Yes, you can access the CDISC Library from SAS!

Conference Paper · April 2021

CITATIONS
0

READS
53

1 author:

Angelo Tinazzi
Cytel
100 PUBLICATIONS **3,046** CITATIONS

SEE PROFILE

2021 CDISC Europe Virtual Interchange

Yes, you can access the CDISC Library from SAS!

Angelo Tinazzi, Ctyel Inc.

ABSTRACT

With the availability of the CIDSC Library, vendors can now develop software from which you can instantly access standards i.e., CDSC Controlled Terminology. This is also true for users working with traditional software such as SAS.
In SAS, the CDISC Library can be queried through the CDISC API using PROC HTTP; its code is simple, and you can easily "interpret answers" e.g., json or xml.
The main purpose of this presentation is to provide my first "impressions" on the use of CDISC Library especially from the SAS user perspective. I will first discuss requirements to access the CDISC Library and the two different methods to access it, "Data Standards Browser" vs "API queries". I will then focus on the API use from SAS and the main "request" you can make. The Current limitations of the CDISC library will be also discussed.

INTRODUCTION

I started to think about this topic during last Christmas holiday and because we couldn't go anywhere I started to play with the CDISC Library from SAS to see how we could make use of it at Cytel, as although we are currently assessing alternative solutions or application and also developing internally some tools for more or less if not automatizing at least semi-automatize some of the process, the reality is that today especially for ADaM most if not all the programming is done in SAS and many metadata are handled using some sort of central SAS datasets repository we regularly update.
There have been several presentations already in the past events on how to access the library and how to make use of the CDISC Library API (Reference nr 1,2,3,4); also, some code snipsets/examples, including SAS, are available on the CDISC Library webpage (https://wiki.cdisc.org/display/LIBSUPRT/Getting+Started%3A+Programmatically+connect+to+CDISC+Library+API). However, I thought that could be beneficial to provide more examples and to clarify a couple of technical concepts, a sort CDISC Library for dummies explained by a CDISC Library dummy.

HOW TO ACCESS THE CDISC LIBRARY

THE CDISC LIBRARY IN A NUTSHELL

The CDISC Library is a metadata repository, the official CDISC metadata repository.
From the library we can extract the metadata in various formats, including our loved excel:

- .xlsx
- .csv
- .xml
- .json

The CDISC Library is structured, meaning that his format is predictable, whatever export format you chose.
Access to the library is available to CDISC Members and Open Source Dev only. Two accesses are possible:

- Data Standards Browser (https://library.cdisc.org)
- Querying through an API

Figure 1. Accessing the CDISC Data Standards Browser

DATA STANDARDS BROWSER VS API ACCESS

The following CDISC standards resources are currently available:

- CDASH
- SEND
- SDTM
- ADAM
- CDISC Controlled Terminology
- Draft Standards for testing purpose

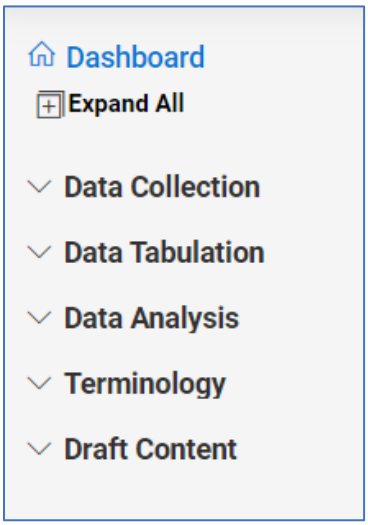

Figure 2. Browsing CDISC Standards Metadata

The Data Standards Browser

When you access the data standards library you have access to a dashboard where you can access several supportive resources (figure 3).

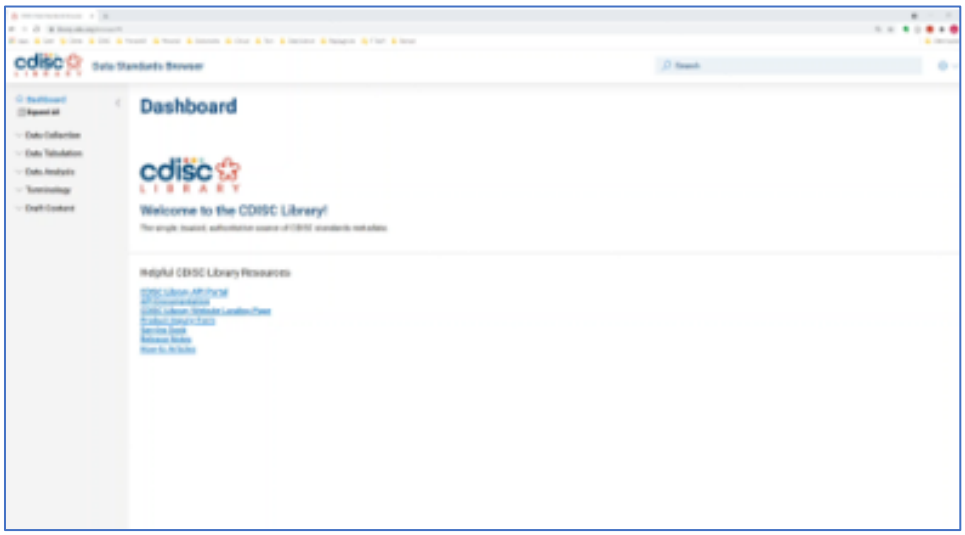

Figure 3. CDISC Library Dashboard

From there you can then interactively browse any version of any available standards. For example, for SDTM you can access all available standards and Implementation Guidance (IG), but also additional guidance such as the SDTM guidance for Medical Device.
Once you click the standard you want to browse, for example the latest SDTM IG, you will see the SDTM classes from which you can access the applicable standard SDTM domains. For example, if we select the Interventions class we can see all standard domains belonging to that class; we can for example browse the content of the EX/Exposure dataset then, still interactively, look at the details of each variable (figure 4), and if a variable has also associated a standard CDISC codelist, we can also look at the applicable codelist standard terms (figure 5), and from the same we can export the full content of the selected standard into either excel or csv format.

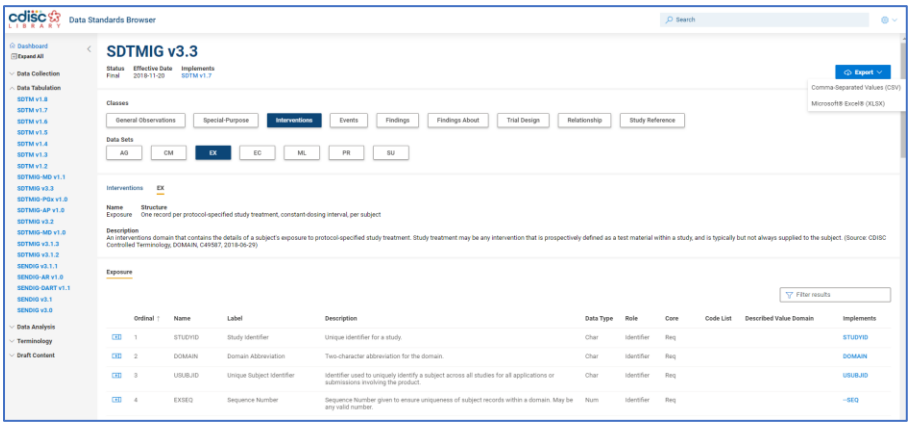

Figure 4. Browsing the SDTM IG 3.3

So with the CDISC Library Data Standards Browser it is not necessary anymore to look into the CDISC PDF documents except for the "assumptions“ and the implementation examples for which you still need to look into the IG, as of now at least.
The other method for accessing the CDISC Library is through the API, but before entering onto this topic, I need to clarify a couple of technical aspects and terms.

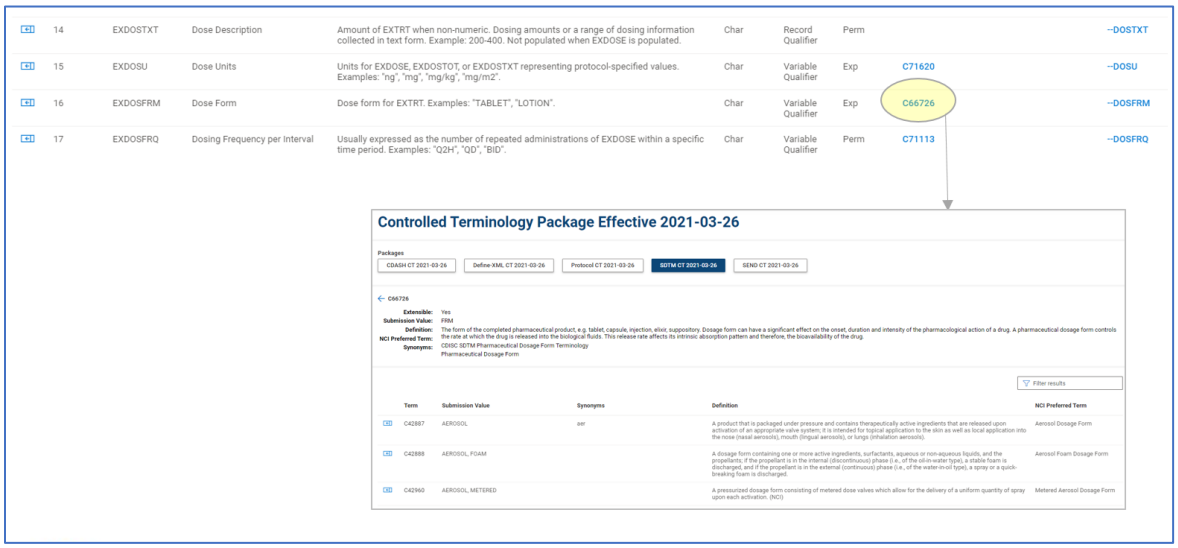

Figure 5. Applicable CDISC terminologies

API, REST and JSON

Let me try to explain what those API, REST and JSON are about also giving the fact they are mentioned in the CDISC Library documentation (https://www.cdisc.org/cdisc-library/api-documentation).

INTRO TO API AND REST

API stands for “Application Programming Interfaces” and it allows your “product” (app) or “service” to talk to another product or service. API are often used to give people access to data that the owner of the data decides to make accessible, either freely or with a payment. For example, many financial web sites give access to their users through API from which the user can write their own programs to for example access historical stocks data.

The Library Browser makes use of API in background, the CDISC Library API. This is for the user hidden and, as we saw before, when we browse the library through the Data Standards Browser a request is made to the CDISC Library server, the server elaborate the request and it returns to your internet browser the result in a readable format, but the real format it is something else (figure 6).

The CDISC Data Standards Browser makes use of the CDISC Library API. So, with the CDISC Library API, CDISC allows your application to access the CDISC metadata.

REST is the most popular API standard for web application, and it is made of two main statuses or methods for accessing the application (the user request for example) and the response with some data returned to the requestor.

The request is composed of four main elements:

- the Address, or the so called endpoint, where the address make uses of the URL format, so the way we already use to write web pages address when we want to access an internet resource through our internet browser
- an Header, where we supply additional instructions to the server, so to the API
- a Method, the type of action we want the server to do for us. For the time being only the GET method is available with the CDISC Library, meaning that we can only query the CDISC Library
- Data, this is only applicable for other methods I'm not covering here e.g. POST and PUT

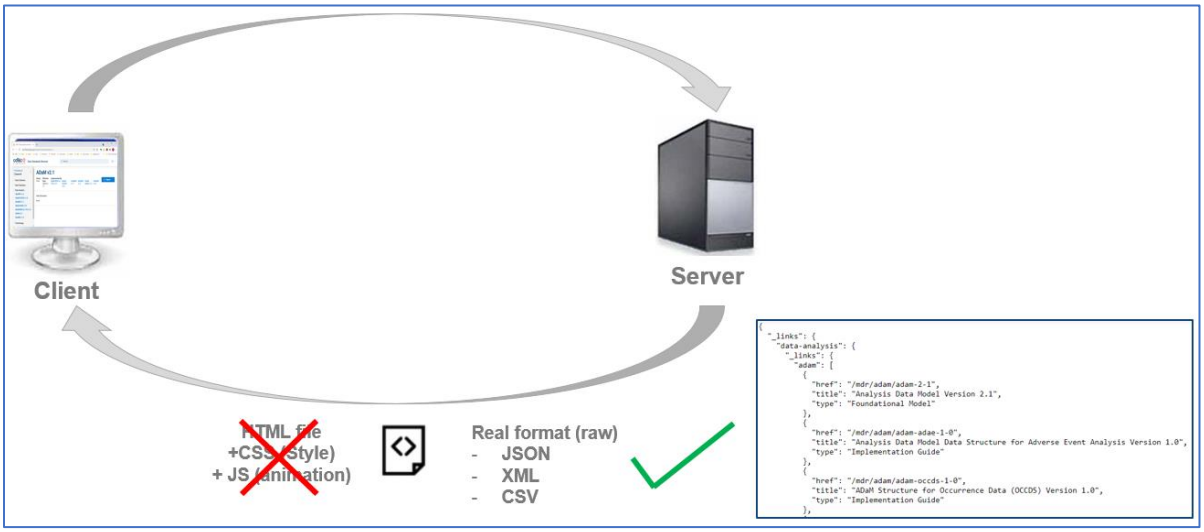

Figure 6. Interacting with the API

With the GET method you get a response, a file that can be based on different formats such as xml, json, but also csv and excel; the response has the following elements:

- a Header
- the Data we requested, so in the case of the CDISC Library it could be the content of a recent released CDISC CT
- the Status, whether the request was correct and the resource requested available on the server, or if the server was not available
- default Format for the CDISC API Library is JSON

JSON FORMAT

JSON stands for JavaScript Object Notation. If you are already familiar with the XML format, JSON is comparable to xml. Among its characteristics, the size of the JSON file is more compressed compared to xml. For example, if we query the CDIC Library to get the SDTM CT from last December the JSON file will be around 9 MB while the same in xml format is around 11MB (figure 7).

QUERYING THE CDISC LIBRARY FROM SAS

So, SAS is our client from which we want to access the CDISC Library, so being the CDISC library the server (figure 8). The SAS PROC HTTP it is the way to query an API from SAS, by sending a request to the API, in our case the CDISC Library API at the CDISC Library address (https://library.cdisc.org/api), and we want to get the answer in JSON format. The request is then interpreted by the server and a response is sent back to the SAS client in the format requested; of course we need to interpret the JSON

format that is not native to SAS and we can do that in SAS by using for example the JSON libname engine.

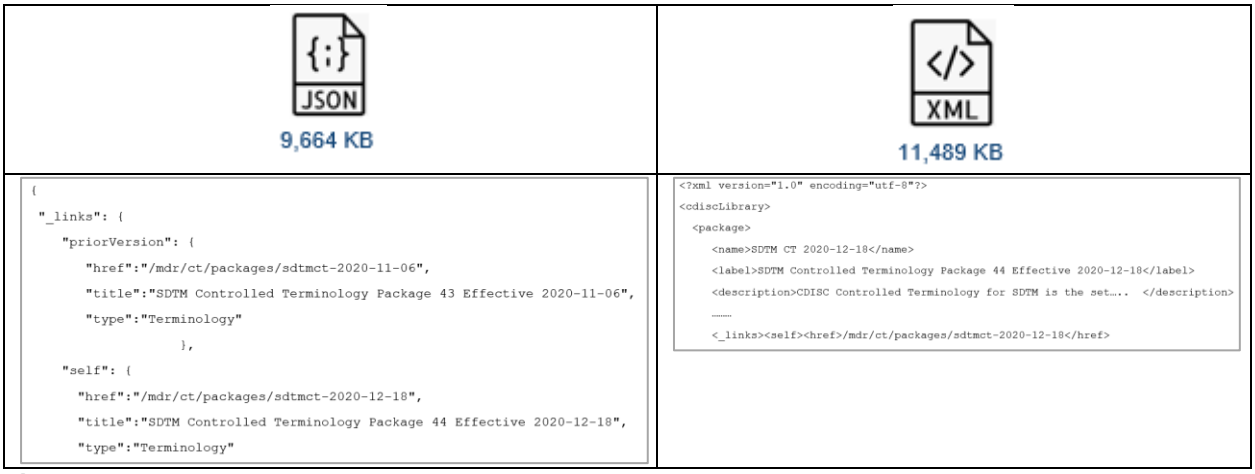

Figure 7. JSON vs XML

The query in the example in figure 8, returns the full list of products, standards, available in the CDISC Library. We can see that one of the product is the ADaM Standard version 2.1 and by using its address with the SAS proc HTTP we can access and get the metadata from the standard (figure 9).
Thus, in this model we can say that through the proc http we make a request to the librarian, our CDISC Librarian, the librarian makes the search and it returns to us the requested resource and through the proper SAS libname engine, in this case JSON, we can try to interpret the answer and make use of it.

Figure 8. Querying the CDISC Library from SAS

VIEWTABLE: Work.Allproducts

	P	P1	P2	P3	P4	P5	V	Value
1	1	_links					0	
2	2	_links	data-analysis				0	
3	3	_links	data-analysis	_links			0	
4	4	_links	data-analysis	_links	adam		0	
5	5	_links	data-analysis	_links	adam	href	1	/mdr/adam/adam-2-1
6	5	_links	data-analysis	_links	adam	title	1	Analysis Data Model Version 2.1
7	5	_links	data-analysis	_links	adam	type	1	Foundational Model
8	4	_links	data-analysis	_links	adam		0	
9	5	_links	data-analysis	_links	adam	href	1	/mdr/adam/adam-adae-1-0
10	5	_links	data-analysis	_links	adam	title	1	Analysis Data Model Data Structure for Adverse Event Analysis Version 1.0
11	5	_links	data-analysis	_links	adam	type	1	Implementation Guide
12	4	_links	data-analysis	_links	adam		0	

Figure 9. Getting full list of CDISC standards available in the CDISC Library

ANATOMY OF A PROC HTTP CDISC LIBRARY QUERY

Let's decompose now the PROC HTTP syntax.
The main part is the full address of the resource we need to access, that is made of the CDISC Library URL

https://library.cdisc.org/api/

followed by the endpoint so the exact resource we need to get

mdr/products

The full set of endpoints, so requests we can make to the CDISC Library API, is available at the following address: https://api.developer.library.cdisc.org, or it can be obtained with the *mdr/products* endpoint we saw in the earlier example (figure 10).

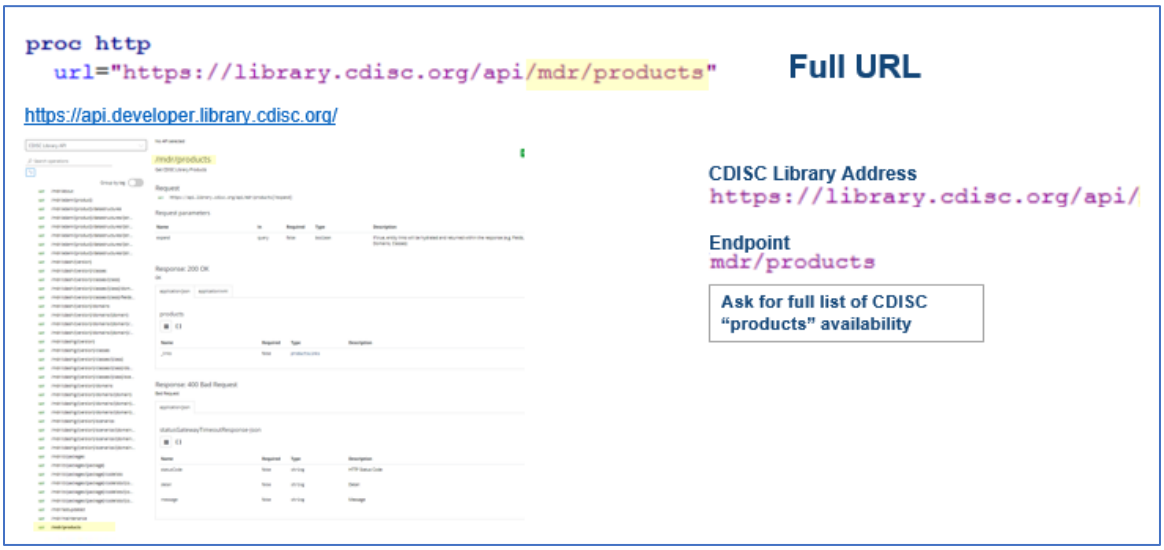

Figure 10. CDISC Library Endpoints

CHECKING THE RESPONSE FROM PROC HTTP QUERY

The response when returned will be saved in the file indicated in the OUT command so in our example in figure 11 the file will be simply named response.json and saved in my personal user area.
The header will have the authentication key I will have meanwhile obtained (generated) through my API developer CDISC library account (https://api.developer.library.cdisc.org); of note, in all the examples shown here the real key is masked.

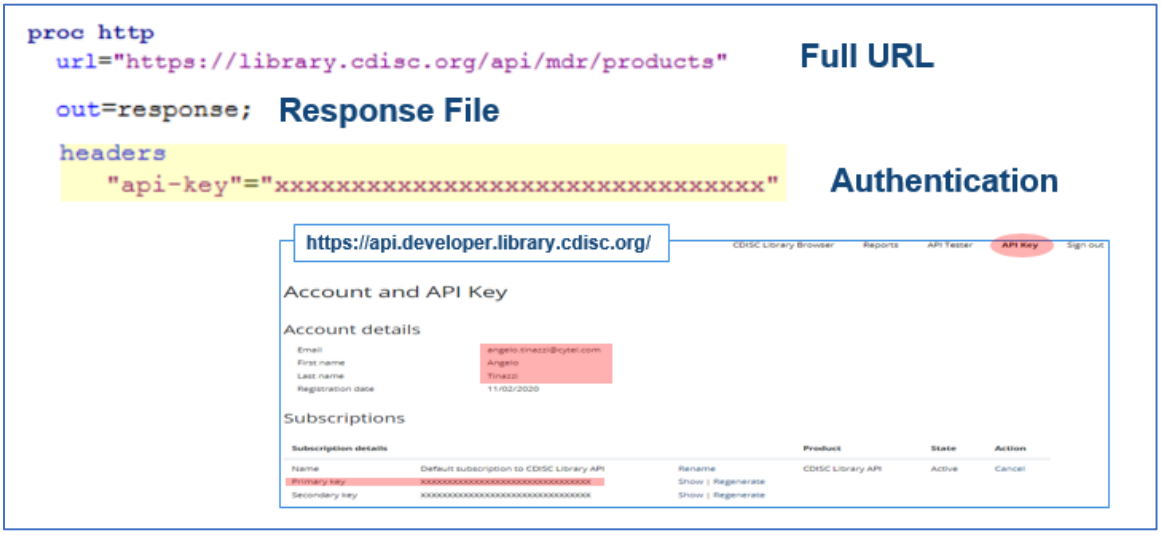

Figure 11. Anatomy of a PROC HTTP CDISC Library Query

After a while, hopefully a little while, we get the response back. The first thing we need to check is if our request was valid and therefore if the CDISC Library API was able to handle our request. This can be checked with the automatic macro variable *&SYS_PROCHTTP_STATUS_CODE*. For the CDISC Library API all the responses when correctly executed returns the code 200 so, because I did a little typo in my request, SAS returned an error code, or better the CDISC Library returned an error code (figure 12). The status code of each CDIC Library API can be consulted at the web site earlier mentioned.

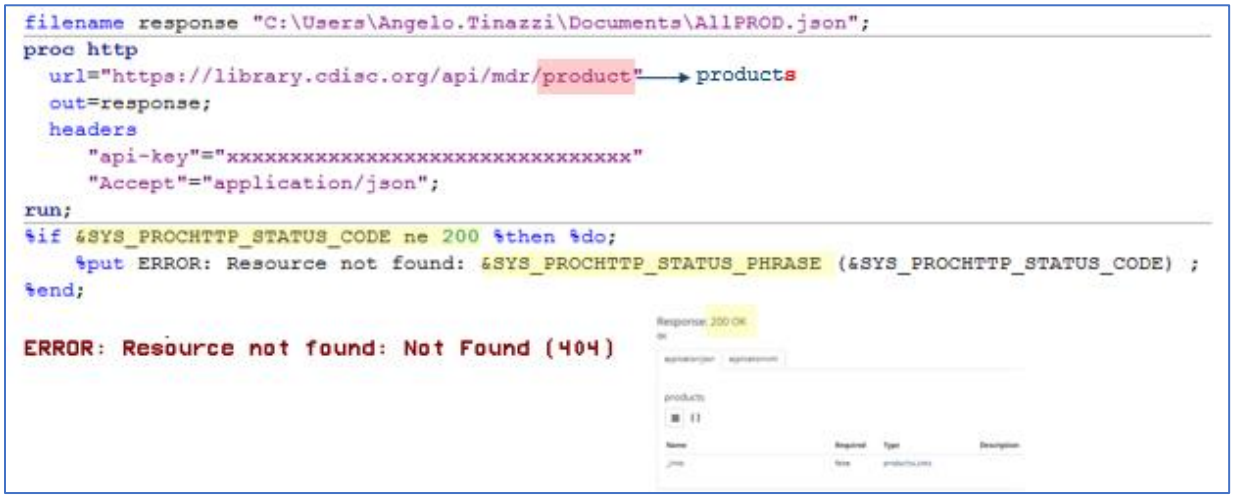

Figure 12. Checking the response status

INTERPRETING THE QUERY ANSWER –THE JSON ENGINE

Now, as I previously said, if we get a successful response, our returned file will have our request, in our case the info we need from the CDISC Standards, in our example a JSON file. We can then use some parsing techniques to interpret the JSON file or we can make use of the JSON SAS Libname engine. What the engine does for you is to try to "guess", or interpret, the structure of your JSON file and organize the information into several datasets. The number and type of datasets depend on the type of request we made, as you can see with the list and number of datasets (figure 13) returned by this other request where we asked for the SDTM IG 3.3 metadata for DM domain.

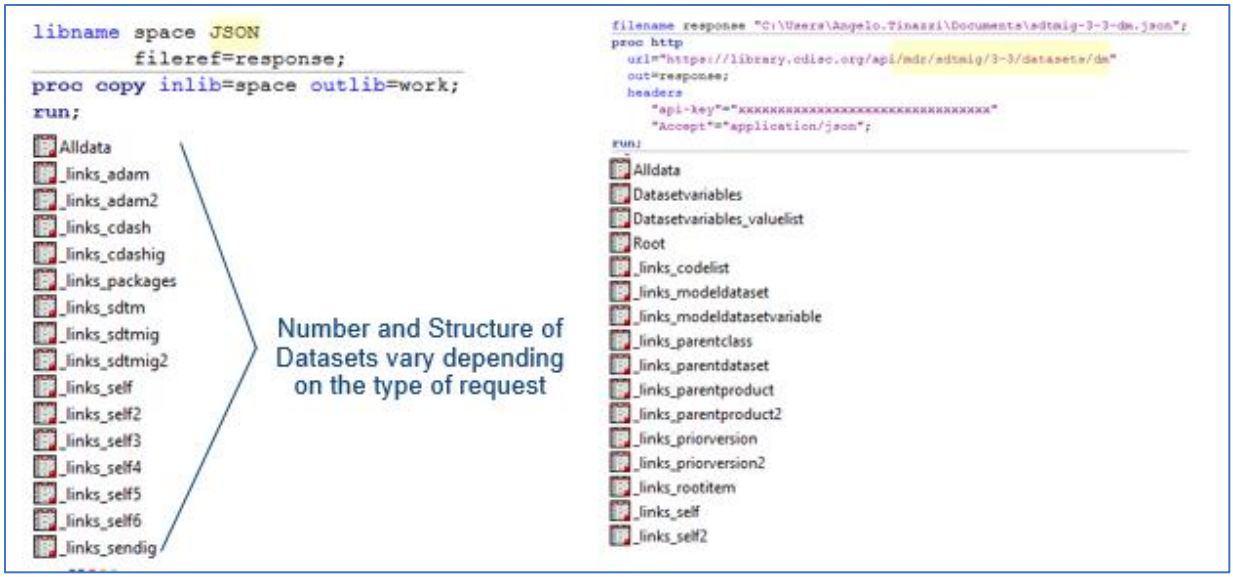

Figure 13. Capturing JSON response

WHAT USE OF THE CDISC LIBRARY WE CAN MAKE OF IN SAS

So now that we know how to query the CIDSC library from SAS, how much do you think it would be an added value for the SAS programmers to query and get use of the CDISC Library API from SAS? Well in my opinion accessing the library from SAS can be very useful to regularly check with some sort of automatic scheduled SAS programs if there is any update e.g. a new CDISC CT instead of regularly checking the CDISC website for any update or setting a quarterly reminder for example for the CDISC CT, and based on the fact that there is an update, we can automatically update our local tables, SAS datasets, where our organization stores CDISC metadata used for example by our SAS programmers when they do write code to either migrate legacy datasets to SDTM or when they create ADaM datasets. For this reason, because the way CDISC standards are handled nowadays by SAS programmers, there is not a big need of querying the library interactively from for example SDTM mapping programs.
I do also think, when available, getting responses in excel format (**application/vnd.ms-excel**) rather than JSON or XML, would be easier as the structure of the excel file is already in a tabular format, so very closed to how SAS organize data, although the JSON and XML formats have the advantage of having other metadata not available in excel such as the link to earlier versions.
If we look for example at the full *Media Type Support Matrix* at https://wiki.cdisc.org/display/LIBSUPRT/Media+Type+Support+Matrix, for the main guidance / standards we can easily get the excel file of for example one of the latest CDISC-CT or the last SDTM IG version 3.3, while instead if we want to get only a partial set of metadata, such as only the DM domain metadata, these are only available in either XML or JSON format.

CONCLUSIONS

From this first experience using the CDISC Library I can say that, while the Browser is a useful tool to interactively query the standards, as a SAS Statistical Programmer the API offers limited advantages. In general company doing the SDTM mapping directly in SAS, have metadata already organized somehow i.e., SAS datasets, so for this type of applications the library is only needed to regularly check for update and eventually automatically update sponsor / CRO metadata without going onto the CDISC website and check for any update and get the excel version of the metadata and import it.
There are number of aspects that I think can be improved, such as repeating mistakes in the IG, or the ability of more granular queries i.e. extracting all SDTM DM variables description, while the ability of comparing between versions i.e., diff file, as per last CDISC webinar, it seems this will be made soon available. Also, as per the recent CDISC webinar (*CDISC Webinar April 8th "CDISC Library: Ideas for using the CDISC Library and a Look at What's Coming Next", A. Chow and S. Hume*), there is a plan to make available additional content such as TAUGs and conformance rules and many other:

- ODM Media Type
- Implementation Guidance
- Conformance Rules
- Initial CDISC 360 Content (Draft)
- Diff Functionalities
- More Content to come

REFERENCES

1. How Will You Use CDISC Library?, S. Cassells, PHUSE-EU Connect. 2019
2. Implementing the CDISC Library API in software applications: first experiences, J. Aerts, PHUSE-EU Connect, 2019
3. CDISC Library Update, CDISC Data Science Team, PharmaSUG Japan, 2019
4. CDISC Library try-out: from implementation to evaluation of the API. R. Radelicki, PHUSE-US Connect, 2020
5. A Beginners Guide to Consuming RESTful Web Services in SAS®, L. de Walick, SAS Global Forum, 2020
6. The ABCs of PROC HTTP, J. Henry, SAS Global Forum, 2019
7. Creating and Controlling JSON Output with the JSON Procedure, A. Linker, SAS, SAS Global Forum, 2019
8. The JSON LIBNAME Engine: Real-World Application Using Open APIs, M. Drutar, E. Thies
9. REST at Ease with SAS: How to use SAS to get your REST, J. Henry

ACKNOWLEDGEMENTS

I would like thanks Lex Janssen and Jozef Aerts for their valuable input.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Angelo Tinazzi
Cytel Inc.
Route de Prè-Bois 20
1215 Geneva – Switzerland
Email: angelo.tinazzi@cytel.com
Web: www.cytel.com