1. **Otwarcie projektu**

Otwórz projekt, nad którym pracowałeś podczas poprzedniego warsztatu.

1. Otwórz plik **weather-service-impl.xml**
2. Skopiuj zawartość kostki **Set Payload** z getWeather-flow
3. Utwórz nowy szablon getWeatherResponse.tpl

Umieść plik w katalogu src/main/resources/tmpl

1. Usuń komponent **Set Payload** i zastąp go komponentem **Parse Template**
2. Powiąż komponent z plikiem getWeatherResponse.tpl
3. Przetestuj usługę

Uruchom SoapUI. Otwórz projekt Weather. Przejść do GlobalWeatherSoap12 a następnie do GetWeather i Request 1. Kliknij przycisk Submit.

Oczekiwany rezultat:

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

<soap:Body>

<web:GetWeatherResponse xmlns:web="http://www.webserviceX.NET">

<web:GetWeatherResult>{Kraj}:{Miasto}. {data} 16 stopni</web:GetWeatherResult>

</web:GetWeatherResponse>

</soap:Body>

</soap:Envelope>

1. Otwórz plik getWeatherResponse.tpl i uzupełnij placeholdery {xxx} odpowiednimi wyrażeniami MEL

|  |  |
| --- | --- |
| Pole | Wyrażenie |
| Kraj | xpath3('//\*:CountryName') |
| Miasto | xpath3('//\*:CityName') |
| Data | server.dateTime |

1. Przetestuj usługę

Uruchom SoapUI. Otwórz projekt Weather. Przejść do GlobalWeatherSoap12 a następnie do GetWeather i Request 1. Kliknij przycisk Submit.

Oczekiwany rezultat:

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

<soap:Body>

<web:GetWeatherResponse xmlns:web="http://www.webserviceX.NET">

<web:GetWeatherResult>Polska:Warszawa. 2018-07-13T14:46:25.379+02:00 16 stopni</web:GetWeatherResult>

</web:GetWeatherResponse>

</soap:Body>

</soap:Envelope>

Nowa operacja POST /countries/{country/cities zakłada wysłanie nowe miasta i odpowiedź podsumowującą. Zapoznaj się z przykładami requesta i responsa.

1. Utwórz plik **getCitiesByCountry.xsl** w katalogu src/main/resources/xslt

W jego treści wklej:

<xsl:stylesheet version="3.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<web:GetCitiesByCountryResponse xmlns:web="http://www.webserviceX.NET">

<web:GetCitiesByCountryResult>{Kraj}: Warszawa {Data}</web:GetCitiesByCountryResult>

</web:GetCitiesByCountryResponse>

</xsl:template>

</xsl:stylesheet>

1. Usuń komponent **Set Payload** i w jego miejsce umieść **XSLT** w przepływie getCitiesByCountry-flow
2. Przetestuj przepływ

Uruchom SoapUI. Otwórz projekt Weather. Przejść do GlobalWeatherSoap12 a następnie do GetCitiesByCountry i Request 1. Kliknij przycisk Submit.

Oczekiwany rezultat:

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

<soap:Body>

<web:GetCitiesByCountryResponse xmlns:web="http://www.webserviceX.NET">

<web:GetCitiesByCountryResult>{Kraj}: Warszawa {Data}</web:GetCitiesByCountryResult>

</web:GetCitiesByCountryResponse>

</soap:Body>

</soap:Envelope>

1. Dodaj parametr **today** do pliku XSLT i wstaw jego wartość w miejsce {Data}
2. Zmodyfikuj komponent XSLT, ustawiając parametr **today** na datę serwera
3. W miejsce **{Kraj}** umieść wyrażenie XSLT, które wstawi wartość z pola **CountryName**
4. **{Kraj}: Warszawa** powinien wyświetlać się tylko gdy **CountryName** równa się słowu **Polska**. Wprowadź zmianę w pliku XSLT
5. Przetestuj przepływ

Uruchom SoapUI. Otwórz projekt Weather. Przejść do GlobalWeatherSoap12 a następnie do GetCitiesByCountry i Request 1. Kliknij przycisk Submit.

Oczekiwany rezultat:

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">

<soap:Body>

<web:GetCitiesByCountryResponse xmlns:web="http://www.webserviceX.NET">

<web:GetCitiesByCountryResult>Polska: Warszawa 2017-01-01T11:12:33Z</web:GetCitiesByCountryResult>

</web:GetCitiesByCountryResponse>

</soap:Body>

</soap:Envelope>