**Przykład „0” – konfiguracja dla środowiska .NET**   
  
(Proszę zwrócić uwagę na ustawienie messageEncoding !)

<system.serviceModel>

<bindings>

<wsHttpBinding>

<binding name="e3" messageEncoding="Mtom">

<security mode="Transport">

<transport clientCredentialType="None" />

</security>

</binding>

</wsHttpBinding>

</bindings>

<client>

<!-- Wersja testowa -->

<endpoint address="https://Wyszukiwarkaregontest.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc"

binding="wsHttpBinding" bindingConfiguration="e3" contract="ServiceRegon.IUslugaBIRzewnPubl" name="e3" />

<!-- Wersja produkcyjna

<endpoint address="https://Wyszukiwarkaregon.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc"

binding="wsHttpBinding" bindingConfiguration="e3" contract="ServiceRegon.IUslugaBIRzewnPubl" name="e3" /> -->

</client>

</system.serviceModel>

**Przykład 1 – VB.NET**

Sub PolaczBIRpubl()

'Create the binding.

Dim myBinding = New WSHttpBinding

myBinding.Security.Mode = SecurityMode.Transport

myBinding.Security.Transport.ClientCredentialType = HttpClientCredentialType.None

myBinding.MessageEncoding = WSMessageEncoding.Mtom

Dim ea As New EndpointAddress("https://wyszukiwarkaregontest.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc") 'test

Dim cc As New UslugaBIRpubl.UslugaBIRzewnPublClient(myBinding, ea)

cc.Open()

'ZALOGUJ (bez sida)

Dim strSID As String = cc.Zaloguj("....klucz....")

txtResult.Text &= "### Zaloguj:" & vbCrLf & strSID & vbCrLf & vbCrLf

Using (New OperationContextScope(cc.InnerChannel))

Dim requestMessage As Channels.HttpRequestMessageProperty = New Channels.HttpRequestMessageProperty

requestMessage.Headers("sid") = strSID

OperationContext.Current.OutgoingMessageProperties(Channels.HttpRequestMessageProperty.Name) = requestMessage

'DANE SZUKAJ 1

Dim objParametryGR1 As New UslugaBIRpubl.ParametryWyszukiwania

objParametryGR1.Nip = "1111111111"

txtResult.Text &= "### DaneSzukaj (grupa 1):" & vbCrLf & cc.DaneSzukaj(objParametryGR1) & vbCrLf & vbCrLf

''DANE POBIERZ PEŁNY RAPORT

txtResult.Text &= "### DanePobierzPelnyRaport:" & vbCrLf & cc.DanePobierzPelnyRaport("39002176400000", "PublDaneRaportPrawna", "") & vbCrLf & vbCrLf

'WYLOGUJ

txtResult.Text &= "### Wyloguj:" & vbCrLf & cc.Wyloguj(strSID) & vbCrLf & vbCrLf

'GET VALUE

txtResult.Text &= "### GetValue(KomunikatKod):" & vbCrLf & cc.GetValue("KomunikatKod") & vbCrLf

txtResult.Text &= "### GetValue(KomunikatTresc):" & vbCrLf & cc.GetValue("KomunikatTresc") & vbCrLf

txtResult.Text &= "### GetValue(StatusSesji):" & vbCrLf & cc.GetValue("StatusSesji") & vbCrLf

End Using

cc.Close()

End Sub

**Przykład 2 – VB.NET**

Sub SOAPpubl()

'Create the binding.

Dim myBinding = New WSHttpBinding

myBinding.Security.Mode = SecurityMode.Transport

myBinding.MessageEncoding = WSMessageEncoding.Mtom

myBinding.Security.Transport.ClientCredentialType = HttpClientCredentialType.None

'adresy wewnętrzne

Dim ea As New EndpointAddress("https://wyszukiwarkaregontest.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc)") 'test

'Create the client.

Dim cc As New UslugaBIRpubl.UslugaBIRzewnPublClient(myBinding, ea)

Dim manualWebClient As New System.Net.WebClient()

manualWebClient.Headers.Add("Content-Type", "application/soap+xml; charset=utf-8")

Dim strSOAP As String = String.Empty

strSOAP &= "<soap:Envelope xmlns:soap=""http://www.w3.org/2003/05/soap-envelope"" xmlns:ns=""http://CIS/BIR/PUBL/2014/07"">" & System.Environment.NewLine

strSOAP &= "<soap:Header xmlns:wsa=""http://www.w3.org/2005/08/addressing"">" & System.Environment.NewLine

strSOAP &= "<wsa:To>https://wyszukiwarkaregontest.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc</wsa:To>" & System.Environment.NewLine

strSOAP &= "<wsa:Action>http://CIS/BIR/PUBL/2014/07/IUslugaBIRzewnPubl/Zaloguj</wsa:Action></soap:Header>" & System.Environment.NewLine

strSOAP &= " <soap:Body>" & System.Environment.NewLine

strSOAP &= " <ns:Zaloguj>" & System.Environment.NewLine

strSOAP &= " <ns:pKluczUzytkownika>...klucz...</ns:pKluczUzytkownika>" & System.Environment.NewLine

strSOAP &= " </ns:Zaloguj>" & System.Environment.NewLine

strSOAP &= " </soap:Body>" & System.Environment.NewLine

strSOAP &= "</soap:Envelope>" & System.Environment.NewLine

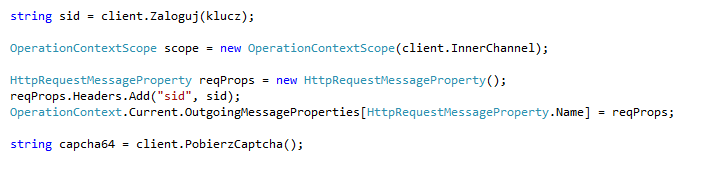
Dim bytArguments As Byte() = System.Text.Encoding.ASCII.GetBytes(strSOAP)

Dim bytRetData As Byte() = manualWebClient.UploadData("https://wyszukiwarkaregontest.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc", "POST", bytArguments)

txtResult.Text = System.Text.Encoding.ASCII.GetString(bytRetData)

End Sub

**Przykład 3 - Przekazanie wartości sid jako dodatkowego nagłówka żądania. Język C#**



**Przykład 4 – przejście ze środowiska testowego na produkcyjne. Język C#**

Najczęstszą przyczyną problemów przy przechodzeniu ze środowiska testowego na produkcyjne jest sytuacja, polegająca na tym, że adres usługi jest pobierany z pliku wsdl: <https://wyszukiwarkaregontest.stat.gov.pl/wsBIR/wsdl/UslugaBIRzewnPubl.xsd>  
który zawiera adresy usługi testowej.

**1)** Dlatego dla wersji produkcyjnej potrzebne jest wskazanie explicite adresu usługi produkcyjnej. Np.:string AdresUslugi = "<https://wyszukiwarkaregon.stat.gov.pl/wsBIR/UslugaBIRzewnPubl.svc>";

// (…)  
Usluga.Endpoint.ListenUri = new Uri(AdresUslugi);

xUsluga.Zaloguj("KluczUzytkownika");

**2)** W środowisku .NET proszę także pamiętać o właściwym ustawieniu w App.config:  
  
…  


Poniżej zamieszczamy informacje otrzymane od Usługobiorców i za ich zgodą dot. implementacji w innych środowiskach niż .NET**. Informacje, te przekazujemy „as is”, bez jakiekolwiek weryfikacji zasobów zamieszczonych pod n/w adresami oraz bez wsparcia technicznego dla tych rozwiązań.**

**5. Informacja dla tworzących oprogramowanie klienckie do usługi BIR1 w języku PHP**

„Jeśli będą mieli Państwo zapytania dotyczące dostępu do API z poziomy PHP to mogą Państwo przesyłać link do biblioteki stworzonej przeze mnie w języku PHP [https://github.com/johnzuk/GusApi](https://mail.stat.intra/owa/Regon_Bir@stat.gov.pl/redir.aspx?REF=odzlaTXRP3YfBe_tCjUxNxeU69NsVP-8P5xMywhbLdROOQfojzjUCAFodHRwczovL2dpdGh1Yi5jb20vam9obnp1ay9HdXNBcGk.)

**6. Informacja dla tworzących oprogramowanie klienckie do usługi BIR1 w języku JavaScript**  
„Stworzyliśmy paczkę do Node.js, którą śmiało mogą wykorzystać inni ludzie zainteresowani tematem <https://github.com/Appsolutly/node-regon>

**7. Informacja dla tworzących oprogramowanie klienckie do usługi BIR1 w języku Python**  
„Jeśli będą mieli Państwo użytkowników, którzy będą korzystać z Państwa usługi z pomocą programów w Pythonie, możecie im Państwo przesłać link do naszej Open Source’owej biblioteki litex.regon: [https://pypi.python.org/pypi/litex.regon](https://mail.stat.gov.pl/owa/Regon_Bir@stat.gov.pl/redir.aspx?SURL=H3Iz3e0FanACXzW271HbeeV_uS_y6pznf7mXdeGLK64JuUbbr5HSCGgAdAB0AHAAcwA6AC8ALwBwAHkAcABpAC4AcAB5AHQAaABvAG4ALgBvAHIAZwAvAHAAeQBwAGkALwBsAGkAdABlAHgALgByAGUAZwBvAG4A&URL=https%3a%2f%2fpypi.python.org%2fpypi%2flitex.regon)

**8. Informacja dla tworzących oprogramowanie klienckie do usługi BIR1 w języku Java**

„Klienta można w łatwy sposób wygenerować używając skryptu m.in. z pakietu Apache CXF wsdl2java. Rozwiązanie, z polisą AddressingFeature działa, lecz do zapewnienia pełnej funkcjonalności konieczne jest po uzyskaniu *sid'a,* dokleić go do nagłówków HTTP wychodzących requestów SOAPa.

Przykład użycia:  
  
UslugaBIRzewnPubl service = new UslugaBIRzewnPubl();

service.setHandlerResolver(new SoapHandlerResolver()); //<- doklejanie SID'a do HTTP HEADER

IUslugaBIRzewnPubl port = service.getE3(new AddressingFeature());

String statusUslugi = "StatusUslugi";

String result = port.getValue(statusUslugi);

// logowanie, jezeli sesja wygasla, badx logowanie pierwszy raz

if ((SoapMessageHandler.sessionCookie.equals("")) || (! result.equals("1"))){

String sid = port.zaloguj("abcde12345abcde12345");

SoapMessageHandler.sessionCookie = sid;

}

// przykład wyszukiwania po NIPie

ObjectFactory objectFactory = new ObjectFactory();

JAXBElement<String> nipParam = objectFactory.createParametryWyszukiwaniaNip("1234567890");

ParametryWyszukiwania parametryWyszukiwania = new ParametryWyszukiwania();

parametryWyszukiwania.setNip(nipParam);

String raport = port.daneSzukaj(parametryWyszukiwania);

**9**. **Informacja dla tworzących oprogramowanie klienckie do usługi BIR1 w środowisku Delphi**Zostało udostępnione kompletne rozwiązanie (cała klasa do komunikacji).   
Plik do pobrania znajduje się tutaj: <http://pliki.merco.pl/izzy.gus.zip>

**10. Informacja dla tworzących oprogramowanie klienckie do usługi BIR1 przy użyciu Ruby on Rails.**Link do naszej biblioteki Open Source’owej biblioteki gus\_bir1:  <https://github.com/espago/gus_bir1>

Należy dodać do pliku Gemfile:

gem 'gus\_bir1'

Skonfigurować dostępy

# /config/initializers/gus\_bir1.rb

# TEST CONF

GusBir1.production = false

GusBir1.client\_key = 'abcde12345abcde12345'

Wyszukiwać:

response = GusBir1.find\_by(regon: '00033150100000')

Pobierać dane:

response = GusBir1.find\_and\_get\_full\_data(nips: '8992689516,5261040828')

Przykładowe użycie oraz pełny opis dostępny na stronie biblioteki.