

REDCap API for SAS

Posted on July 6, 2016

Updated on November 26, 2019

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

Sarah Worley, Cleveland Clinic, Cleveland, OH

Dongsheng Yang, Cleveland Clinic, Cleveland, OH

Feedback from our users:

To use the API via SAS, the user needs to do the following:

- 1) Create a text file (named myfile.txt below) that contains the API token information
- 2) Create a CSV file (named myfile.csv below) via PROC HTTP using the following code:

***Pull data using API;**

```
filename myfilein "<LOCATION OF TEXT FILE WITH API TOKEN INFO>\myfile.txt";
```

```
filename myfileout "<LOCATION OF OUTPUT CSV FILE>/myfile.csv";
```

```
proc http
```

```
    in= myfilein out=myfileout
```

```
    url="https://redcap.partners.org/redcap/api/"
```

```
        method="post"
```

```
        ct="application/x-www-form-urlencoded";
```

```
run;
```

```
quit;
```

- 3) Read in the data from the CSV file (using the SAS code downloaded from REDCap, but replacing the file referenced on the INFILE line with the newly created CSV file:

***Read in new version of CSV file;**

```
data REDCAP; %let _EFIERR_ = 0;
```

```
infile "<LOCATION OF OUTPUT CSV FILE>\myfile.csv" delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2;
```

```
    informat participant_id $500. ;
```

```
    informat redcap_survey_identifier $500. ;
```

```
    etc.
```

If there are newline characters (carriage return) within some of the text fields, you may receive errors. When you export a .csv file from REDCap these characters are removed, you should have no trouble calling the data into SAS. But when you create a .csv file using the above code those newline characters are still in the .csv, and SAS goes to a new line when it hits them and you will get an error trying to import the data from the .csv into SAS (if you open the .csv it looks perfect).

Use this code from the SAS website:

```
/*                      */
/* This program UPDATES IN PLACE, create a backup copy before */
/* running.                      */
/*                      */
/****** CAUTION *****/

/* Replace carriage return and linefeed characters inside */
/* double quotes with a specified character. This sample */
/* uses '@' and '$', but any character can be used, including */
/* spaces. CR/LFs not in double quotes will not be replaced. */

%let repA=' ';      /* replacement character LF */
%let repD=' ';      /* replacement character CR */

%let dsnnme="H:\redcap_outavika.csv"; /* use full path of CSV file */

data _null_;
  /* RECFM=N reads the file in binary format. The file consists */
  /* of a stream of bytes with no record boundaries. SHAREBUFFERS */
  /* specifies that the FILE statement and the INFILE statement */
  /* share the same buffer. */
  infile &dsnnme recfm=n sharebuffers;
  file &dsnnme recfm=n;

  /* OPEN is a flag variable used to determine if the CR/LF is within */
```

```
/* double quotes or not. Retain this value. */
```

```
retain open 0;
```

```
input a $char1.;
```

```
/* If the character is a double quote, set OPEN to its opposite value. */
```

```
if a = '"' then open = ^ (open);
```

```
/* If the CR or LF is after an open double quote, replace the byte with */
```

```
/* the appropriate value. */
```

```
if open then do;
```

```
if a = '0D'x then put &repD;
```

```
else if a = '0A'x then put &repA;
```

```
end;
```

```
run;
```

Go to [KB0027897](#) in the IS Service Desk



FOUNDED BY BRIGHAM AND WOMEN'S HOSPITAL
AND MASSACHUSETTS GENERAL HOSPITAL

[Home](#) [Events](#) [Careers](#) [Contact Us](#) [Website Feedback](#) [Web Privacy Statement](#)

© 2010-2020 Partners HealthCare. All rights reserved