

# Web Services

Web services allow invoking methods and fetching data on external web services which support the [WSDL standard](#).

- To add a new service click plus ( + ) symbol in the web services list from the process designer dashboard.
- Enter the web service address in the URL field. (Example: <http://www.websvcex.net/globalweather.asmx?WSDL> )
- Currently only WSDL standard is supported, but other service types will be added in the next releases.
- After clicking on 'Ok' button, service is automatically discovered in couple seconds, and service methods are displayed on screen.

Rest based services do not have any meta description and there is no need to register or discovery before using. You can directly call any rest service with [\\$Rest](#) object.

Service discovery scans the available methods on service and automatically registers the web service xml data types in "External Xml Schemas". Web service data types can also be directly used in the process data model.

In example, "GetWeather" is a method in service and expects the "CityName", "CountryName" fields to invoke and return GetWeatherResult type. Gray box contains the example of a usage for scripting. Return value of the method may be a basic value or a complex data type. You may need to inspect the data type from external xml schemas.

## Test Mode URL

This field is optional, you may leave it blank. If you want to use another service for testing environment, you can specify the URL of service. If the workflow instance is in test mode, this address is used automatically. Test and live service definition must be identical, otherwise unpredictable errors may occur.

## Consuming SOAP Service

### Calling Service from Scripts

After service is registered you can call the service from scripting environments including browser based client scripts. See [\\$Services](#) for further information.

```
var result = $Services.GlobalWeather.GlobalWeatherSoap.GetWeather({
  CityName : 'Amsterdam',
  CountryName : 'Netherlands',
});
```

### Using Service Results as Data Source

You may also use the service methods to populate lists (for example DropDowns).

After service is registered, add a new data source as "Web Service Query".

Click on "Edit" button to edit the data source. Service method is now displayed on screen and specify input parameters.

### Additional Options

All standard SOAP headers are automatically handled by the system, but in some scenarios you may need to set additional SOAP headers transfer information.

While invoking service, after specifying request body you may give secondary object to headers body. If service definition has matching header by name, specified value is imported to the SOAP message.

```
var result = $Services.GlobalWeather.GlobalWeatherSoap.GetWeather({
  CityName : 'Amsterdam',
  CountryName : 'Netherlands',
}, {
  MyHeader : 'MyHeaderValue'
});
```

## Consuming Rest Service

Rest services does not provide any discovery information and are dynamically invoked.

Please refer to the [\\$Rest](#) page to get information for consuming a rest service.