

SSL SOAP Web Servis (Kurulumu & Konfigürasyonu & Kullanımı)

Web Servisler, elektronik cihazlar arasında [www\(World Wide Web\)](http://www.WorldWideWeb) üzerinden iletişim kuran yapılardır. Web servisler aracılığıyla veri transferleri gerçekleştirilir. Bu örnekte HTTPS ile güvenli bir şekilde veri transferinin nasıl yapıldığını inceleyeceğiz.

HTTPS ile güvenli(şifreli) iletişim sağlayabilmek için servis sağlayıcı tarafından bize iletilen **SSL** sertifikasını "**STRUST**" işlem kodundan SAP sistemine yüklememiz gerekiyor. Bunun için aşağıdaki "**Figure 1**" ' de ki adımlarını izliyoruz.

SSL Kurulumu

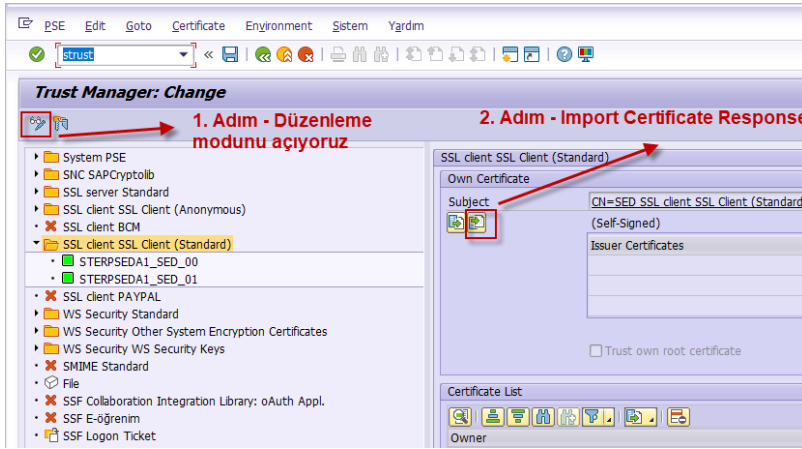


Figure: 1.1 STRUST - SAP üzerine sertifika yüklemesi

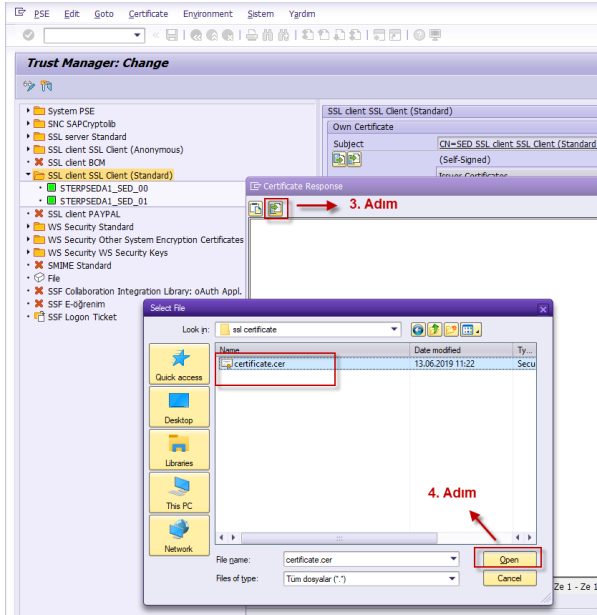


Figure: 1.2 STRUST - SAP üzerine sertifika yüklemesi

SOAP Web Servis Kurulumu

SAP sistemine servis kurulumu için servis sağlayıcının tarafımıza iletilmiş olduğu bir wsdl linki var ise linki kullanabiliriz. Ancak servis sağlayıcıları tarafından güvenlik sebebiyle web servis linkine doğrudan erişimin kısıtlandığı durumlar olabilir. Bu durumlarda SSL sertifikasıyla beraber tarafımıza(*client tarafına*) wsdl uzantılı ("**DosyaAdi.wsdl**") bir dosyanın iletilmesini bekleriz. Z'li bir paket içine **link** ya da **wsdl** dosyası üzerinden client tarafına web servis kurulumu için "**Figure 2**" ' de ki adımları izliyoruz.

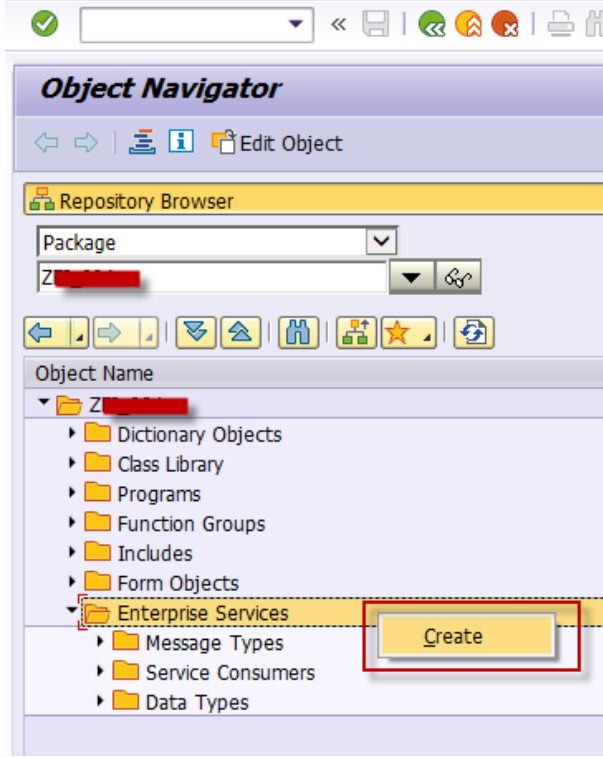


Figure: 2.1 Enterprise Services - Web servis kurulumu

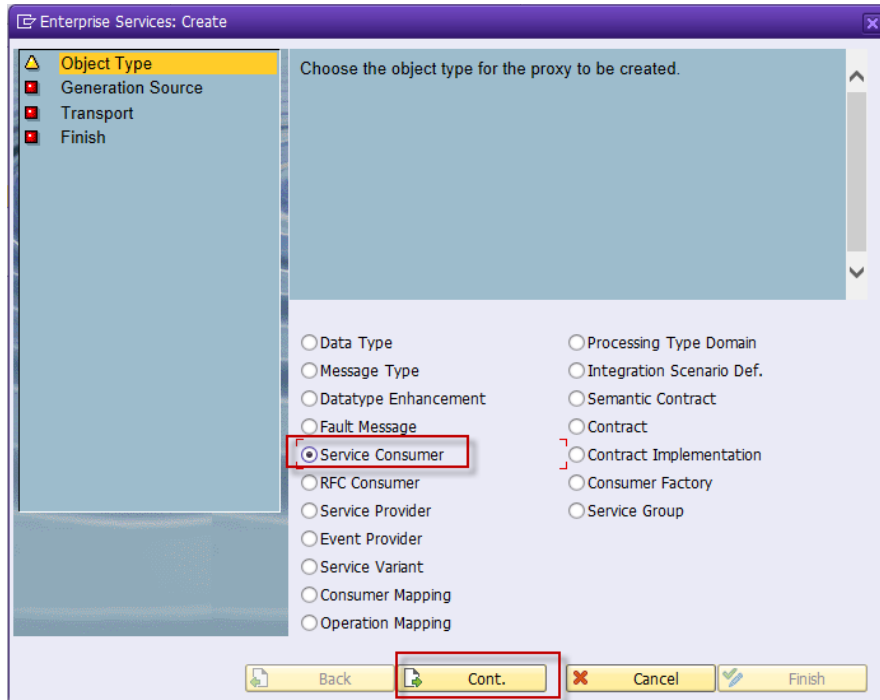


Figure: 2.2 Enterprise Services - Web servis kurulumu

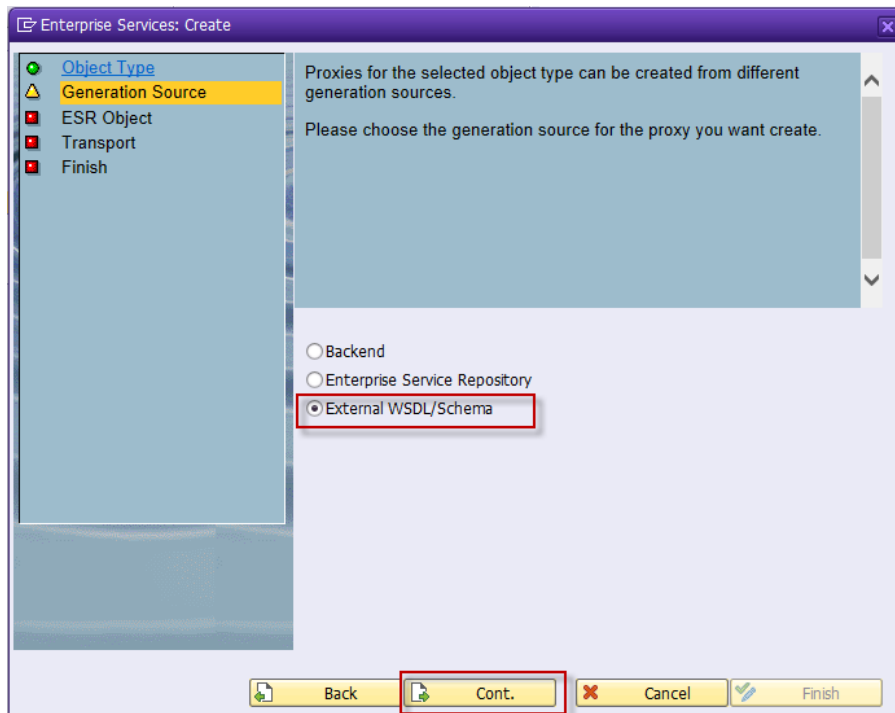


Figure: 2.3 Enterprise Services - Web servis kurulumu

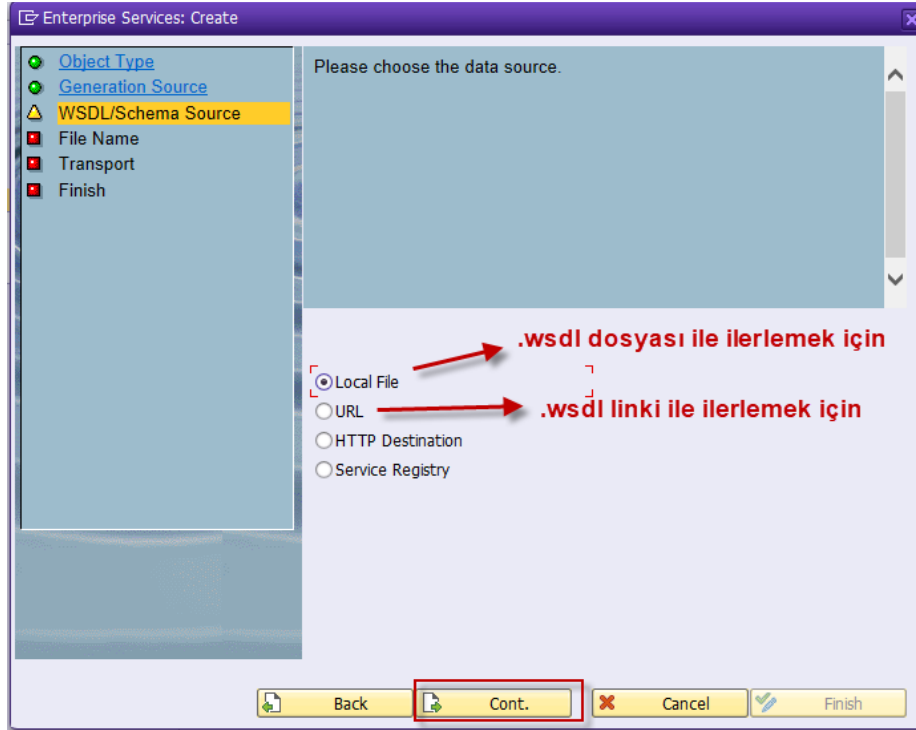


Figure: 2.4 Enterprise Services - Web servis kurulumu

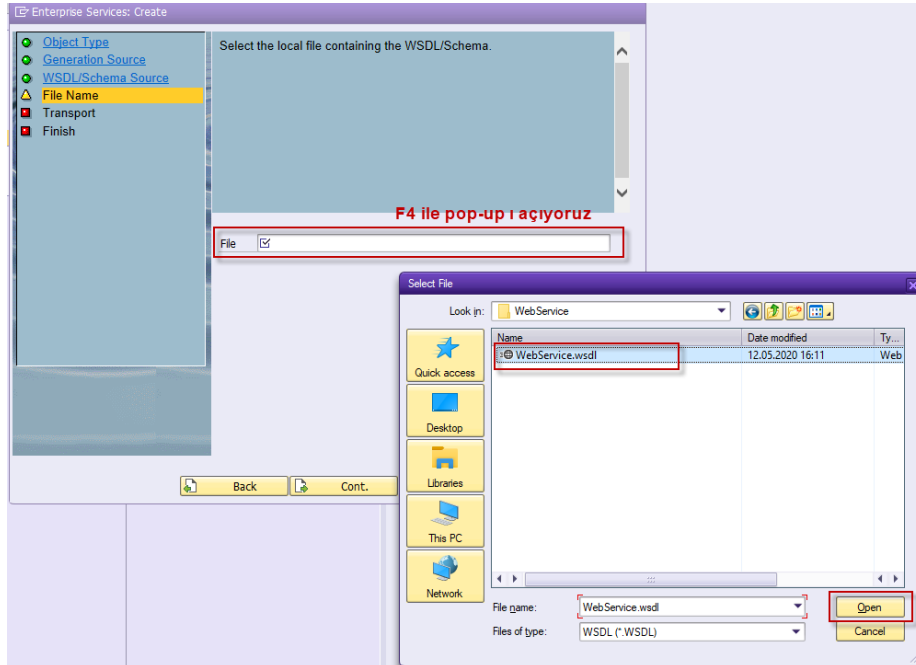


Figure: 2.5 Enterprise Services - Web servis kurulumu

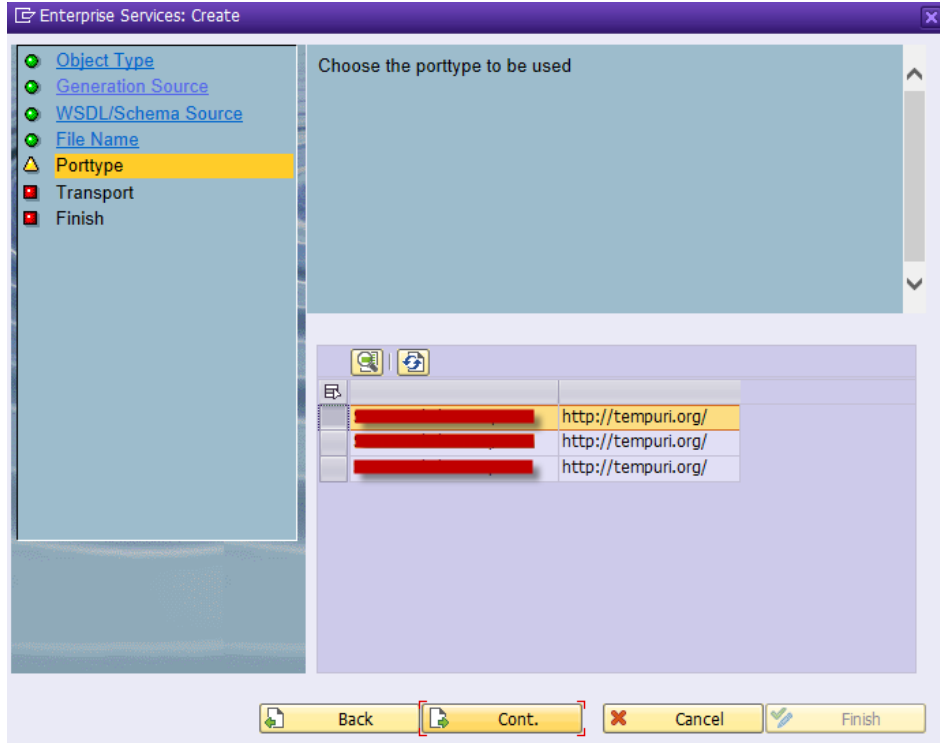


Figure: 2.6 Enterprise Services - Web servis kurulumu

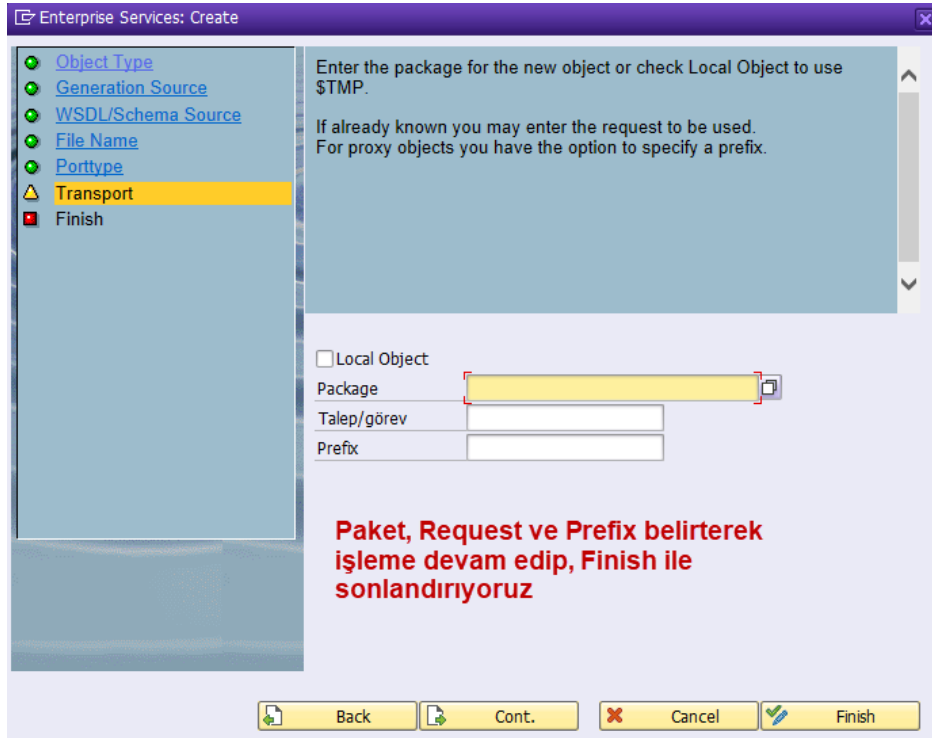
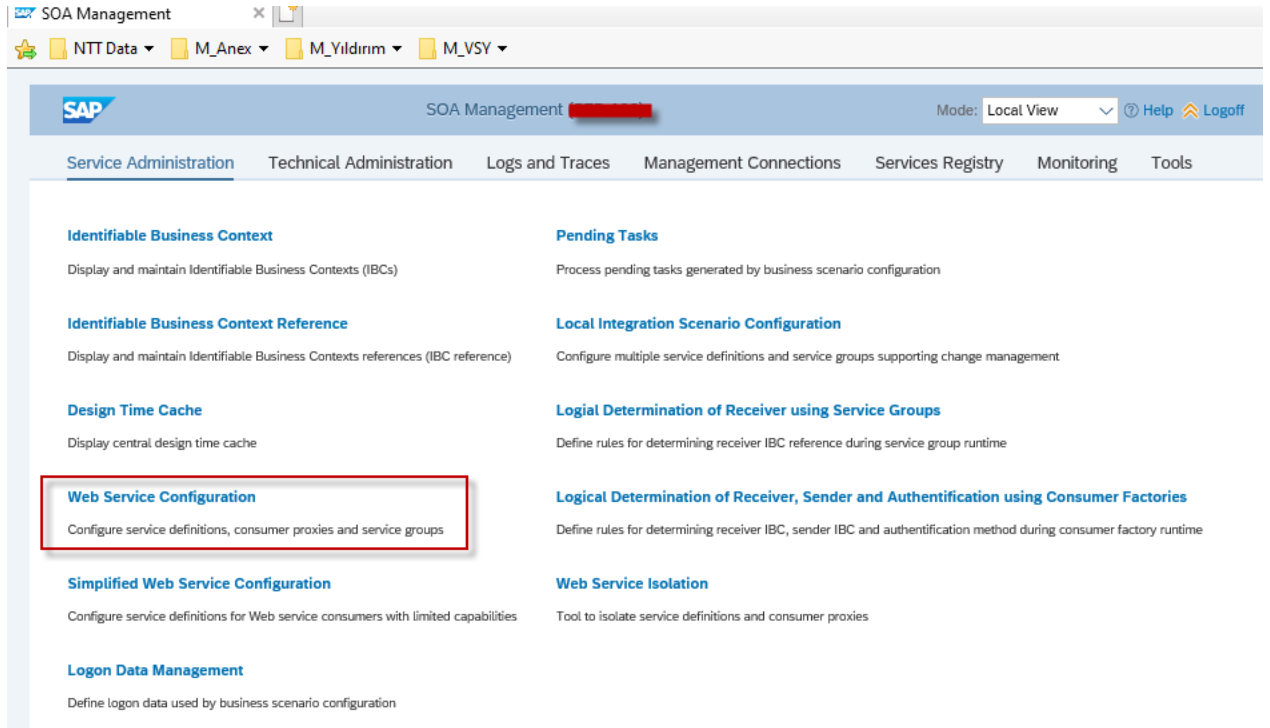


Figure: 2.7 Enterprise Services - Web servis kurulumu

SOAP Web Servis Konfigürasyonu

Soamanager üzerinden web servis konfigürasyon ayarlarını yapıyoruz. Bunun için "**T-Code**" kısmına "**soamanager**" yazıyoruz. Soamanager konfigürasyonu için "**Figure 3**" ' de ki adımları izliyoruz.



The screenshot displays the SAP SOA Management interface. The top navigation bar includes the SAP logo, the text 'SOA Management', and a 'Mode: Local View' dropdown menu. Below the navigation bar, there are several menu items: 'Service Administration', 'Technical Administration', 'Logs and Traces', 'Management Connections', 'Services Registry', 'Monitoring', and 'Tools'. The main content area is divided into two columns of configuration options. The 'Web Service Configuration' option is highlighted with a red rectangular box. The options listed are:

- Identifiable Business Context**: Display and maintain Identifiable Business Contexts (IBCs)
- Identifiable Business Context Reference**: Display and maintain Identifiable Business Contexts references (IBC reference)
- Design Time Cache**: Display central design time cache
- Web Service Configuration**: Configure service definitions, consumer proxies and service groups
- Simplified Web Service Configuration**: Configure service definitions for Web service consumers with limited capabilities
- Logon Data Management**: Define logon data used by business scenario configuration
- Pending Tasks**: Process pending tasks generated by business scenario configuration
- Local Integration Scenario Configuration**: Configure multiple service definitions and service groups supporting change management
- Logical Determination of Receiver using Service Groups**: Define rules for determining receiver IBC reference during service group runtime
- Logical Determination of Receiver, Sender and Authentication using Consumer Factories**: Define rules for determining receiver IBC, sender IBC and authentication method during consumer factory runtime
- Web Service Isolation**: Tool to isolate service definitions and consumer proxies

SAP

Design Time Object Search Configuration Search

Search criteria

Object Type is All

Object Name contains Z

Maximum Number of Results: 100

Search Clear values Reset search criteria

Search Result

Internal Name	Type
Z	Consumer Proxy

Servis adına tıklıyoruz

Figure: 3.1 Soamanager - Web servis konfigürasyonu

Overview Configurations Details

Define Logical Ports

Create Set Log.Port Default Activate

WSDL Based Configuration

Manual Configuration

Process Integration Runtime

Local Shortcut Configuration

Service Registry Based Configuration

Template Based Configuration

WSDL based Configuration with Template

Figure: 3.2 Soamanager - Web servis konfigürasyonu

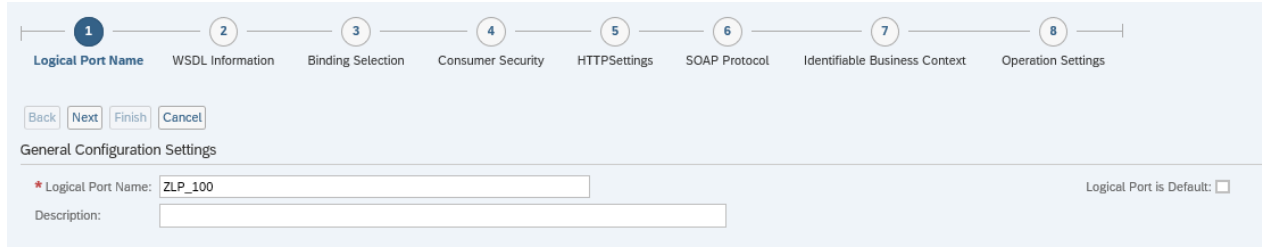


Figure: 3.5 Soamanager - Web servis konfigürasyonu

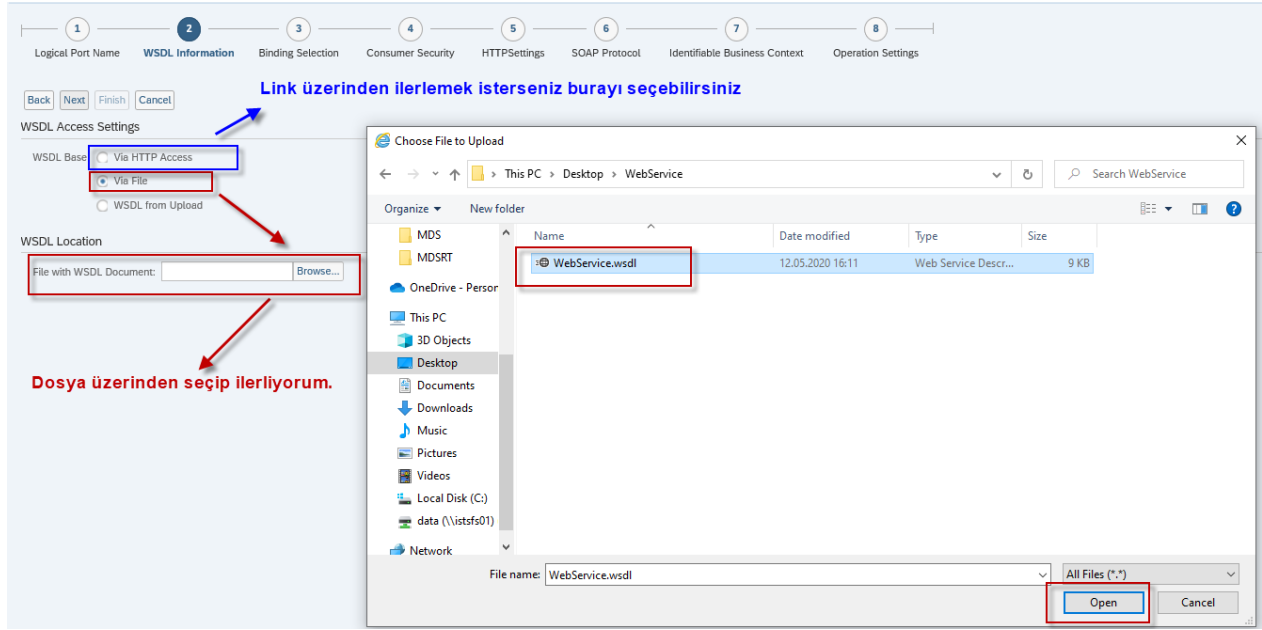


Figure: 3.3 Soamanager - Web servis konfigürasyonu

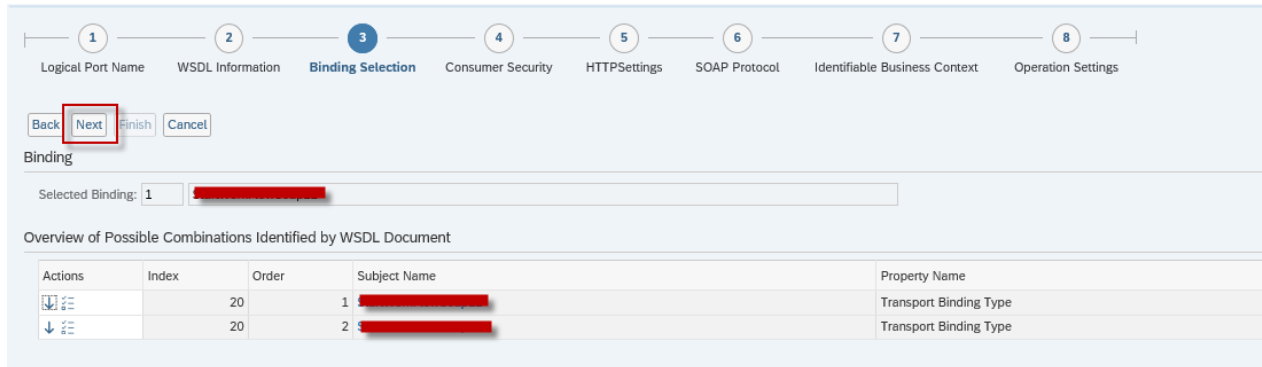


Figure: 3.4 Soamanager - Web servis konfigürasyonu

1 Logical Port Name 2 WSDL Information 3 Binding Selection 4 **Consumer Security** 5 HTTPSettings 6 SOAP Protocol 7 Identifiable Business Context 8 Operation Settings

Back Next Finish Cancel

Configuration of Consumer Settings additional to WSDL Document Information LP=ZLP_100

User ID/Password

User Name:
 Password:

Properties from WSDL Document

Authentication

Authentication Method:

Transport Security

Signature Expected:
 Encryption Expected:
 Sign Message:
 Add Encryption:

HTTPS servisi olduğu için servis kullanıcı bilgilerini kod kısmında servisin header bilgileri üzerinden vereceğiz. Bu kısmı boş bırakıyoruz

Figure: 3.5 Soamanager - Web servis konfigürasyonu

1 Logical Port Name 2 WSDL Information 3 Binding Selection 4 Consumer Security 5 **HTTPSettings** 6 SOAP Protocol 7 Identifiable Business Context 8 Operation Settings

Back Next Finish Cancel

URL Access Path

Complete URL URL components

* URL:

Logon Language:

Proxy

Name of Proxy Host:
 Port Number of Proxy Host:
 User Name for Proxy Access:
 Password of Proxy User:

Transport Binding

Make Local Call:

* Transport Binding Type:

Maximum Wait for WS Consumer:

Optimized XML Transfer:

Compress HTTP Message:

Compress Response:

Figure: 3.6 Soamanager - Web servis konfigürasyonu

1 Logical Port Name 2 WSDL Information 3 Binding Selection 4 Consumer Security 5 HTTPSettings 6 SOAP Protocol 7 Identifiable Business Context 8 Operation Settings

Back Next Finish Cancel

Message ID (Synchronous)

Message ID Protocol: Suppress ID Transfer

Metering of Service Calls

Data transfer scope: Enhanced Data Transfer

Transfer protocol: Transfer via SOAP header

Message Attachment Handling

Process Attachments: No

Figure: 3.7 Soamanager - Web servis konfigürasyonu

1 Logical Port Name 2 WSDL Information 3 Binding Selection 4 Consumer Security 5 HTTPSettings 6 SOAP Protocol 7 Identifiable Business Context 8 Operation Settings

Back Next Finish Cancel

Identifiable Business Context

Sender IBC Identifier:

Receiver IBC Identifier:

Suppress sending of IBC Identifier:

Figure: 3.8 Soamanager - Web servis konfigürasyonu

1 Logical Port Name 2 WSDL Information 3 Binding Selection 4 Consumer Security 5 HTTPSettings 6 SOAP Protocol 7 Identifiable Business Context 8 Operation Settings

Back Next Finish Cancel

Operation	Trans
<input checked="" type="radio"/> [Redacted]	<input checked="" type="checkbox"/>
<input type="radio"/> [Redacted]	<input type="checkbox"/>
<input type="radio"/>	<input type="checkbox"/>
<input type="radio"/>	<input type="checkbox"/>
<input type="radio"/>	<input type="checkbox"/>

WS /

Figure: 3.9 Soamanager - Web servis konfigürasyonu

SOAP Web Servis Kullanımı

Gelelim kod kısmına,

SSL sertifikalı bir SOAP web servisi kullandığımız için **<soap:Header>** tag ı altında servise ait kullanıcı bilgilerini iletiyoruz. Servis verileri şifreli olarak sunucu tarafına iletilecektir.

```
FUNCTION z*****.  
*-----  
  
CONSTANTS:  
    lc_lp_100                TYPE prx_logical_port_name VALUE 'ZLP_100'.  
  
DATA :  
    ls_input                 TYPE zsoap_in,  
    ls_output                TYPE zsoap_out,  
    lv_file_name             TYPE string.  
  
DATA: lv_xml TYPE string.  
  
DATA:  
    lr_header_protocol TYPE REF TO if_wsprotocol_ws_header,  
    lr_xml_root         TYPE REF TO if_ixml_element,  
    lr_xml_element      TYPE REF TO if_ixml_element,  
    lr_xml_document     TYPE REF TO if_ixml_document,  
    lv_xstring          TYPE xstring.  
  
DATA(lv_port) = COND prx_logical_port_name( lc_lp_100 ).  
  
lv_xml =  
    `<<soap:Header>` &&  
        `<<ValidationSoapHeader xmlns="http://tempuri.org/">` &&  
            `<<User>` && username && `<</User>` &&  
            `<<Password>` && password && `<</Password>` &&  
            `<</ValidationSoapHeader>` &&  
        `<</soap:Header>` .  
  
DATA(lo_service) = NEW zco_work_flow_soa(  
    logical_port_name = lv_port ).  
  
lr_header_protocol ?= lo_service->get_protocol( if_wsprotocol=>ws_header ).  
  
* convert to xstring  
lv_xstring = cl_proxy_service=>cstring2xstring( lv_xml ).  
  
IF NOT lv_xstring IS INITIAL.  
* create ixml dom document from xml xstring
```

```

CALL FUNCTION 'SDIXML_XML_TO_DOM'
  EXPORTING
    xml          = lv_xstring
  IMPORTING
    document     = lr_xml_document
  EXCEPTIONS
    invalid_input = 1
    OTHERS       = 2.

IF sy-subrc = 0 AND NOT lr_xml_document IS INITIAL.
  lr_xml_root = lr_xml_document->get_root_element( ).
  lr_xml_element ?= lr_xml_root->get_first_child( ).
* add header element by element to soap header
  WHILE NOT lr_xml_element IS INITIAL.
    DATA(name) = lr_xml_element->get_name( ).
    DATA(namespace) = lr_xml_element->get_namespace_uri( ).
    lr_header_protocol->set_request_header( name = name namespace = namespace dom = lr_xml_element ).
    lr_xml_element ?= lr_xml_element->get_next( ).
  ENDWHILE.
ENDIF.
ENDIF.

TRY.
  lo_service->MethodName(
    EXPORTING
      input = ls_input
    IMPORTING
      output = ls_output ).

  ev_return = ls_output- result.

  CATCH cx_ai_system_fault INTO DATA(lx_error).

ENDTRY.

ENDFUNCTION.

```