Global options are passed to the client's constructor and are specific to a service.

Although they are called "global options", they really are local to a client instance. Savon version 1 was based on a global Savon.configure method to store the configuration. While this was a popular concept back then, adapted by tons of libraries, its problem is global state. I tried to fix that problem.

#### wsdl

Savon accepts either a local or remote WSDL document which it uses to extract information like the SOAP endpoint and target namespace of the service.

Savon.client(wsdl: "http://example.com?wsdl") Savon.client(wsdl: "/Users/me/project/service.wsdl")

For learning how to read a WSDL document, read the [Beginner's Guide](http://predic8.com/wsdl-reading.htm) by Thomas Bayer. It's a good idea to know what you're working with and this might really help you debug certain problems.

#### endpoint and namespace

In case your service doesn't offer a WSDL, you need to tell Savon about the SOAP endpoint and target namespace of the service.

Savon.client(endpoint: "http://example.com", namespace: "http://v1.example.com")

The target namespace is used to namespace the SOAP message. In a WSDL, the target namespace is defined on the wsdl:definitions (root) node, along with the service's name and namespace declarations.

<wsdl:definitions name="AuthenticationWebServiceImplService" targetNamespace="http://v1\_0.ws.auth.order.example.com/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

The SOAP endpoint is the URL at which your service accepts SOAP requests. It is usually defined at the bottom of a WSDL, as the location attribute of a soap:address node.

<wsdl:service name="AuthenticationWebServiceImplService"> <wsdl:port binding="tns:AuthenticationWebServiceImplServiceSoapBinding" name="AuthenticationWebServiceImplPort"> <soap:address location="http://example.com/validation/1.0/AuthenticationService" /> </wsdl:port> </wsdl:service> </wsdl:definitions>

You can also use these options to overwrite these values in a WDSL document in case you need to.

#### raise\_errors

By default, Savon raises SOAP fault and HTTP errors. You can disable both errors and query the response instead.

Savon.client(raise\_errors: false)

### HTTP

#### proxy

You can specify a proxy server to use. This will be used for retrieving remote WSDL documents and actual SOAP requests.

Savon.client(proxy: "http://example.org")

#### headers

Additional HTTP headers for the request.

Savon.client(headers: { "Authentication" => "secret" })

#### timeouts

Both open and read timeout can be set (in seconds). This will be used for retrieving remote WSDL documents and actually SOAP requests.

Savon.client(open\_timeout: 5, read\_timeout: 5)

### SSL

Unfortunately, SSL options were [missing from the initial 2.0 release](https://github.com/savonrb/savon/issues/344). Please update to at least version 2.0.2 to use the following options. These will be used for retrieving remote WSDL documents and actual SOAP requests.

#### ssl\_verify\_mode

You can disable SSL verification if you know what you're doing.

Savon.client(ssl\_verify\_mode: :none)

#### ssl\_version

Change the SSL version to use.

Savon.client(ssl\_version: :SSLv3) # or one of [:TLSv1, :SSLv2]