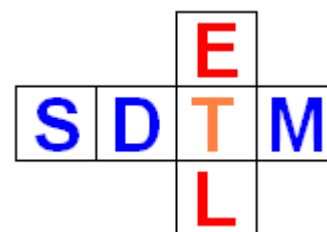


SDTM-ETL 5.0 User Manual and Tutorial: Using RESTful Web Services



Author: Jozef Aerts, XML4Pharma

Last update: 2025-02-27

Table of contents:

Introduction:.....	1
Using SDTM-ETL pre-defined RESTful web services	1
CDISC-CT and CDISC-SDTM functions:.....	1
LOINC functions:.....	2
Unit conversion RESTful web services functions.....	3
Creating your own RESTful web services functions	4
Calling a RESTful Web Service from within a mapping script.....	4
Using company-specific RESTful web services	6

Introduction:

RESTful web services are everywhere. Whether we order something on Amazon, book a flight, search for information, a RESTful web service is often behind the scene. RESTful web services become more and more important in medicine and pharma.

RESTful web services is about machine-machine communication (software to software), and, as the name states, is about "services". Using RESTful web services enables computer programs to use services from other computers, either from within the own company, or using publicly available services such as from the "[National Library of Medicine](#)" (NLM) or even as [provided by the FDA](#).

SDTM-ETL allows to directly use such services from within the mapping script.

A number of such services have been **pre-defined** in our scripting language, and are mostly related to the use of CDISC controlled terminology and to the use of LOINC coding. Others (e.g. company-internal ones) can easily be implemented.

Using SDTM-ETL pre-defined RESTful web services

SDTM-ETL comes with a number of pre-defined RESTful web service functions that can be used directly in mapping scripts. These are all related to the use of CDISC Controlled Terminology (CT) and to the use of LOINC, especially for lab tests. Keep in mind that, at least for lab results, the use of [LOINC coding is mostly mandatory by the FDA](#) in SDTM submissions!

The currently available RESTful Web Services functions are:

CDISC-CT and CDISC-SDTM functions:

A. `rws:testNameFromTestCode(test-code, test-code-system):`

returns the CDISC test name (--TEST value) from the CDISC test code (--TEST).

For example:

```
$lbtstname = rws:testNameFromTestCode('ALB', 'LBTESTCD');
```

Will return "Albumin" and store it in the variable \$lbtstname.

This function is especially interesting to automate the generation of LBTEST, VSTEST, ...
It automatically uses the latest version of CDISC-CT available.

B. rws:sdtmLabel(sdtm-version, sdtm-variable):

returns the label of an SDTM variable given the variable name and the SDTM model version (1.2, 1.3,1.4,1.7 or 2.0)

For example:

```
$label = rws:sdtmLabel('1.7', 'VISITNUM');
```

Will return "Visit Number" and store it in the variable \$label.

LOINC functions:

The following LOINC functions are extremely useful to obtain information from the LOINC code to be used in mappings to SDTM

We also have the "CDISC LOINC to SDTM-LB" mappings, for which there is a separate set of RESTful web services, as documented in the separate document "*Using the LOINC-SDTM-LB Mapping and similar functions*".

Already implemented LOINC services are:

A. rws:loincClass(LOINC-code):

returns the LOINC class from the LOINC code.

For example:

```
$loincname = rws:loincClass('1751-7');
```

Will return "CHEM" (chemistry) and store it in the variable \$loincname.

B. rws:loincComponent(LOINC-code):

returns the LOINC "component" (i.e. the chemical compound) from the LOINC code.

For example:

```
$component = rws:loincComponent('1751-7');
```

Will return "Albumin" and store it in the variable \$component.

C. rws:loincProperty(LOINC-code):

returns the measured property from the LOINC code.

For example:

\$property = rws:loincProperty('1751-7');

Will return "MCnc" ("mass concentration") and store it in the variable \$property.

- D. **rws:loincSystem**(LOINC-code):
returns the system from which the test is taken.

For example:

\$system = rws:loincSystem('1751-7');

Will return "Ser/Plas" (Serum / Plasma) and store it in the variable \$system.

- E. **rws:loincMethod**(LOINC-code):
returns the method used (if provided and important to uniquely identify the test).

For example:

\$method = rws:loincMethod('5792-7');

Will return "Test strip" and store it in the variable \$method.

- F. **rws:loincScale**(LOINC-code):
returns the scale type of the test (quantitative, ordinal, nominal, ...)

For example:

\$scale = rws:loincScale('1751-7');

Will return "Qn" ("Quantitative") and store it in the variable \$scale

- G. **rws:loincTimeAspect**(LOINC-code):
returns the time aspect of the test.

For example:

\$timeasp = rws:loincTimeAspect('21482-5');

Will return "24H" ("Protein concentration in 24 hour urine") and store it in the variable \$timeasp

- H. **rws:partExplanation**(part-type, part-name):
returns the explanation (definition) of a LOINC part.

For example:

\$explanation = rws:partExplanation('method','BCG');

returns the explanation for the LOINC method with LOINC name "BCG", which is:
"Bromocresol green (BCG) dye binding method"

Unit conversion RESTful web services functions

When generating SDTM datasets, generating (correct) code for unit conversions can be a nightmare. Not so with SDTM-ETL. We developed a number of unit conversion functions that are

based on the National Library of Medicine RESTful web services (see <https://ucum.nlm.nih.gov/ucum-service.html>).

A separate tutorial about using these functions "**Performing Unit Conversions in SDTM-ETL**" is also available from our [website](#).

Creating your own RESTful web services functions

SDTM-ETL users can easily create their own functions to use RESTful web services that are either available from their company's intranet, or from [publicly available RESTful web services](#) such as these provided by the [National Library of Medicine](#) (NLM).

In SDTM-ETL, it is very easy to create own reusable "functions" by adding some the code for it in the "functions.xml" file (in the "stylesheets" folder). Using the same mechanism, it is also very easy to create own RESTful web services functions by extending the file "**functions_restful_web_services.xml**".

Doing so requires some knowledge of XSLT, but every IT student nowadays knows XSLT very well. So, if you cannot do so yourself, just call your local IT department and ask for someone who knows XSLT.

The existing RESTful web service functions in the file "functions_restful_web_services.xml" can easily be used as starting examples for your own RESTful web service function calls. It is highly recommended to use the "http://www.xml4pharma.com/SDTM-ETL/RWS/ns" namespace and the "rws" prefix, as these are recognized by the SDTM-ETL software. When using another namespace and prefix, a warning will be given by the software, but one will be able to continue anyway.

Remark that currently, only "HTTP GET" can be used in SDTM-ETL functions and scripts. "HTTP POST" RESTful web service calls are currently not yet possible.

Calling a RESTful Web Service from within a mapping script

One can also pretty easily do a call to an existing RESTful web service, either from a public source such one of the many RESTful web services of the National Library of Medicine, or from the company's own intranet.

Often, it will be useful to create a variable as a "global" variable containing the "base" of the RESTful web service query string. As "global" variables can be referenced anywhere, this makes it easy to apply "write once, run many times".

Example: calling NLM RESTful web services for RxNorm

For example, suppose you would like to call the NLM RESTful web service to obtain information for a medication which has an [RxNorm](#) number. This function is described at: <https://rxnav.nlm.nih.gov/RxTermsAPIs.html> (there are many other functions).



RxTerms API

The RxTerms API is a web service for accessing the current RxTerms data set. No license is needed to use the RxTerms API.

Note: Please check the [Terms of Service](#) for restrictions on using the APIs.

The API can be accessed by clients in two different ways:

- RxTerms **RESTful** web services .
- RxTerms **SOAP** web services .

Functions and Resources

In the table, the base URI (<https://rxnav.nlm.nih.gov/REST/RxTerms/>) for the REST resources has been omitted to improve readability.

SOAP Function	REST Resource	Description
getAllRxTermInfo	/rxcuri/{rxcuri}/allinfo	Retrieve RxTerms information for a specified RxNorm concept
getRxTermDisplayName	/rxcuri/{rxcuri}/name	Retrieve the RxTerms display name for a specified RxNorm concept
getRxTermsVersion	/version	Retrieve the RxTerms version
getAllConcepts	/allconcepts	Retrieve all RxTerms concepts

The base of the function is: <https://rxnav.nlm.nih.gov/REST/RxTerms> and we store it in the "global" variable \$RXTERM, i.e. the script is:

```
$RXTERMBASE = "https://rxnav.nlm.nih.gov/REST/RxTerms";
```

As we learn from the API, we can ask all info about a medication with a specific "curi" (identifier) using:

```
https://rxnav.nlm.nih.gov/REST/RxTerms/rxcui/{curi}/allinfo.xml
```

e.g. calling "<https://rxnav.nlm.nih.gov/REST/RxTerms/rxcui/198440/allinfo.xml>" results in a response:

```
▼<rxtermsdata>
  ▼<rxtermsProperties>
    <brandName/>
    <displayName>Acetaminophen (Oral Pill)</displayName>
    <synonym>APAP</synonym>
    <fullName>Acetaminophen 500 MG Oral Tablet</fullName>
    <fullGenericName>Acetaminophen 500 MG Oral Tablet</fullGenericName>
    <strength>500 mg</strength>
    <rxtermsDoseForm>Tab</rxtermsDoseForm>
    <route>Oral Pill</route>
    <termType>SCD</termType>
    <rxcuri>198440</rxcuri>
    <genericRxcui>0</genericRxcui>
    <rxnormDoseForm>Oral Tablet</rxnormDoseForm>
    <suppress/>
  </rxtermsProperties>
</rxtermsdata>
```

In the specific SDTM-ETL mapping script for getting the "full name" of the medication, we then write e.g.

```
$rxcul = '198440';  
# the RESTful web service as a string  
$restquery = concat($RXTERMBASE, '/rxcul/', $rxcul, '/allinfo.xml');  
# call the RESTful web service and retrieve the "full name" (XML element "fullName")  
$fullname = doc($restquery)/rxtermsdata/rxtermsProperties/fullName;
```

The latter part (after doc(\$restquery)) being the path to be followed within the result XML.

IMPORTANT: all this is case sensitive!

And then using the value of \$fullname further on in the script.

The result that is obtained is the string "Acetaminophen 500 MG Oral Tablet".

Using company-specific RESTful web services

RESTful web services have the advantage that they are easy to implement. SDTM-ETL supports any RESTful web service (currently limited to "GET" methods) for which the API (i.e. the list with method descriptions) is known by the user of the software. This of course includes company-internal RESTful web services.

Users can either use these services directly from within the mapping scripts (as described above) or can be "encapsulated" into company-internal SDTM-ETL functions by extending the file "**functions_restful_web_services.xsl**".

For the server side of the RESTful web service, if you need help or assistance, [mail us](#). We have already developed dozens of RESTful web service for use in clinical research, including [a number of free and public ones](#), for which the documentation can be found on our "[RESTful Web Services web page](#)".