

## **SAS® and REDCap API: Efficient and Reproducible Data Import and Export**

Sarah Worley, Cleveland Clinic, Cleveland, OH  
Dongsheng Yang, Cleveland Clinic, Cleveland, OH

### **ABSTRACT**

REDCap (Research Electronic Data Capture), a web application for the development and use of online research databases, allows SAS® software users to download automatically-generated SAS code for importing, labeling, and formatting data. REDCap Application Programming Interface (REDCap API) provides SAS users with the option to import and export data, files, and database settings through SAS programs. The use of REDCap API with both PC and server installations of SAS 9.3 will be demonstrated.

### **INTRODUCTION**

Academic medical centers, universities, and other research organizations are increasingly turning to REDCap (Research Electronic Data Capture), a web-based application designed to support data capture for research studies, as an alternative to desktop spreadsheet and database programs. Through a graphical user interface (GUI), REDCap users can design study databases, enter records, and export data for analysis and reporting. In early versions of REDCap, study data, associated files, and data dictionaries were downloaded solely through the GUI. Recently, the implementation of the REDCap API provides SAS users with the option to import and export data, files, and database settings from within SAS. The use of REDCap API with both PC and server installations of SAS 9.3 will be demonstrated in this paper.

### **THE REDCAP APPROACH TO DATA CAPTURE**

Each REDCap project consists of a single database table and its associated metadata (the 'data dictionary'), a list of users with their access privileges, and other project settings. Variables ('fields') can be organized into data collection forms for convenience, but field names must be unique due to the single table structure. Traditional databases contain one observation for each unique record identifier; longitudinal databases allow for multiple observations per record. In a longitudinal REDCap project, users define multiple 'events', the times at which the same variables (grouped into forms) can be repeatedly collected, and may also define 'arms', or subgroups of records that will have the same forms collected for the same events.

Data are exported as a single comma separated value (.CSV) file, which can then be read into nearly any spreadsheet, text editor, or statistical software program. Project metadata can also be exported this way, and then modified and uploaded to alter the database structure.

Through the REDCap GUI, a user can download a SAS program generated by REDCap which will read in, label, and format the .CSV data file using the DATA step with INFILE statement and the Format procedure. The SAS user need make only minimal changes to this SAS program, such as correcting path names or writing formats and the final data set to a permanent library. Because this SAS program references the .CSV data file, it depends only on the structure of the database and does not need to be updated when individual records are added, modified, or deleted.

### **WHY USE REDCAP API?**

For projects in which data will be exported and analyzed only once, obtaining data through the REDCap GUI is easy and efficient. However, if analyses and reports must be made repeatedly during the life of the project, regular reporting is required for a number of ongoing projects, or multiple REDCap projects are used in a single research project to implement a multi-table database structure, the GUI export process is time-consuming. Furthermore, project settings such as user rights and longitudinal database event and arm lists can be viewed through the GUI, but cannot be exported in a useful format for including in a report.

With REDCap API, a single, reusable SAS program can quickly obtain a REDCap project's data and settings, and produce reports and analyses.

## HOW REDCAP API WORKS

REDCap API is a tool for performing many of the GUI tasks via other software using HTTP protocols. Actions available in REDCap API include: export/import observations ("records"), export data dictionary, export/import/delete file linked to subject, and export user list. For longitudinal studies, additional actions are available: export events, export arms, and export form-event table.

A REDCap user with API user permissions can request a user- and project-specific code ("token") via the GUI. This token takes the place of a username and password, and is usable until it is revoked by the REDCap user. Each API action has a set of required and optional parameters that define the request and are supplied to the REDCap API from SAS, such as the project token, the content requested, and the type of file to return. The REDCap API will also return a header file with HTTP status codes and other information, such as '200 OK' or '404 Not Found'. The list of available actions, required and optional API parameters for each action, and HTTP return codes are listed on the REDCap API Help Page, which is linked from the API application screen in API-enabled projects.

## SAS AND REDCAP API

The basic SAS code needed to exchange data with REDCap through the API consists of a DATA step to create a file with the API parameters that define the action to be performed by REDCap API, and an Http procedure (available in Base SAS beginning with version 9.2) to transfer the API parameter file and the data file between SAS and REDCap. Data exported from REDCap are saved as .CSV files and can be read into SAS using the REDCap-generated SAS program, or through traditional data import methods such as a DATA step or Import procedure.

The REDCap API parameter file is a text file consisting of a case-sensitive text string which contains the required and optional parameters for the action, of the form *parameter1=option1 & parameter2=option2*. Since the & character, which is used in SAS Macro Language to indicate macro variables, is now needed to separate each pair of parameters and options in the REDCap API parameter file, a SAS Macro Language function, %NRSTR, is used to mask the & symbol during execution of the DATA step. References to macro variables can be used in the DATA step if placed outside of the %NRSTR function, as shown in the example below.

Note that REDCap API may time out when a request for the export of a large file is made. In these cases, it is often helpful to request the export of subsets of data (such as specific records, fields, events, and/or arms), and merge these data sets in SAS.

The following detailed example shows how to export data from REDCap and read it into a SAS data set, using SAS 9.3 for Windows. With appropriate corrections to the FILENAME statements, the same code will work for SAS 9.3 installed on a UNIX server.

## EXAMPLE DATA EXPORT

```
/******  
  
Step 1. Define file names and macro variable for the  
project-specific token.  
  
*****/  
  
*** Text file for API parameters that the define the request sent to  
REDCap API. Will be created in a DATA step. Extension can be .csv,  
.txt, .dat ***;  
  
filename my_in "C:\myproject\api_parameter.txt";  
  
*** .CSV output file to contain the exported data ***;  
  
filename my_out "C:\myproject\redcap_data.csv";
```

```

*** Output file to contain PROC HTTP status information returned from
REDCap API (this is optional) ***;

filename status " C:\myproject\redcap_status.txt";

*** Project- and user-specific token obtained from REDCap ***;

%let mytoken = XXXXXXXXXXXXXXX;

/*****

Step 2. Request all observations (CONTENT=RECORDS) with one
row per record (TYPE=FLAT). Note: Obtain your site-specific
url from your local REDCap support team.

*****/

*** Create the text file to hold the API parameters. ***;

data _null_ ;
file my_in ;
put "%NRStr(content=record&type=flat&format=csv&token=)&mytoken";
run;

*** PROC HTTP call. Everything except HEADEROUT= is required. ***;

proc http
  in= my_in
  out= my_out
  headerout = status
  url ="https://redcap.mysite.org/api/"
  method="post";
run;

/*****

Step 3. Read .CSV data file into SAS and save it.

Uses project-specific SAS code generated through the REDCap
Data Export Tool with a modification (shown below) to the
first DATA step in the program to work with the REDCap
API data.

*****/

*** Code copied from the REDCap-generated SAS program
(this is optional, may use PROC IMPORT or a DATA step) ***;

*** Change INFILE name to data named in PROC HTTP OUT = ***;
*** Change FIRSTOBS=1 to FIRSTOBS=2 to indicate that
a header row exists ***;

<lines omitted>

data REDCAP; %let _EFIERR_ = 0;
infile my_out delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;

<lines omitted>

run; quit;

```

```

*** Save as a permanent SAS data set ***;

libname mydata "C:\myproject";
data mydata.redcap_export;
    set redcap;
run;

```

## ADDITIONAL EXAMPLES

Line breaks in the REDCap API parameter file are ignored as long as the & symbol precedes each parameter name, and may be used to improve the readability of the code and REDCap API parameters file. For example, the DATA step at the beginning of Step 2 in the example above could be replaced by the following code, which creates a REDCap API parameter file with the token on the first line and the rest of the parameters and options on the second line.

```

%let req1 = token=&mytoken.;
%let req2 = %nrstr(&)content=record%nrstr(&)format=csv%nrstr(&)type=flat;

data api_parameters;
    length name $200.;
    name = "&req1.";output;
    name = "&req2.";output;
run;

data _null_;
    file my_in;
    set api_parameters;
    put name;
run;

```

The table below lists selected REDCap API actions with the corresponding changes to the SAS code given in the example above. No change to the PROC HTTP statement is required.

<b>Examples of REDCap API tasks and corresponding SAS code. Substitute the PUT statement from the table for the one given in the DATA step in Step 2 of the example above. If a FILENAME statement is listed in the table, substitute it for the one given in Step 1 of the example above.</b>	
<b>REDCap API actions</b>	<b>SAS code for request.</b>
Export REDCap data, limiting to fields <i>record_id</i> , <i>male</i> , and <i>age_yrs</i>	put "%NRStr(content=record&type=flat&format=csv &fields=record_id,male,age_yrs&token=) &token";
Export REDCap data, limiting to records with ids equal to 1, 3, 5, and 7	put "%NRStr(content=record&type=flat&format=csv &records=1,3,5,7&token=) &token";
Export data dictionary to .CSV file	put "%NRStr(content=metadata&format=csv &token=) &token";
Export list of users to .CSV file	put "%NRStr(content=user&format=csv &token=) &token";
Export a file attached to record number 1. The REDCap database contains a 'File Upload' field named <i>consent</i> . The file extension, but not the name, of the file referenced by MYOUT must match the actual document stored in REDCap.	filename my_out "C:\myproject\Subject1_Consent.docx"; put "%NRStr(content=file&action=export&record=1 &field=consent&format=csv&&token=) &token";

<b>Examples of REDCap API tasks and corresponding SAS code. Substitute the PUT statement from the table for the one given in the DATA step in Step 2 of the example above. If a FILENAME statement is listed in the table, substitute it for the one given in Step 1 of the example above.</b>	
<b>REDCap API actions</b>	<b>SAS code for request.</b>
Export list of data access groups for a multi-center study. Request labeled rather than the default raw data.	<pre>put "%NRStr (content=record&amp;type=flat&amp;format=csv &amp;exportDataAccessGroups=true&amp;rawOrLabel=label &amp;token=) &amp;token";</pre>
Export list of events with timing of each in a longitudinal database.	<pre>put "%NRStr (content=event&amp;format=csv &amp;token=) &amp;token";</pre>
Export list of arms in a longitudinal database.	<pre>put "%NRStr (content=arm&amp;format=csv &amp;token=) &amp;token";</pre>
Export list of events and corresponding data collection forms in a longitudinal database.	<pre>put "%NRStr (content=formEventMapping &amp;format=csv&amp;token=) &amp;token";</pre>

## CONCLUSION

REDCap API is a useful tool that enables SAS programmers to perform repetitive data and file operations more efficiently. The ability to perform these tasks within SAS improves the reproducibility and documentation of the research process. SAS users who work with REDCap should keep the existing capabilities and benefits in mind, and look for improvements to the API when reading release notes for new versions of REDCap.

## REFERENCES

Paul A. Harris, Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, Jose G. Conde, Research electronic data capture (REDCap) - A metadata-driven methodology and workflow process for providing translational research informatics support, J Biomed Inform. 2009 Apr;42(2):377-81.

## CONTACT INFORMATION

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

Sarah Worley  
 Department of Quantitative Health Sciences  
 Cleveland Clinic  
 9500 Euclid Ave. /JJN3-01  
 Cleveland, OH 44195  
 E-mail: worleys@ccf.org

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.