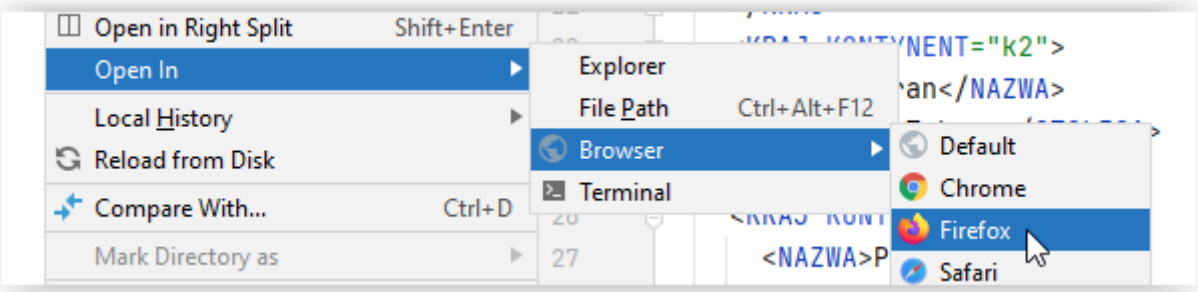


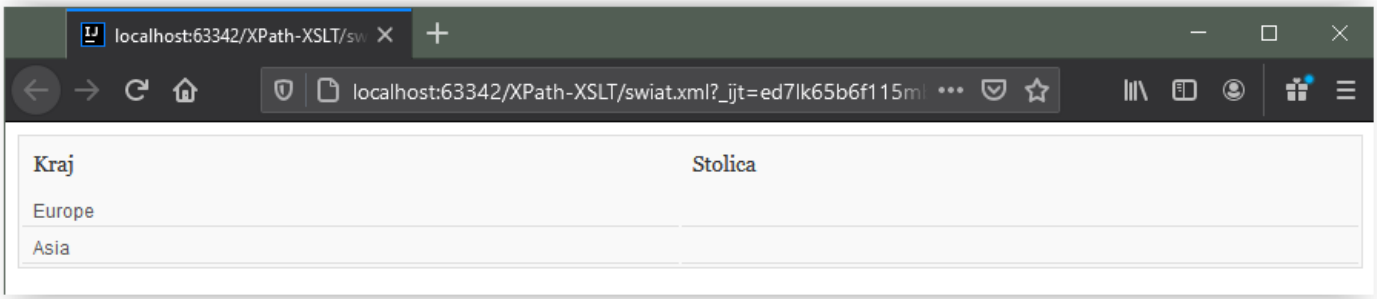

4. Otwórz plik `swiat.xml`. Na początku pliku (poniżej nagłówka) znajdziesz odwołanie do jego definicji, a następnie instrukcję przetwarzania, pozwalającą na jego transformację arkuszem stylów `swiat.xsl`.




5. Za pomocą menu kontekstowego otwórz plik `swiat.xml` za pomocą przeglądarki. **Niektóre** przeglądarki obsługują instrukcję przetwarzania dla aplikacji `xml-stylesheet` i działają jako procesory XSLT. W takich

przypadkach automatycznie podejmują transformację i prezentują jej wynik zamiast prezentacji źródłowego pliku XML.




6. Wspomniana transformacja tworzy na podstawie pliku `swiat.xml` dokument XHTML zawierający tabelę. Jak widać tabela ta zawiera dość dziwne dane – w kolumnie kraj zawarte są nazwy kontynentów. Otwórz plik `swiat.xsl` aby znaleźć przyczynę.

7. Poniższy fragment jest głównym „winowajcą”.
   *Nie przejmuj się jeśli nie rozumiesz konstrukcji zawartych w arkuszu stylów – poznasz je wkrótce. One w tej chwili nie są ważne. Na razie koncentruj się na wyrażeniach XPath.*

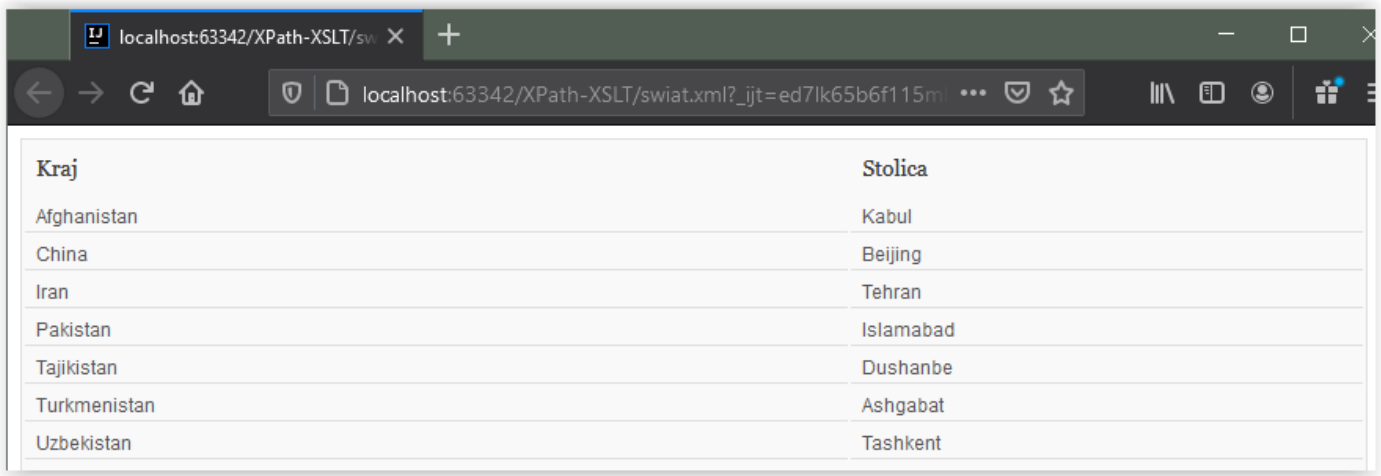
```
9   <body>
10    <table>
11      <tr><th>Kraj</th><th>Stolica</th></tr>
12      <xsl:apply-templates select="SWIAT/KONTYNENTY/KONTYNENT"/>
13    </table>
14  </body>
```

8. Jak widać, po wygenerowaniu nagłówka tabeli, arkusz stylów przetwarza elementy `KONTYNENT`. Wskazuje je za pomocą ścieżki względnej `SWIAT/KONTYNENTY/KONTYNENT`. Kontekstem tego wyrażenia ścieżkowego jest korzeń dokumentu, co można zaobserwować w elemencie `xsl:template`, w którym znajduje się nasz „winowajca”.

```
3     version="1.0">
4     <xsl:template match='/'>
5       <html>
```

9. Zamiarem twórcy arkusza stylów było przetwarzanie nie kontynentów a krajów, które w dokumencie `swiat.xml` także się znajdują. Popraw wyrażenie ścieżkowe `SWIAT/KONTYNENTY/KONTYNENT` na wyrażenie, które w kontekście korzenia dokumentu będzie wskazywało elementy `KRAJ`. Nie zapominaj o zapisywaniu na dysku zmodyfikowanego arkusza stylów.

10. Odśwież w przeglądarce plik `swiat.xml`. Powinieneś otrzymać rezultat jak poniżej.




| Kraj | Stolica |
|---|---|
| Afghanistan | Kabul |
| China | Beijing |
| Iran | Tehran |
| Pakistan | Islamabad |
| Tajikistan | Dushanbe |
| Turkmenistan | Ashgabat |
| Uzbekistan | Tashkent |

11. Wyświetlona lista krajów obejmuje kraje z dwóch kontynentów, ogranicz ją do krajów z Europy. Dokonaj w tym celu stosownej modyfikacji wcześniej zmienianego wyrażenia ścieżkowego.
Na początku ogranicz listę do tych elementów `KRAJ`, które posiadają atrybut `@KONTYNENT` ze stosowną wartością (`k1`).

| Kraj | Stolica |
|---|---|
| Albania | Tirane |
| Greece | Athens |
| Macedonia | Skopje |
| Serbia and Montenegro | Belgrade |

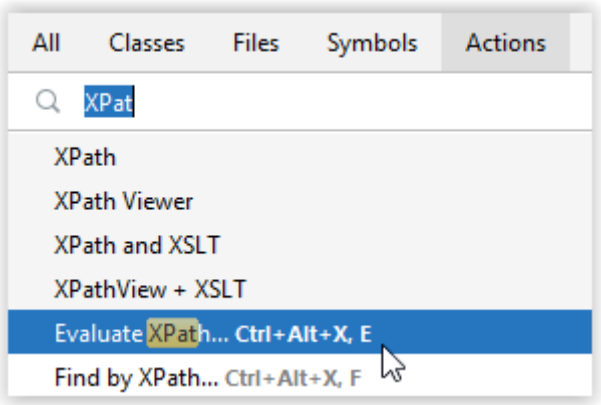
. . .

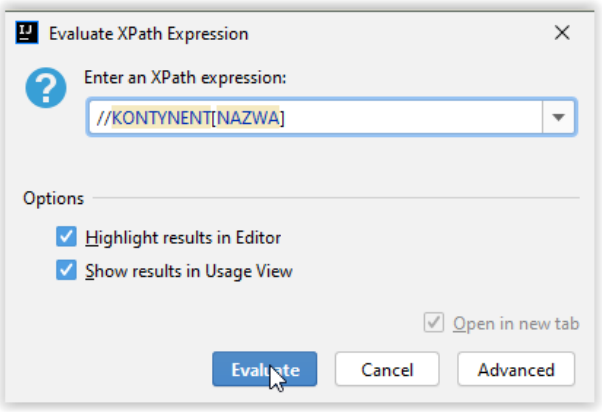
12. Jeśli to ci się udało, to spróbuj uzyskać to samo, ale zamiast wartości `k1` użyj wyrażenia ścieżkowego znajdującego atrybut `ID` kontynentu o nazwie `Europe`.
*Wskazówka: Wyrażenie ścieżkowe wskazujące `ID` kontynentu o nazwie `Europe` powinno być bezwzględne.*

| Kraj | Stolica |
|---|---|
| Albania | Tirane |
| Greece | Athens |
| Macedonia | Skopje |
| Serbia and Montenegro | Belgrade |

. . .

W przypadku problemów możesz przetestować swoje wyrażenia XPath korzystając narzędzia dostarczanego przez IntelliJ. Za pomocą kombinacji klawiszy *Ctrl+Shift+A* wyszukaj to narzędzie a następnie je uruchom.

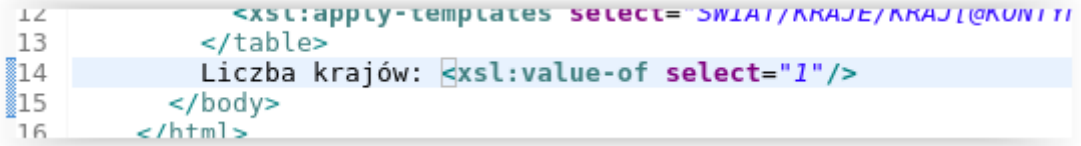



13. Uzupełnimy teraz wynikowy dokument naszej transformacji o dodatkowe informacje.
W tym celu umieść w arkuszu stylów **poniżej elementu `table`** następujący fragment

```
Liczba krajów: <xsl:value-of select="1"/>
```

Powinno to wyglądać następująco:




14. Sprawdź efekt tej zmiany.

| Portugal | Lisbon |
|---|---|
| United Kingdom | London |

Liczba krajów: 1

15. Zmień statyczną wartość 1 na wyrażenie XPath, które zliczy kraje znajdujące się w Europie – te same kraje, które są wyświetlane w ramach listy – wyrażenie XPath wskazujące te kraje będzie identyczne.

| | |
|---|---|
| Portugal | Lisbon |
| United Kingdom | London |

Liczba krajów: 34

16. Pamiętasz funkcję `position()` dostępną wśród wielu funkcji XPath? Wykorzystamy ją do ponumerowania wierszy tabeli. Wstaw fragment

```
<td><xsl:value-of select="1"/></td>
```

jako pierwszą kolumnę generowaną w drugim szablonie (elemencie `template`). Po zmianie szablon powinien wyglądać następująco:

```
17    </xsl:template>
18    <xsl:template match="*">
19      <tr>
20        <td><xsl:value-of select="1"/></td>
21        <td><xsl:value-of select="NAZWA"/></td>
22        <td><xsl:value-of select="STOLICA"/></td>
23      </tr>
24    </xsl:template>
```

17. Zmień wartość 1 na wywołanie wspomnianej funkcji.

18. Popraw jeszcze kolumny generowane w nagłówku tabeli, dodając jako pierwszą kolumnę `lp`

```
9       <body>
10        <table>
11          <tr><th>lp</th><th>Kraj</th><th>S
12          <xsl:apply-templates select="SWIA
13        </table>
```

19. Sprawdź efekt wykonanych zmian.

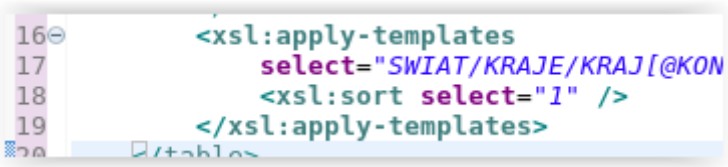
| lp | Kraj | Stolica |
|---|---|---|
| 1 | Albania | Tirane |
| 2 | Greece | Athens |
| 3 | Macedonia | Skopje |
| 4 | Serbia and Montenegro | Belgrade |

. . .

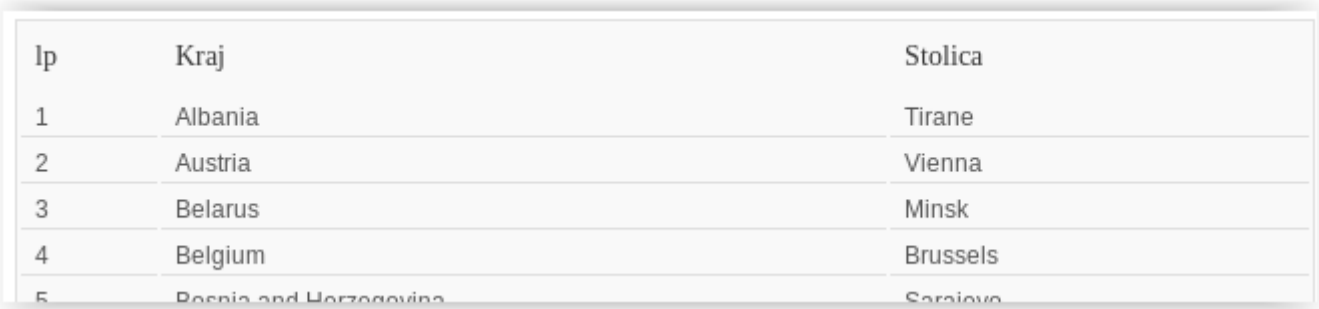
20. Na zakończenie posortujemy wynikową tabelę. W tym celu „otwórz” element `apply-templates`, a następnie wpisz do jego wnętrza następujący fragment

```
<xsl:sort select="1"/>
```

Powinno to wyglądać następująco:




21. Atrybut `select` za pomocą wyrażenia ścieżkowego pozwala na wskazanie węzła definiującego porządek przetwarzania węzłów. Umieść we wnętrzu atrybutu wyrażenie ścieżkowe, które w kontekście elementu `KRAJ` wskazywało będzie element zawierający jego nazwę.
Sprawdź uzyskany efekt.



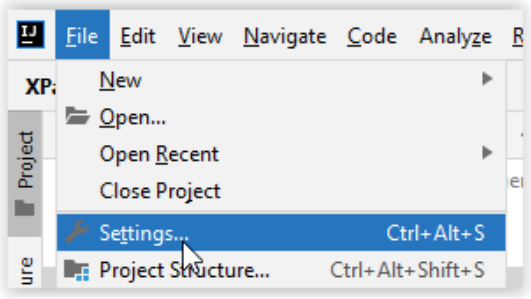
| lp | Kraj | Stolica |
|---|---|---|
| 1 | Albania | Tirane |
| 2 | Austria | Vienna |
| 3 | Belarus | Minsk |
| 4 | Belgium | Brussels |
| 5 | Bosnia and Herzegovina | Sarajevo |

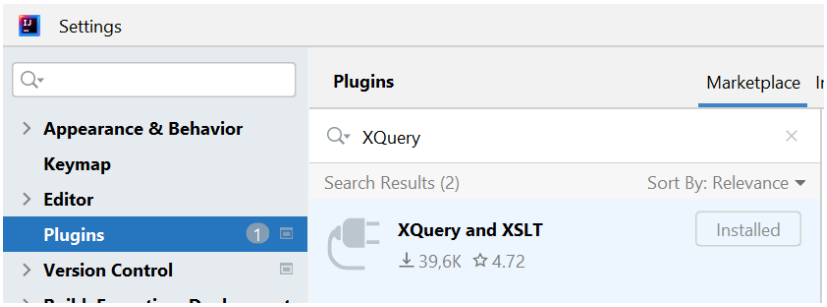
. . .

## XPath w XQuery

Język XQuery jest obsługiwany przede wszystkim w XML-owych bazach danych np. eXist lub BaseX. Jednakże, po doinstalowaniu stosownej wtyczki, do ewaluacji wyrażeń XQuery można wykorzystać również IntelliJ IDEA.

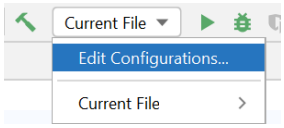
22. Wywołaj ustawienia, a następnie, sprawdź i ewentualnie doinstaluj wtyczkę XQuery and XSLT. W razie potrzeby zrestartuj środowisko IntelliJ.



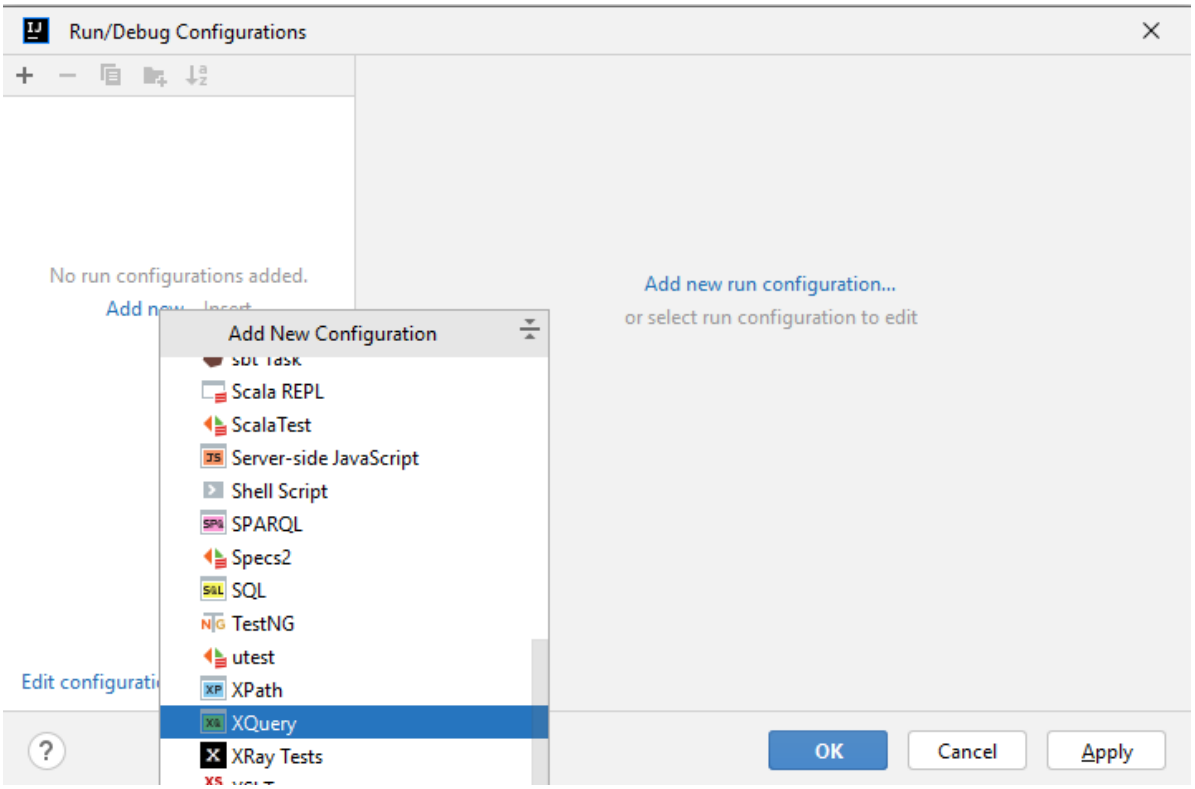

23. Zainstalowana wtyczka do swojego działania wymaga procesora XQuery. Może w tym zakresie wykorzystać dostępną instancję serwera XML-owej bazy danych lub implementację procesora w formie biblioteki.
W dalszej części zadań skorzystamy z tej drugiej konfiguracji. Pobierz bibliotekę Saxon w wersji HE (darmowa Home Edition) ze strony https://www.saxonica.com/download/java.xml. Rozpakuj pobrane archiwum w wybranym przez siebie katalogu.

24. Z zaznaczonym w panelu projektów projektem `XPath-XQuery` wywołaj edytor konfiguracji.

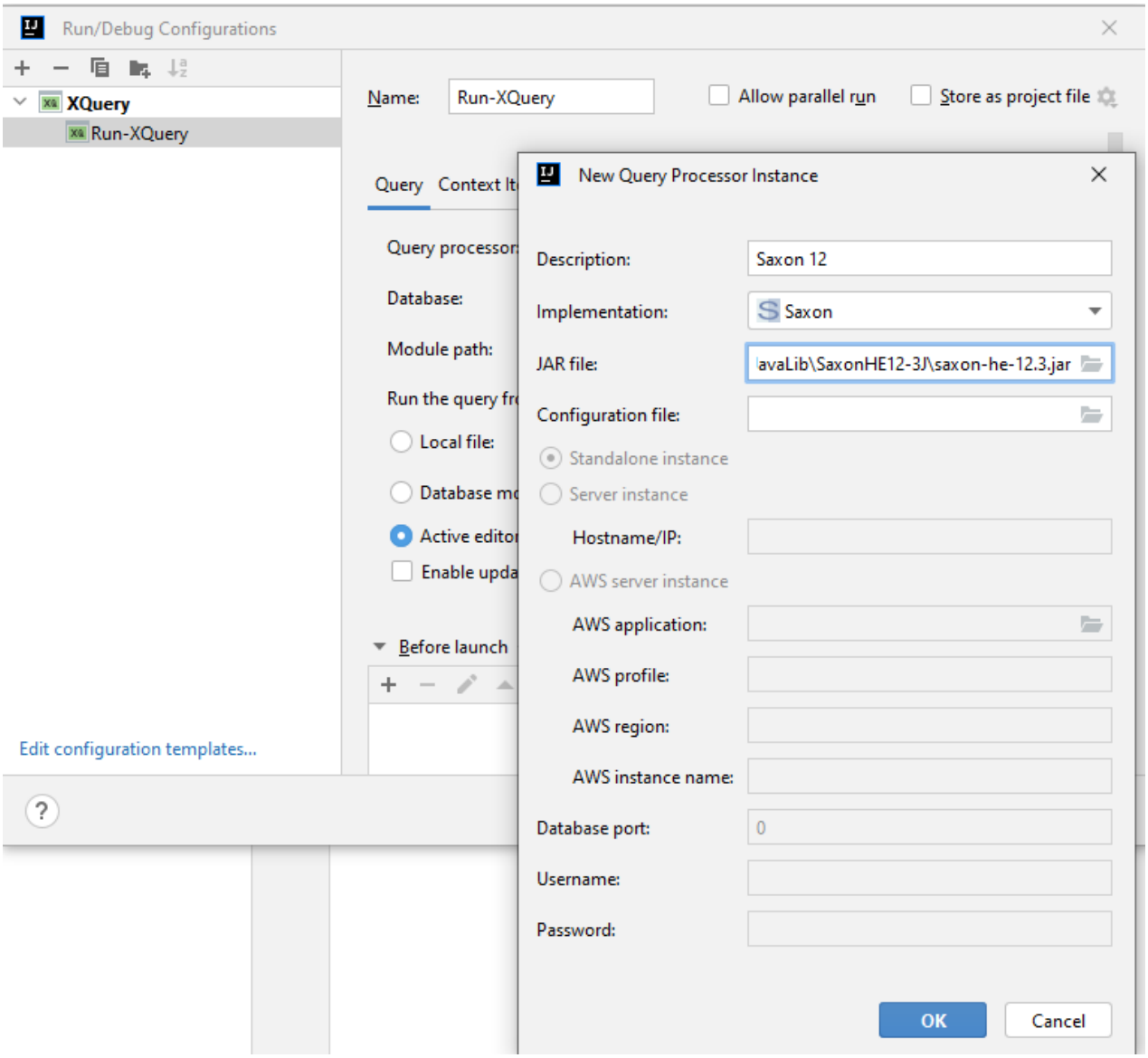



25. Dodaj nową konfigurację klikając Add new…

    a. Jako typ konfiguracji wybierz z listy XQuery.



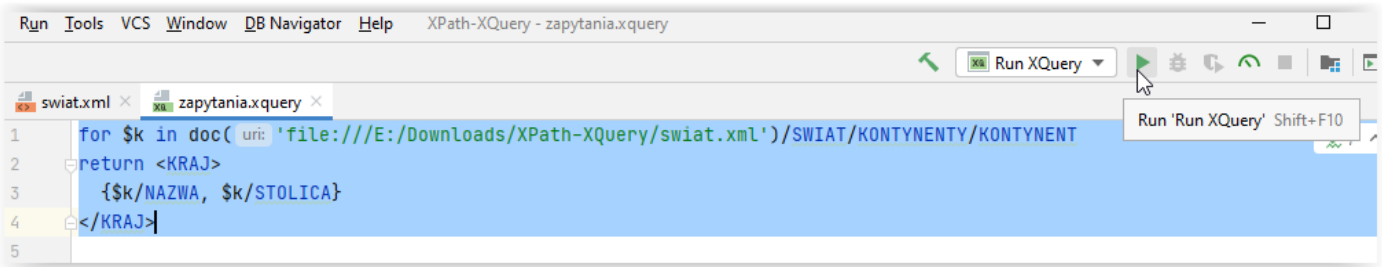

    b. Jako nazwę konfiguracji podaj `Run-XQuery.`
    c. Wskaż, że zapytania mają być brane z bieżącego pliku otwartego w edytorze.
    d. Kliknij przycisk z wielokropkiem przy liście procesorów zapytań. W otwartym okienku do zarządzania procesorami zapytań kliknij plus aby dodać nowy procesor.
    e. Jako nazwę procesora podaj `Saxon 12`. Z listy implementacji wybierz `Saxon`. W polu JAR file wskaż archiwum JAR saxon-he-*.*.jar z lokalizacji, w której została rozpakowana biblioteka. Kliknij OK w każdym z kaskady otwartych okienek dialogowych.



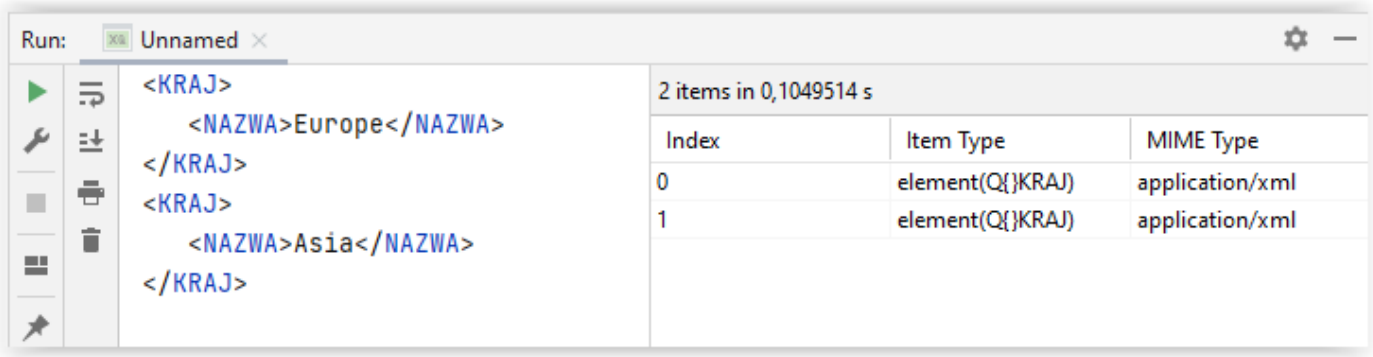

26. Utwórz nowy plik o nazwie `zapytania.xquery`. Wprowadź do niego poniższą zawartość. Popraw ścieżkę do pliku `swiat.xml`

```
for $k in doc('file:///E:/XPath-XQuery/swiat.xml')/SWIAT/KONTYNENTY/KONTYNENT
return <KRAJ>
  {$k/NAZWA, $k/STOLICA}
</KRAJ>
```

Uruchom zapytanie.




Wynik powinien pojawić się w nowej zakładce. Proste?




*Nie przejmuj się jeśli nie rozumiesz konstrukcji użytych w poleceniu XQuery – poznasz je wkrótce – skup się na wyrażeniach XPath.* W składni zapytań XQuery ***modyfikuj tylko zagnieżdżone wyrażenie XPath.***

27. Podobnie jak miało to miejsce w przypadku transformacji arkuszem stylów, popraw wyrażenie ścieżkowe, tak aby odwoływało się do elementów `KRAJ`.

```
<KRAJ>
    <NAZWA>Afghanistan</NAZWA>
    <STOLICA>Kabul</STOLICA>
</KRAJ>
<KRAJ>
    <NAZWA>China</NAZWA>
    <STOLICA>Beijing</STOLICA>
</KRAJ>
<KRAJ>
    <NAZWA>Iran</NAZWA>
    <STOLICA>Tehran</STOLICA>
</KRAJ>
```

28. Ogranicz listę krajów do tych, których NAZWA zaczyna się od litery A. Skorzystaj z funkcji `starts-with(string1,string2)`.

```
<KRAJ>
    <NAZWA>Afghanistan</NAZWA>
    <STOLICA>Kabul</STOLICA>
</KRAJ>
<KRAJ>
    <NAZWA>Armenia</NAZWA>
    <STOLICA>Yerevan</STOLICA>
</KRAJ>
<KRAJ>
    <NAZWA>Azerbaijan</NAZWA>
    <STOLICA>Baku</STOLICA>
</KRAJ>
```

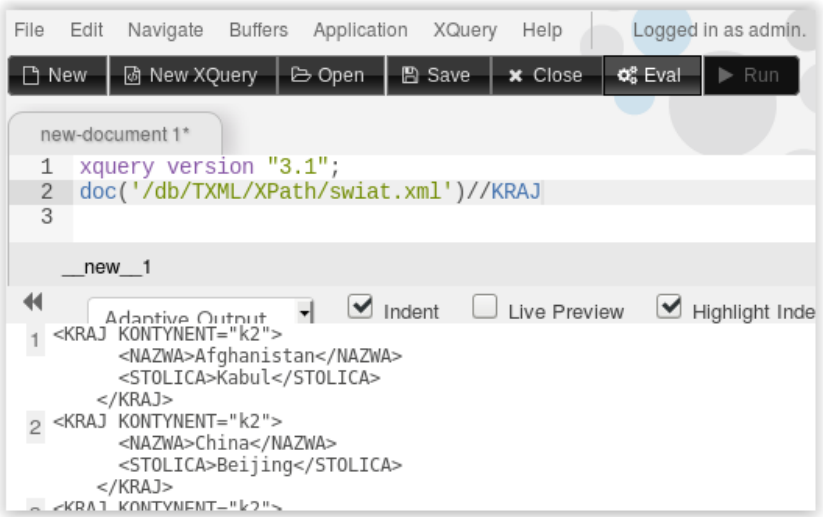
29. Usuń poprzednie ograniczenie i dodaj takie, które pozwoli na pobranie takich krajów, które mają nazwę rozpoczynającą się od tej samej litery do ich stolica. Tym razem skorzystaj z funkcji `substring(string,start,len)`.

```
1 <KRAJ>
      <NAZWA>Belgium</NAZWA>
      <STOLICA>Brussels</STOLICA>
  </KRAJ>
2 <KRAJ>
      <NAZWA>Sweden</NAZWA>
      <STOLICA>Stockholm</STOLICA>
  </KRAJ>
3 <KRAJ>
      <NAZWA>Kuwait</NAZWA>
      <STOLICA>Kuwait</STOLICA>
  </KRAJ>
```

30. Ogranicz postać całego zapytania do następującej postaci

```
doc('popraw_ścieżkę/swiat.xml')//KRAJ
```

Zwróć uwagę, że ono również działa – samodzielne wyrażenia XPath są poprawnymi wyrażeniami XQuery. Wykorzystamy to w następnych zadaniach.




# XPath

Dalsze zadania pozwolą Ci nabrać biegłości w konstruowaniu wyrażeń XPath

## Dokument `zesp_prac.xml`

31. Zmień postać zapytania na następującą:

```
doc('popraw_ścieżkę/zesp_prac.xml')
```

Uruchom go. Zapytanie to daje w wyniku korzeń dokumentu – innymi słowy jest równoznaczne z zapytaniem `/`.

Zapoznaj się z budową dokumentu, a następnie uzupełniając zapytanie o właściwe wyrażenia XPath wykonaj poniższe zadania.

Pamiętaj o tym aby każdą ścieżkę bezwzględną poprzedzić wyrażeniem adresującym korzeń przetwarzanego dokumentu czyli `doc...`

32. Wyświetl elementy `NAZWISKO` pracowników

```
<NAZWISKO>WEGLARZ</NAZWISKO>
<NAZWISKO>MAREK</NAZWISKO>
<NAZWISKO>BRZEZINSKI</NAZWISKO>
```

```
<NAZWISKO>MORZY</NAZWISKO>
<NAZWISKO>KROLIKOWSKI</NAZWISKO>
. . .
<NAZWISKO>BIALY</NAZWISKO>
<NAZWISKO>HAPKE</NAZWISKO>
<NAZWISKO>BLAZEWICZ</NAZWISKO>
```

33. Wyświetl nazwiska pracowników zatrudnionych w zespole SYSTEMY EKSPERCKIE

```
SLOWINSKI
ZAKRZEWICZ
BIALY
HAPKE
```

34. Wyświetl liczbę pracowników zatrudnionych w zespole o ID=10

```
2
```

35. Wyświetl elementy NAZWISKO tych pracowników, których szefem jest pracownik o identyfikatorze 100

```
<NAZWISKO>MAREK</NAZWISKO>
<NAZWISKO>BRZEZINSKI</NAZWISKO>
<NAZWISKO>SLOWINSKI</NAZWISKO>
<NAZWISKO>BLAZEWICZ</NAZWISKO>
```

36. Wyświetl sumę płac podstawowych pracowników zatrudnionych w tym samym zespole co pracownik o nazwisku BRZEZINSKI (wliczając jego pensję)

```
4316.2
```