Paper BB-05-2015

Retrieving Survey Data using the Qualtrics REST API with SAS®

Katie Tanner, Capella University, Minneapolis, MN

ABSTRACT

Qualtrics Research Suite is a powerful tool for collecting data online. It features a Representational State Transfer (REST) Application Programming Interface that allows for other programs to interact with the Qualtrics system. Using the HTTP procedure first introduced in SAS® 9.2, SAS has the ability to interface directly with the Qualtrics system. Programs can be easily created to seamlessly transfer survey data from the data collection tool in to SAS® for further analysis and manipulation.

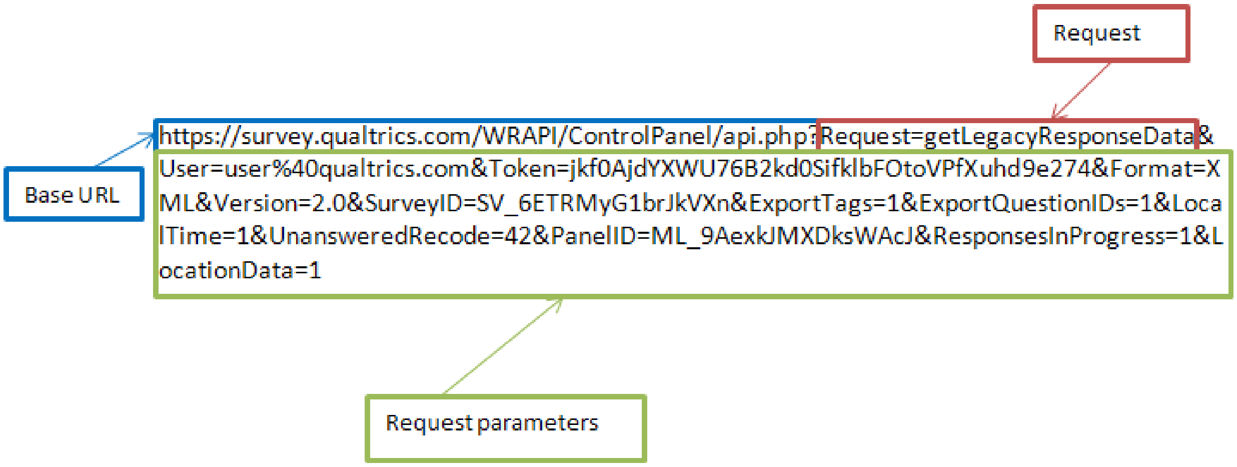
This paper will walk through a process for accessing the Qualtrics API and storing the resulting data in a SAS dataset. If the survey remains unchanged, this process could run automatically to provide continuous update of survey information for reporting and archival purposes.

INTRODUCTION

Qualtrics Research Suite is a survey hosting platform that allows for the collection and analysis of data online. The software interface is designed to be intuitive, however advanced back-end capabilities such as incorporating JavaScript into question properties and custom CSS are also available to end users. This allows for a good balance of ease of use and flexibility for a wide spectrum of users.

Survey data can be extracted from the tool in two ways: by using the tool interface to download the data to a text file or by using the end-user Application Programming Interface (API). Qualtrics uses a Representational State Transfer, or REST-based API protocol, which uses HTTP requests to post, read and modify data. This allows users to automate many activities of survey administration, including adding survey participants, modifying survey messages, and downloading responses. Furthermore, survey administrator tasks such as managing user accounts and consolidating lists of participant opt-out requests can also be done through the API.

Communication with the API is called a request, and this happens through a URL. The request begins with a base URL that point to the Qualtrics system. The next part of the request is known as the parameters. The parameters tell the API what to do. Some parameters are required for the request to work. Others are optional, and allow for the customization of the output, also known as the API response. Below is an example of a common API request, in this case downloading a XML file of survey data from Qualtrics. Entering this URL into a web browser generates the response from the API. Sometimes, this response is data, other times it can be information as to whether or not the request was successfully executed.



```
https://survey.qualtrics.com/WRAPI/ControlPanel/api.php?Request=getLegacyResponseData&
User=user%40qualtrics.com&Token=jkf0AjdYXWU76B2kd0SifklbFOtoVPfXuhd9e274&Format=X
ML&Version=2.0&SurveyID=SV_6ETRMyG1brJkVXn&ExportTags=1&ExportQuestionIDs=1&Loca
lTime=1&UnansweredRecode=42&PanelID=ML_9AexkJMXDksWAcJ&ResponsesInProgress=1&L
ocationData=1
```

Figure 1. A common API request to get data from Qualtrics

Because the API requests can be cobbled together from different combinations of different parameters and values, SAS can be used create URLs that send requests to the Qualtrics API. Values for the parameters can be written into a SAS program or the API calls can be dynamic and data driven using macros. Furthermore, SAS can send the request URL to the Qualtrics system using the HTTP procedure. The HTTP procedure was introduced in SAS 9.2 as a way for SAS to communicate with the REST Protocol API services. The procedure works by reading in a file, identified by a fileref and writes output to another file. Within the procedure, SAS issues HTTP requests, such as “get” to obtain information from the Qualtrics system or “post” to upload information to the Qualtrics system. From there, the API response data can be brought in to SAS for further analysis or for archive.

PROGRAM OVERVIEW

This paper will present a basic process for seamlessly transferring data out of Qualtrics and in to SAS. The program steps are listed below:

1. Set up formats
2. Set up output directory using FILENAME statement.
3. Send the API request using the HTTP procedure’s “GET” request
4. Set up for XML parsing using libname xml and xmlmap
5. Bring in to SAS using the DATA step to move to a permanent library and perform other data manipulation

SET UP FORMATS

Data will be coming in to SAS via a XML file that is downloaded using the Qualtrics API. To translate XML data to a data set, the programmer can use a XMLMAP. The XMLMAP allows the programmer to define responses and columns, however it will also do some manipulation through the application of formats and informats that can be defined. The application of the user defined formats can occur in the XMLMAP so long as formats are defined previously in the program.

```
/*User defined formats and informats*/

   proc format;

   /*I want to recode some of the data that comes out of Qualtrics*/

       invalue flag 2=0;

   /*Now I want to add some data labels for the numeric data*/

       value tell   1='No, thank you'
                    2='Sure. Something'
                    3='What do you want me to say?';

       value flg_txt1='Yes'
                    0='No';
   run;
```

SET UP OUTPUT DIRECTORY

In order for the HTTP procedure to run, SAS needs to be able to read in and output a file. Therefore, the first step in our program needs to reference the creation of a file. File paths can be long, so a filename statement at the beginning of the program can allow us to create an alias for this file throughout the rest of the program.

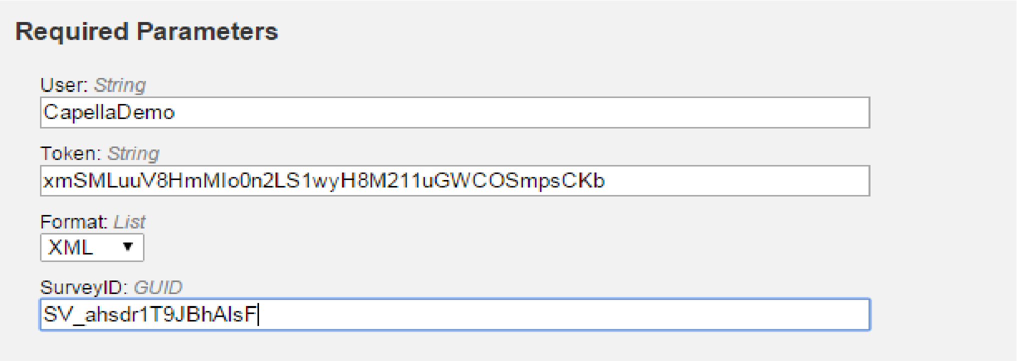
```
   /*Set up our filename path - this tells SAS where to save the output data from the
   API*/
   filename Qdata "C:\Users\KTanner1\Qualtrics data download &SYSDATE9..xml";
```

Macro variables can be added to this filename statement to prevent file overwrite and to provide versioning.

CALL THE QUALTRICS API USING HTTP PROCEDURE

This process of retrieving data out of Qualtrics using the API only requires three arguments. The method tells SAS what kind of REST API request we are making, whether data is to be obtained from the system (“GET”) or data is to be pushed to the system (“POST”). In this case, data is to be obtained from the system, so the appropriate method is “GET”. The REST API allows information to be posted to Qualtrics through the API as well.

The next argument is to input the URL that contains the request. In this case, URL is written in to the program. Static API request URLs can be built from the Qualtrics REST API documentation website by filling in parameter information into form fields as shown below.



Required Parameters

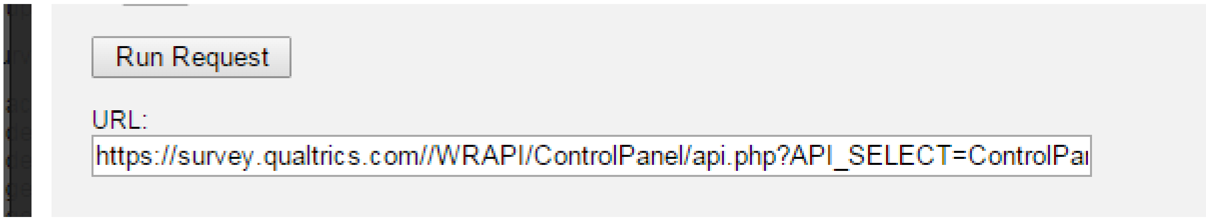
User: *String*
CapellaDemo

Token: *String*
xmSMLuuV8HmMIo0n2LS1wyH8M211uGWCOSmpsCKb

Format: *List*
XML ▼

SurveyID: *GUID*
SV_ahsdr1T9JBhAlsF

Figure 2. Qualtrics form for creating an API request URL



URL:
https://survey.qualtrics.com//WRAPI/ControlPanel/api.php?API_SELECT=ControlPa

Figure 3. The resulting API request URL

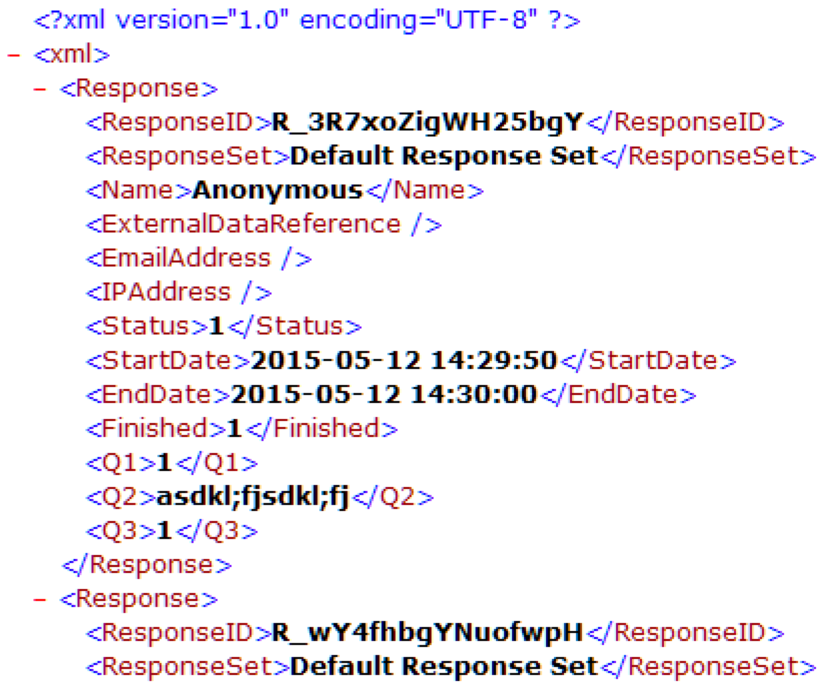
The final argument involves telling the HTTP procedure where to put the output data. This should point to the filename mentioned in the program previously. The entire procedure call looks like this.

```
/*Call the API in order to retrieve the data*/

proc http method="get"
    url='https://survey.qualtrics.com//WRAPI/ControlPanel/api.php?API_SELECT=Contro
lPanel&Version=2.4&Request=getLegacyResponseData&User=CapellaDemo&Token=xmSMLuuV8Hm
MIo0n2LS1wyH8M211uGWCOSmpsCKb&Format=XML&SurveyID=SV_ahsdr1T9JBhAlsF&Labels=0'
    out=QData;
run;
```

SET UP FOR XML PARSING

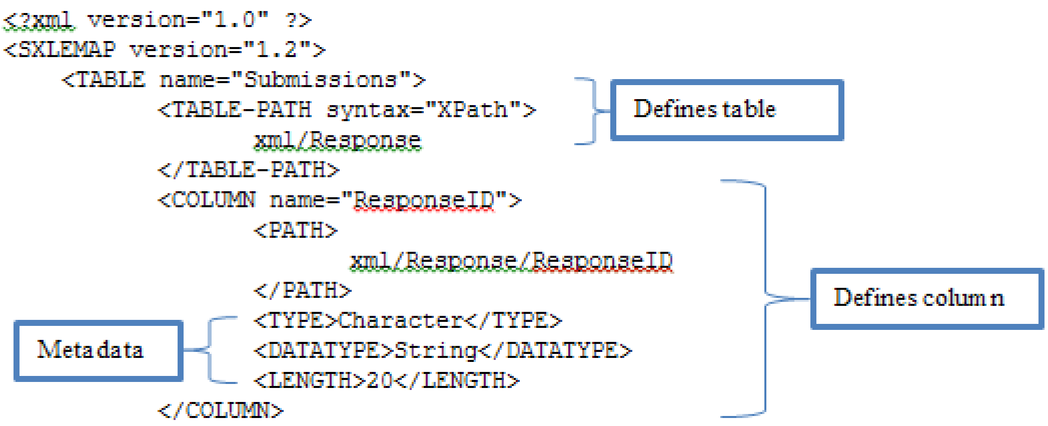
When the procedure is run, the request is sent to the Qualtrics system and a response file is placed in the filename path. The XML file is structured simply, with tags for the beginning and ending of a response and the beginning and ending of the data.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <xml>
  - <Response>
      <ResponseID>R_3R7xoZigWH25bgY</ResponseID>
      <ResponseSet>Default Response Set</ResponseSet>
      <Name>Anonymous</Name>
      <ExternalDataReference />
      <EmailAddress />
      <IPAddress />
      <Status>1</Status>
      <StartDate>2015-05-12 14:29:50</StartDate>
      <EndDate>2015-05-12 14:30:00</EndDate>
      <Finished>1</Finished>
      <Q1>1</Q1>
      <Q2>asdkl;fjsdkl;fj</Q2>
      <Q3>1</Q3>
    </Response>
  - <Response>
      <ResponseID>R_wY4fhbgYNuofwpH</ResponseID>
      <ResponseSet>Default Response Set</ResponseSet>
```

Figure 4. Qualtrics API response from the PROC HTTP get data request

Each additional tag within the Response tags represents survey questions or embedded data. While the structure of the response file is very simple, the SAS xml engine is not able to process the file without a data map. Fortunately, data maps can be easily created due to the simple structure of the API response XML. Within the SXLE file that serves as the data map, the formats that were defined earlier can be used within the data map to perform some manipulation by using the <FORMAT> and <INFORMAT> tags.



```
<?xml version="1.0" ?>
<SXLEMAP version="1.2">
    <TABLE name="Submissions">
            <TABLE-PATH syntax="XPath">
                    xml/Response
            </TABLE-PATH>
            <COLUMN name="ResponseID">
                    <PATH>
                            xml/Response/ResponseID
                    </PATH>
                    <TYPE>Character</TYPE>
                    <DATATYPE>String</DATATYPE>
                    <LENGTH>20</LENGTH>
            </COLUMN>
```

Figure 5. Sample SXLE map representation of a table column

The data map is referenced in the file through another filename statement.

```
/*Set the path for the XML map needed to structure the incoming data*/
/*!!This file sets the metadata (formats, informats, column labels) for the
incoming data! Changes to this information should be made on this map!*/

filename DataMap "C:\Users\KTanner1\datamap.map";
```

The XML engine is then accessed using a libname statement with an option to specify a xmlmap.

```
/*Turn on the XML import engine*/
```

```
   libname Qdata xml xmlmap=DataMap;
```

ACCESS THE XML ENGINE USING LIBNAME XML AND A DATA STEP

With the XML engine turned on, SAS can now read in to a dataset. The DATA step works well for this import, and allows for additional data manipulation to be done to the final SAS file.

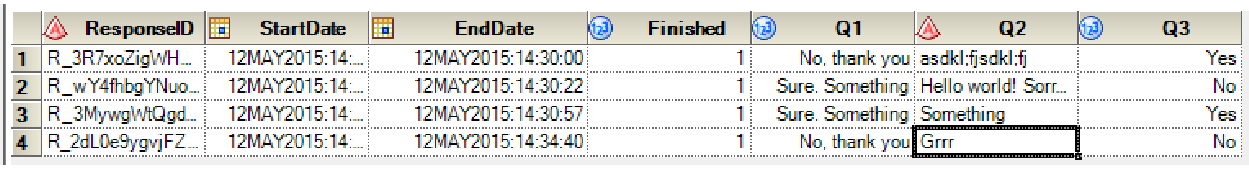
```
/*Now incorporate the API download into a SAS Dataset*/

   data work.AllSubmissions;
       set Qdata.Submissions;
/* additional data manipulation can be placed here*/
   run;
```

What is really nice about this method of bringing data in is that we get a helpful error message in the log if the length that we supply for string variables is not long enough. In this example, the length for the Q2 variable is not long enough. SAS writes this in the log, and then tells me how long the longest record is in the file.

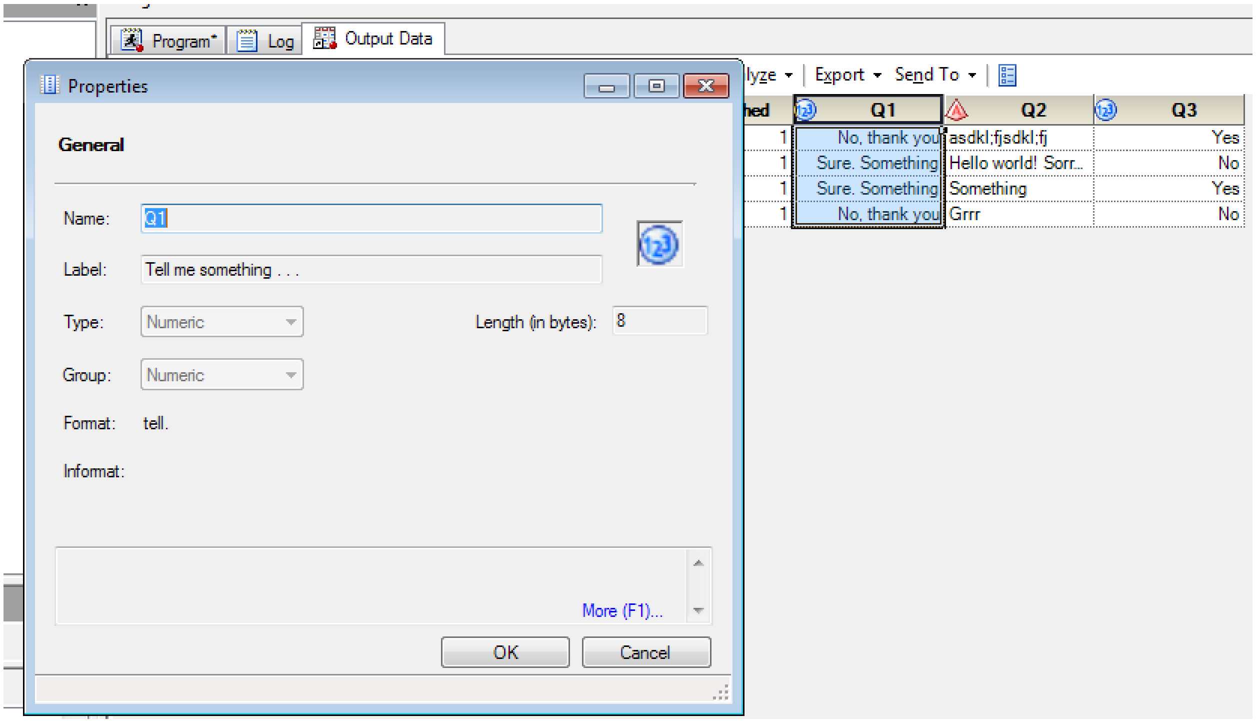
```
WARNING: At least one data value for column Q2 may have been
truncated. Maximum length of data
           encountered during input was 42.
```

Now the data is nicely parsed into a data file, and the table metadata is all included.



	ResponseID	StartDate	EndDate	Finished	Q1	Q2	Q3
1	R_3R7xoZigWH...	12MAY2015:14:...	12MAY2015:14:30:00	1	No, thank you	asdkl;fjsdkl;fj	Yes
2	R_wY4fhbgYNuo...	12MAY2015:14:...	12MAY2015:14:30:22	1	Sure. Something	Hello world! Sorr...	No
3	R_3MywgWtQgd...	12MAY2015:14:...	12MAY2015:14:30:57	1	Sure. Something	Something	Yes
4	R_2dL0e9ygvjFZ...	12MAY2015:14:...	12MAY2015:14:34:40	1	No, thank you	Grrr	No

Figure 6. The resulting data step displaying the parsed XML data



Name: Q1

Label: Tell me something . . .

Type: Numeric

Length (in bytes): 8

Group: Numeric

Format: tell.

Informat:

hed	Q1	Q2	Q3
1	No, thank you	asdkl;fjsdkl;fj	Yes
1	Sure. Something	Hello world! Sorr...	No
1	Sure. Something	Something	Yes
1	No, thank you	Grrr	No

Figure 7. Display of the metadata applied to the data set by the XML map

A libname clear statement releases the XML file from the use of SAS® so that it can be accessed by other programs.

```
   libname Qdata clear;
```

CONCLUSION

The REST API available in Qualtrics' Research Suite allows for easy integration of survey data and administration tasks with other enterprise systems, including SAS®. SAS® can be used to store or build API request URLs, send requests to the Qualtrics API using the HTTP Procedure, and can import the API response file into a dataset that can be further manipulated and stored. Therefore, the entire process lives within one SAS® program and one XML map file and can be set up for batch mode processing if survey changes are infrequent.

RECOMMENDED READING

- The SAS Dummy Blog provides additional information on the use of the HTTP procedure with hosted information.
- SAS Documentation: The HTTP Procedure provides syntax requirements for the HTTP procedure
- Learn REST: A Tutorial provids great introductory information about the REST API and how it differs from other API protocols
- Qualtrics Control Panel REST API Documentation is the comprehensive source for information about the Qualtrics API and includes a form field interface that assists in the build of API request URLs
- Qualtrics Research Suite is Capella University's primary tool for collecting data online.
- Using the Vovici API in SAS 9.2 provides additional information on the HTTP procedure and how SAS® interfaces with online data collection tools.

CONTACT INFORMATION

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

Name: Katie Tanner
Enterprise: Capella University
Address: 225 South Sixth Street, 9th Floor
City, State ZIP: Minneapolis, MN 55402
Work Phone: (612) 977-5645
Fax:
E-mail: Katie.Tanner@capella.edu
Web: