

How It Works

- [Main steps](#)
 - [Create a SOAP client](#)
 - [Generate auth data](#)
 - [Send request](#)
- [SOAP message structure](#)
 - [How to locate an operation](#)
 - [Operation headers](#)

Main steps

In order to exchange data with the web-service, third-party applications must build a SOAP client from WSDL specifications.

When developing a client application that interacts with the web service, you must comply with the following requirements:

Create a SOAP client

Create a SOAP client based on the service description in `voipnowservice.wsdl`.

It is recommended that you use the `voipnowservice.wsdl` file located on the VoipNow 3 server. However, you may keep the `voipnowservice.wsdl` file on a separate host.

If when creating a SOAP client using the `.wsdl` schemas you don't know which is the latest schema version that can be used to properly compile your request, you can use the `<latest>` parameter in the URL, just as shown below:

```
https://ip/soap2/schema/latest/voipnowservice.wsdl
```

VoipNow 3 will process your request using the latest version available for that `.wsdl` schema, for example 3.0.0. You will no longer be required to manually specify the exact version.

Generate auth data

Create a SOAP header containing the authentication data. Please read more on the authentication [here](#).

Send request

For all the requests that the client application sends to the web service, you should:

- Provide the data used by the request (as defined in the corresponding schemas).

Caution



The schemas defining the requests may contain elements highlighted as required. If you fail to include such elements in your request, the message will not be sent to the web service.

- Send the request to the web service.
- Process the response. The response structure is defined in the provided schemas.

SOAP message structure

A SOAP message is an ordinary XML document containing the following elements:

- **Envelope** - defines the XML document as a SOAP message, wraps the **Header** and the **Body**;
- **Header** - contains header information used by VoipNow 3 to authenticate the user that is making the request;
- **Body** - includes the call method and the response information;

You can locate the operations available for specific components by locating the correspondent XML schema. Usually these schemes bear the name of the component. For example, the component **ServiceProvider** has a corresponding schema entitled `ServiceProvider.wsdl`. Please refer to the section below in order to learn the steps you need to take in operating an operation.

How to locate an operation

All operations are listed in the section:

```

<portType name="<SCHEMA_INTERFACE_NAME>">
    ....
</portType>

```

and have the following structure:

```

<operation name="<OPERATION_NAME>">
    <input message="tns:<OPERATION_NAME>In" />
    <output message="tns:<OPERATION_NAME>Out" />
</operation>

```

Here is an example of the **AddServiceProvider** operation that allows you to add a new service provider to the system.

```

<portType name="ServiceProviderInterface">
    ...
    <operation name="AddServiceProvider">
        <input message="tns:addServiceProviderIn" />
        <output message="tns:addServiceProviderOut" />
    </operation>
    ...
</portType>

```

This operation can be found in the `ServiceProvider.wsdl` schema.

Operation headers

The headers used to make the SOAP call are set in the section:

```

<binding name="<SCHEMA_NAME>" type="tns:<SCHEMA_NAME>Interface">
    ...
</binding>

```

and have the following structure:

```

<operation name="<OPERATION_NAME>]">
    <soap:operation soapAction="http://4psa.com/<ENTITY>/3.0.0:<OPERATION_NAME>In" style="document" />
    <input>
        <soap:body use="literal" />
        <soap:header message="tns:<OPERATION_NAME>InHeader1" part="messagePart" use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
        <soap:header message="tns:<OPERATION_NAME>OutHeader1" part="messagePart" use="literal" />
    </output>
</operation>

```

Please find an example for the **AddServiceProvider** operation listed below:

```
<binding name="ServiceProvider" type="tns:ServiceProviderInterface">
  <operation name="AddServiceProvider">
    <soap:operation soapAction="http://4psa.com/ServiceProvider/3.0.0:addServiceProviderIn" style="
document"/>
    <input>
      <soap:body use="literal"/>
      <soap:header message="tns:addServiceProviderInHeader1" part="messagePart" use="literal"
/>
    </input>
    <output>
      <soap:body use="literal"/>
      <soap:header message="tns:addServiceProviderOutHeader1" part="messagePart" use="literal"
/>
    </output>
  </operation>
</binding>
```