SOAP Clients

Without leaving CTP, you can create, configure, and execute a scenario of tests that check the functionality of SOAP-based services—as well as review the outcome of those tests. To do this, you add SOAP Client tools to your test scenarios and configure them to suit your testing needs. You can then create and run a job for executing the related test scenarios—with the appropriate environment provisioned. You can also use these tests for checking the health of specific components and for automated testing during provisioning.

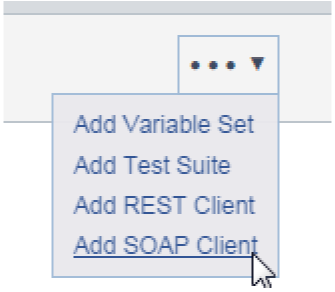
This functionality is intended to provide a fast way to create new SOAP client tools. For more advanced functionality (e.g., message payloads defined by custom scripts), open the related .tst in SOAtest Desktop.

> **Tip: Creating SOAP Clients from Traffic**
>
> If you have a traffic file that captures traffic for a SOAP service, you can generate a test scenario with SOAP Clients configured to test the recorded traffic. For details, see Creating New Test Scenarios and Test Suites.

Adding a SOAP Client

To add a new SOAP Client tool:

1. In the left pane, select the .tst or test suite where you want the new SOAP client added.
2. Choose **Add SOAP Client** from the page-level action menu.



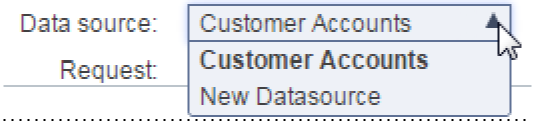
3. (Optional) Modify the name of the newly-created tool.
4. Configure the tool as described below.
5. Save the new tool configuration.

The new tool will be added at the end of the selected test suite.

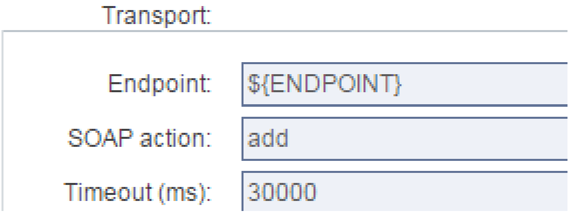
Configuring a SOAP Client

To configure a SOAP Client tool:

1. If the containing test suite includes multiple data sources, you can review and change which data source is used to parameterize this SOAP Client. See Parameterizing with Data Source and Data Bank Values for details.

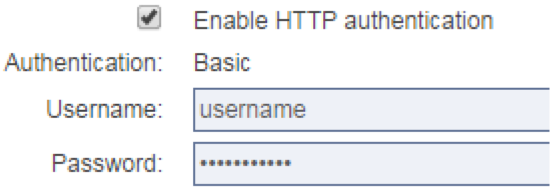

Data source: Customer Accounts

2. In the Transport area,specify the endpoint of the service to invoke as well as the SOAP action you want performed. You can also modify the timeout if desired.


Transport:
Endpoint: ${ENDPOINT}
SOAP action: add
Timeout (ms): 30000

3. (Optional) Specify HTTP authentication settings if desired.

    1. Check **Enable HTTP authentication**.
    2. Specify the username and password.



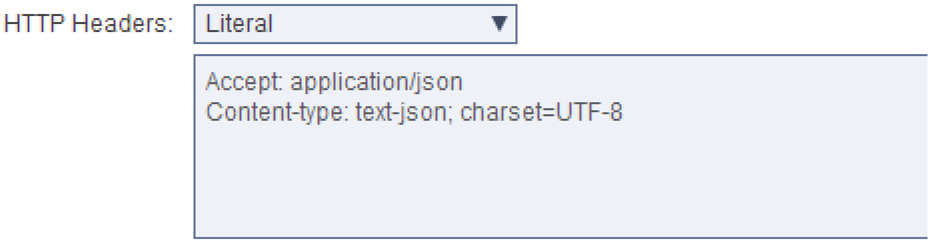
☑ Enable HTTP authentication

Authentication: Basic

Username: username

Password: ••••••••••••

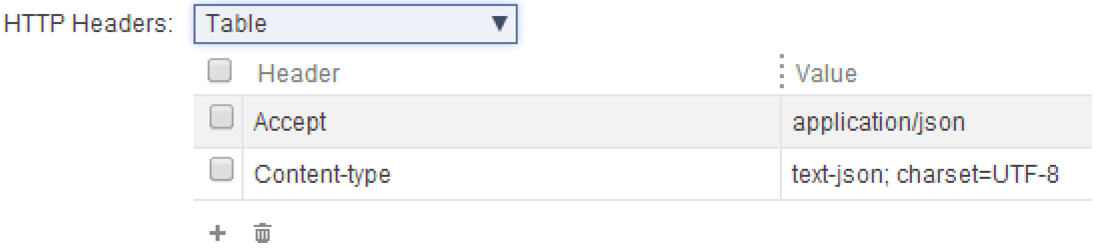
4. (Optional) Add HTTP headers if you want to override the default headers (described in the SOAtest documentation).
    1. Click **Add**.
    2. Specify a header name (this is case insensitive) and value. You can enter values in either table mode or literal mode. Here is an example in literal mode:



HTTP Headers: Literal ▼

```
Accept: application/json
Content-type: text-json; charset=UTF-8
```

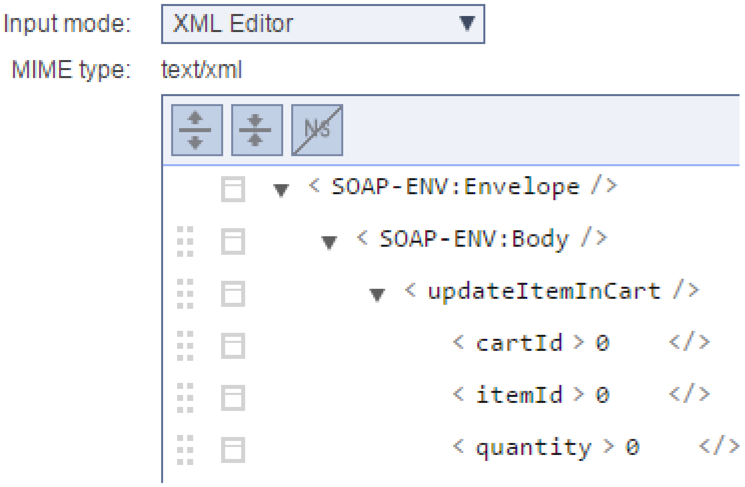
Here is that same example in table mode.



HTTP Headers: Table ▼

□	Header	Value
□	Accept	application/json
□	Content-type	text-json; charset=UTF-8

5. In the Request area, specify the message request that will be sent. Specify the message in the literal text editor, the JSON editor, or the XML editor (see Editing JSON Messages and Editing XML Messages for details and tips).



Input mode: XML Editor ▼

MIME type: text/xml

```
▼ < SOAP-ENV:Envelope />
    ▼ < SOAP-ENV:Body />
        ▼ < updateItemInCart />
              < cartId > 0    </>
              < itemId > 0    </>
              < quantity > 0    </>
```

> **Tip: Using {var_name}**
>
> If you're familiar with SOAtest, you can use the standard ${var_name} notation to reference environment variables, test suite variables, and data source values that are defined for the test scenario. This applies to both URL and Payload.
>
> When you are working with JSON, a special notation is used for parameterizing a number or boolean field within a JSON message: ${number:<value>} or ${boolean:<value>}. For example, to parameterize a number field with the column Count, you would use ${number:Count}.