



Zalecenia w zakresie integracji ze ZSMOPL przy użyciu NET FRAMEWORK

Spis treści

1. Przykładowa konfiguracja połączenia ze ZSMOPL	2
1.1. Środowisko produkcyjne	2
1.2. Środowisko ewaluacyjne.....	3
2. Problem z przesłaniem komunikatu przy użyciu standardowych metod .NET	3
2.1. Opis problemu	3
2.2. Propozycja rozwiązania	4
3. Problem z obsługą w .NET podpisu jakim ZSMOPL opatruje komunikat zwrotny.....	4
3.1. Opis problemu	4
3.2. Propozycja rozwiązania	4



1. Przykładowa konfiguracja połączenia ze ZSMOPL

1.1. Środowisko produkcyjne

Na poniższych zrzutach zaprezentowano przykładową konfigurację połączenia ze ZSMOPL-em na środowisku produkcyjnym.

```
<client>
  <endpoint address="https://zsmopl.ezdrowie.gov.pl/cxf/zsmopl/ws/"
            binding="customBinding"
            bindingConfiguration="customObslugaKomunikatowBinding"
            contract="ZSMOPLService.ObslugaKomunikatow"
            name="ObslugaKomunikatowPort">
    <identity>
      <dns value="NAZWA_CERTYFIKATU"/>
    </identity>
  </endpoint>
</client>
```

```
<bindings>
  <customBinding>
    <binding name="customObslugaKomunikatowBinding">
      <security authenticationMode="MutualCertificate"

        messageSecurityVersion="WSecurity10WSTrustFebruary2005WSSecureConversationFebruary2005WSSecurityPolicy11BasicSecurityProfile10"
        includeTimestamp="false">
    </security>
    <textMessageEncoding messageVersion="Soap11" />
    <httpsTransport requireClientCertificate="false"/>
  </binding>
</customBinding>
</bindings>
```

```
<behaviors>
  <endpointBehaviors>
    <behavior name="customObslugaKomunikatowBehavior">
      <clientCredentials>
        <clientCertificate x509FindType="FindByThumbprint" findValue="CLIENT_THUMBPRINT"/>
      </clientCredentials>
    </behavior>
  </endpointBehaviors>
</behaviors>
```



```
<serviceCertificate>
  <defaultCertificate x509FindType="FindByThumbprint" findValue="CLIENT_THUMBPRINT"/>
</serviceCertificate>
</clientCredentials>
</behavior>
</endpointBehaviors>
</behaviors>
```

W kodzie należy ustawić podpisywanie wiadomości:

```
client.Endpoint.Contract.ProtectionLevel = ProtectionLevel.Sign;
```

1.2. Środowisko ewaluacyjne

Na środowisku ewaluacyjnym konfiguracja jest analogiczna, jak na produkcji – należy tylko zmienić adres na <https://ewa-zsmopl.ezdrowie.gov.pl/cxf/zsmopl/ws/>

```
<client>
  <endpoint address="https://ewa-zsmopl.ezdrowie.gov.pl/cxf/zsmopl/ws/"
    binding="customBinding"
    bindingConfiguration="customObslugaKomunikatowBinding"
    contract="ZSMOPLService.ObslugaKomunikatow"
    name="ObslugaKomunikatowPort">
    <identity>
      <dns value="NAZWA_CERTYFIKATU"/>
    </identity>
  </endpoint>
</client>
```

2. Problem z przesłaniem komunikatu przy użyciu standardowych metod .NET

2.1. Opis problemu

ZSMOPL odrzuca komunikat z zapisem w formie:

```
<zapiszKomunikatOS xmlns="http://csioz.gov.pl/zsmopl/ws/obslugakomunikatow/">
```



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>Unmarshalling Error: cvc-complex-type.2.4.a: Invalid content was found starting with element
'komunikatOS'. One of '{komunikatOS}' is expected.</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

2.2. Propozycja rozwiązania

Konieczne jest zmodyfikowanie generowanego przez WCF komunikatu (np. za pomocą mechanizmu Message Inspectors) w taki sposób, aby zamiast:

```
<zapiszKomunikatOS xmlns="http://csioz.gov.pl/zsmopl/ws/obslugakomunikatow/">
zastosować <obs:zapiszKomunikatOS> (z przestrzenią nazw) definiując
namespace xmlns:obs=http://csioz.gov.pl/zsmopl/ws/obslugakomunikatow/ w elemencie
nadrzędnym.
```

3. Problem z obsługą w .NET podpisu jakim ZSMOPL opatruje komunikat zwrotny

3.1. Opis problemu

Odpowiedź jest podpisana z wykorzystaniem tokenu X509PKIPathv1, który nie jest obsługiwany przez .NET Framework.

3.2. Propozycja rozwiązania

Aby poprawnie otrzymać odpowiedź należy samodzielnie zaimplementować obsługę tokenu lub usunąć element <wsse:Security> z nagłówka odpowiedzi (np. implementując MessageEncoder) oraz ustawić flagę EnableUnsecuredResponse w bindingu aby umożliwić otrzymanie niepodpisanej odpowiedzi.