

## REDCap Ins and Outs

Leanne Goldstein, City of Hope, Duarte, CA [now at Deloitte Consulting]

### ABSTRACT

REDCap (Research Electronic Data Capture) is making waves around the world as the latest way to capture data at academic institutions. Its greatest benefit is that it can provide multi-user, web based created databases, with audit trails and reporting. It is an improvement to using MS Access databases and MS Excel, which investigators have historically used to capture data in academic settings. Application Programming Interfaces (APIs) have been developed for programmers to get data in and out of REDCAPs, but the SAS community has been somewhat left behind. This paper discusses an introduction to REDCAP for the SAS programmer and discusses methods of getting data into and out of REDCAP databases.

### INTRODUCTION

This paper will demonstrate a step-by-step approach towards building a database and then using SAS to extract data from the database and fill the database. The methodology used here applies the techniques of REDCap Application Programming Interface (API) examples that show how to manage REDCap databases with R, Python, and other similar tools. However, no examples as of yet have been built explicitly for SAS, and this may put SAS users at a disadvantage. This program will leverage the examples for R and show how with some simple steps and use of PROC IML, SAS users can use APIs to manage their REDCap databases. The code shown in this paper was developed using SAS 9.4, PROC IML 14.1, R 32-bit 3.3.2, and REDCap 7.1.2.

### CREATE NEW REDCAP PROJECT

For this paper, we will need to create a sample REDCAP database. After establishing a REDCap account at your institution, login and complete the following steps.

1. Click on Create a New Project in REDCap
2. Give the project a name, under Purpose of this Project select Practice/Just for Fun, and select Create an empty project.

Figure 1 below gives an example of the dialogue box when creating a new REDCap Project.

**REDCap** Home My Projects **+ New Project** Help & FAQ Training Videos Send-It

#### + Create a new REDCap Project

You may begin the creation of a new REDCap project on your own by completing the form below and clicking the Create Project button at the bottom.

**Project title:**   
Title to be displayed on project webpage

**Purpose of this project:**   
How will it be used?

**Project notes (optional):**   
Comments describing the project's use or purpose that are displayed on the My Projects page.

**Start project from scratch or begin with a template?**

- Create an empty project (blank slate)
- Upload a REDCap project XML file (CDISC ODM format) [?](#)
- Use a template (choose one below)

**Figure 1. Creating a New REDCap Project Dialogue Box**

### SETUP REDCAP FORM FOR DATA ENTRY

In the next step, click on Project Setup-> Online Designer -> My First Instrument and start adding fields to the survey tool that you would want to collect by clicking Add Field. Add the following variables listed in Table 1 to be compatible with this program. An example dialogue box seen when adding a new field is in Figure 2.

Field Type	Field Label	Variable Name	Validation	Choices
Text Box	Record ID	record_id*	---- None ----	
Text Box	First Name	firstname	---- None ----	
Text Box	Last Name	lastname	---- None ----	
Text Box	Date of Birth	dob	Date (M-D-Y)	
Multiple Choice – Drop-down list	Sex	sex	---- None ----	1, Male 2, Female
Notes Box	Comments	comments	---- None ----	

\*Note: record\_id is by default usually the first variable in a form

**Table 1. Variables for REDCap Form My First Instrument**

**Add New Field**

You may add a new project field to this data collection instrument by completing the fields below and clicking the Save button at the bottom. When you add a new field, it will be added to the form on this page. For an overview of the different field types available, you may view the [Field Types video \(4 min\)](#).

**Field Type:** Text Box (Short Text, Number, Date/Time, ...) ▼

**Field Label** [How to use Piping](#)

First Name

**Action Tags / Field Annotation** (optional)

[Learn about Action Tags](#) or [using Field Annotation](#)

**Variable Name** (utilized during data export)

firstname  Enable auto naming of variable based upon its Field Label?

ONLY letters, numbers, and underscores

**Validation?** (optional)

---- None ---- ▼

– or –

**Enable searching within a biomedical ontology** [?](#)

-- choose ontology to search -- ▼

**Required?\***  No  Yes  
\* Prompt if field is blank

**Identifier?**  No  Yes  
Does the field contain identifying information (e.g., name, SSN, address)?

**Custom Alignment** Right / Vertical (RV) ▼  
Align the position of the field on the page

Save Cancel

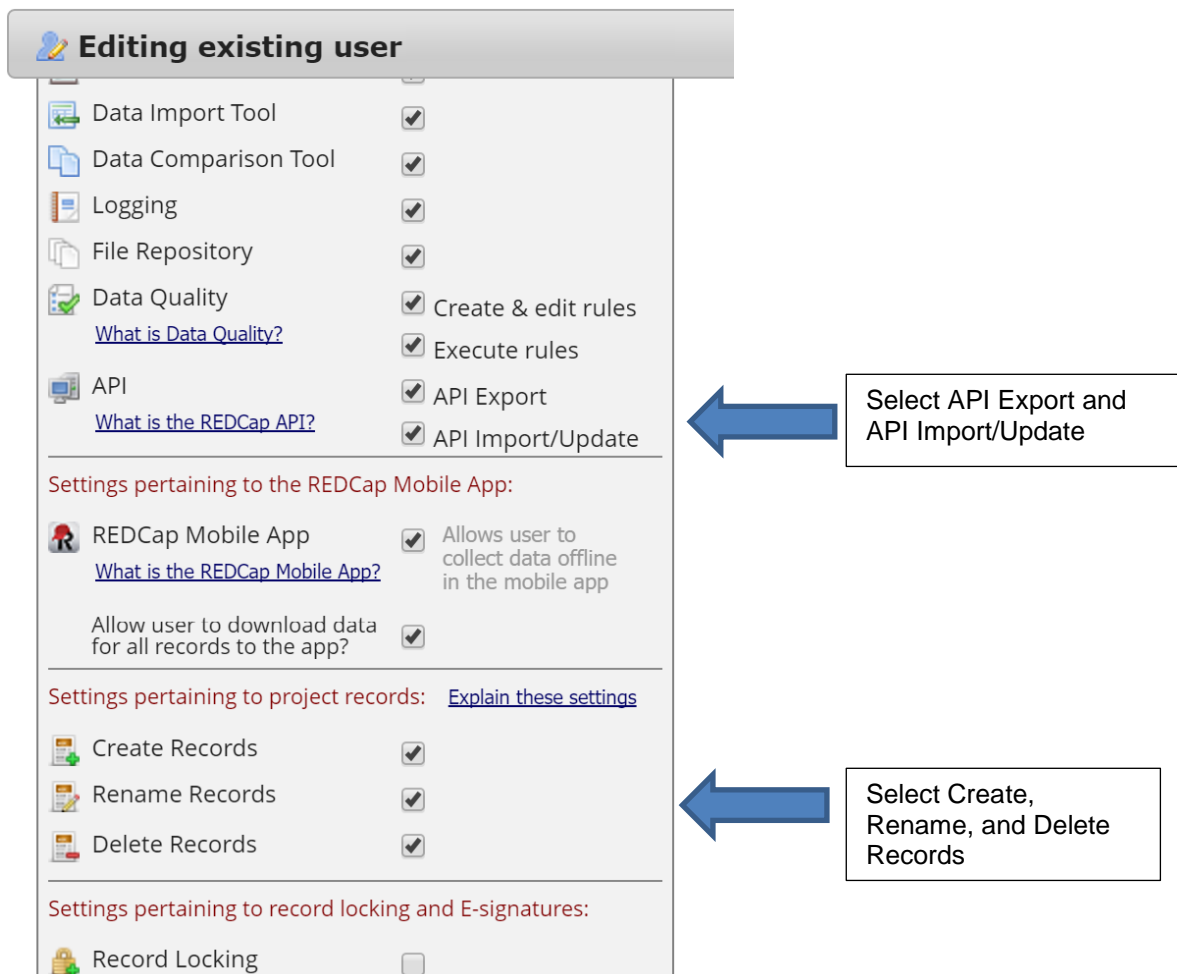
**Figure 2. Add New Field Dialogue Box in REDCap**

## EDIT REDCAP USER PRIVILEGES

Once the form has been built, you need to modify the default user privileges to allow you to

1. Use the Application Programming Interface (API) options and API Playground
2. Create, Modify, and Delete Records

On the left hand side of the REDCap Project, under the Applications menu, select User Rights. Find yourself under username then click and select Edit User Privileges and save. Figure 3 shows the dialogue box for editing user privileges in REDCap.



**Figure 3. Edit User Privileges Dialogue Box in REDCap**

## CREATE NEW RECORDS IN REDCAP DATABASE

Now that the database is setup for entry and the user privileges have been managed, you will want to fill the database with a couple records. I create two records here, but if you want practice, feel free to add more.

On the left hand side of the REDCap Project, under Data Collection, Select Add/Edit Records. Enter data into the form then Click Save & Exit Form. Figure 4 shows an example of the form into which you will enter data.

The screenshot shows a data entry form for Record ID 4. The fields are: First Name (text input), Last Name (text input), Date of Birth (calendar icon, Today, M-D-Y), Sex (dropdown menu), and Comments (large text area). Below the form is a 'Form Status' section with a 'Complete?' dropdown set to 'Incomplete'. At the bottom are buttons for 'Save & Exit Form', 'Save & Stay', and '-- Cancel --'.

Figure 4. Data Entry Form in REDCap

## REQUEST API TOKEN

Once you have created a few records, make a request for an API Token. You will need to request this from the REDCAP administrator and wait for them to fulfill this request. Go to the left hand side of the REDCap Project, under Applications, Select API, and then Click on Generate Token. Once approved by the REDCAP administrator, the API Token will appear under API and this will be needed for API code. Figure 5 shows an example of the API screen once the token request has been fulfilled.

**Your API token for project "REDCAP for PharmaSUG"**

The API token below is **ONLY for you** and will work **ONLY with this project**. This token allows special access to REDCap data and **should NOT be shared with others**. If you think your token has been compromised, then please contact your REDCap administrator immediately *AND* either delete or regenerate your token by using the buttons below.

👤 **API Token:**

**XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX**

Finished using the API for this project? If so, please delete your token for security reasons.

Think someone else knows your token? If so, please regenerate your token for security

Figure 5. API Token for REDCap Project

## SETUP R

In order to run the API code, you will need to download R and install a few libraries.

1. Go to R homepage: <https://cran.r-project.org/> and Download R.

2. Open R and go to the dropdown menu Packages -> Install Packages. It may prompt you first to select a CRAN mirror, Select 0-Cloud [https] or another of your choosing. Navigate to RCurl and click OK. Do the same for Packages: digest and jsonlite.
3. When all packages complete installing, close R.

Figure 6 shows examples of the CRAN mirror menu and Packages menu in R.

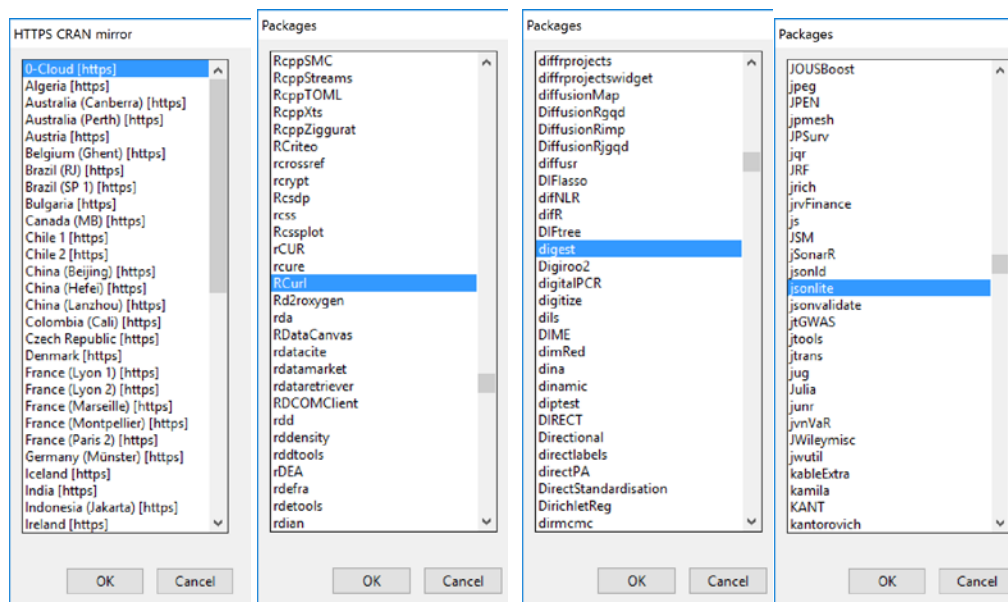


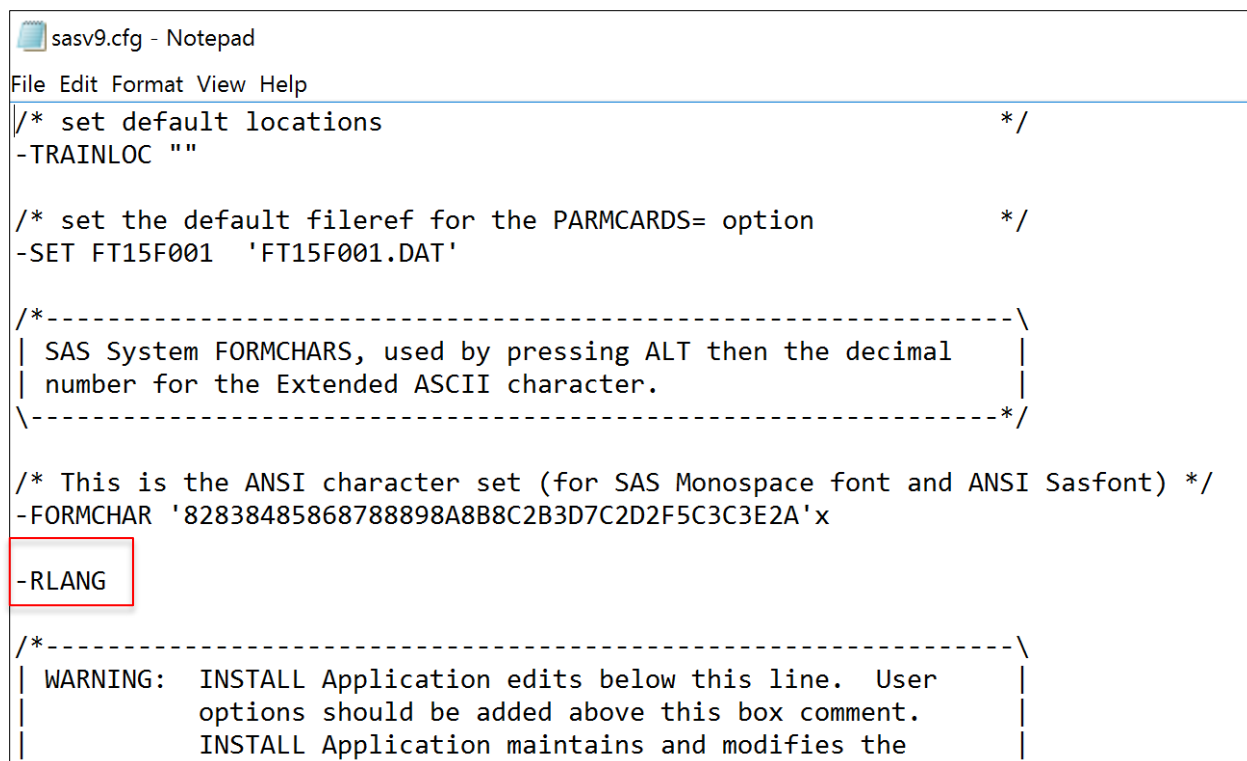
Figure 6. CRAN mirror and Packages menu examples in R

## SETUP SAS

There are a couple SAS setup steps that will need to be executed.

1. Locate your SAS configuration file called sasv9.cfg often located in C:\Program Files\SASHome\SASFoundation\9.4\nls\en
2. Next, open the configuration file and add `-RLANG`. Save and close. *Note first you may want to right click on the configuration file and choose Properties -> Security and see that you have the permission to write to this file.*

Figure 7 shows an example of the SAS configuration file and where to insert `-RLANG`.



```

sasv9.cfg - Notepad
File Edit Format View Help
/* set default locations */
-TRAINLOC ""

/* set the default fileref for the PARMCARDS= option */
-SET FT15F001 'FT15F001.DAT'

/*-----\
| SAS System FORMCHARS, used by pressing ALT then the decimal |
| number for the Extended ASCII character. |
\-----*/

/* This is the ANSI character set (for SAS Monospace font and ANSI Sasfont) */
-FORMCHAR '82838485868788898A8B8C2B3D7C2D2F5C3C3E2A'x

-RLANG

/*-----\
| WARNING:  INSTALL Application edits below this line.  User |
|           options should be added above this box comment. |
|           INSTALL Application maintains and modifies the |
\-----*/

```

**Figure 7. SAS Configuration File**

## SAS CODE API TO EXPORT DATA FROM REDCAP

Once all the setup steps for REDCap, SAS, and R are complete. The following code can be created in SAS to export all the data automatically. While REDCap provides the capability for users to export data from the user interface, this manual process may become cumbersome. Building an API code will allow for ease of automation. The code below does the following:

1. Sets up use of R Language in PROC OPTIONS
2. Uses R code embedded within PROC IML to pull all REDCap data
3. Creates a SAS data set called MyData with all records
4. Prints out the MyData SAS data set

The example provided below uses an example API token and API URL, please obtain the appropriate token and URL from the API area of your REDCap or ask the REDCap administrator.

```

PROC OPTIONS OPTION=RLANG;RUN;

PROC IML;
SUBMIT/R;
require(RCurl)

api_token <- 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
api_url <- 'https://redcap.myuniversity.edu/api'

RAW.API <- postForm(
  api_url,
  token=api_token,
  content="record",

```

```

type="flat",
format="csv",
.opts=curlOptions(ssl.verifypeer=TRUE,verbose=FALSE)
)

RData <- read.table(file = textConnection(RAW.API), header = TRUE,
  sep = ",", na.strings = "", stringsAsFactors = FALSE)

ENDSUBMIT;
call ImportDataSetFromR("Work.MyData", "RData");
QUIT;
PROC PRINT DATA=MYDATA; RUN;

```

The resulting output from the above code is in Output 1. The resulting output from SAS is the whole unformatted data set with raw variable names instead of labels. For example, sex is listed as 1 and 2 instead of Male and Female and my\_first\_instrument\_complete is listed as 0, 1, and 2, instead of Incomplete, Unverified, and Complete. Also, note the data format is YYYY-MM-DD instead of the M-D-Y we designed it to be when we created the field in REDCap.

Obs	record_id	firstname	lastname	dob	sex	comments	my_first_instrument_complete
1	1	Mickey	Mouse	2017-04-05	1	This is a comment.	2
2	2	Donald	Duck	1970-01-01	1	This is another comment.	1

#### Output 1. Output of MYDATA, all records from REDCap Project

Since your project may have many fields, you may want to obtain the formats and variable labels in batch so you don't have to manual code each one. Formats and variable labels can be obtained from the SAS program for exporting data to SAS.

1. On the left hand side of the REDCap Project, under Applications, select Data, Exports, Reports, and Stats
2. Select All Data and Select SAS for Export option
3. Click on the SAS Icon green arrow which downloads the SAS program. Open up the SAS program to retrieve formats and labels, which can be added in to the program on the left in another data step.

Figure 8 shows an example of the dialogue box that appears in REDCap User Interface when exporting data to SAS and Figure 9 is an example of the SAS code with labels and formats that is generated from that export. The code from the program generated can be included in your program and MyData SAS data set can have labels and formats added in an additional DATA step.

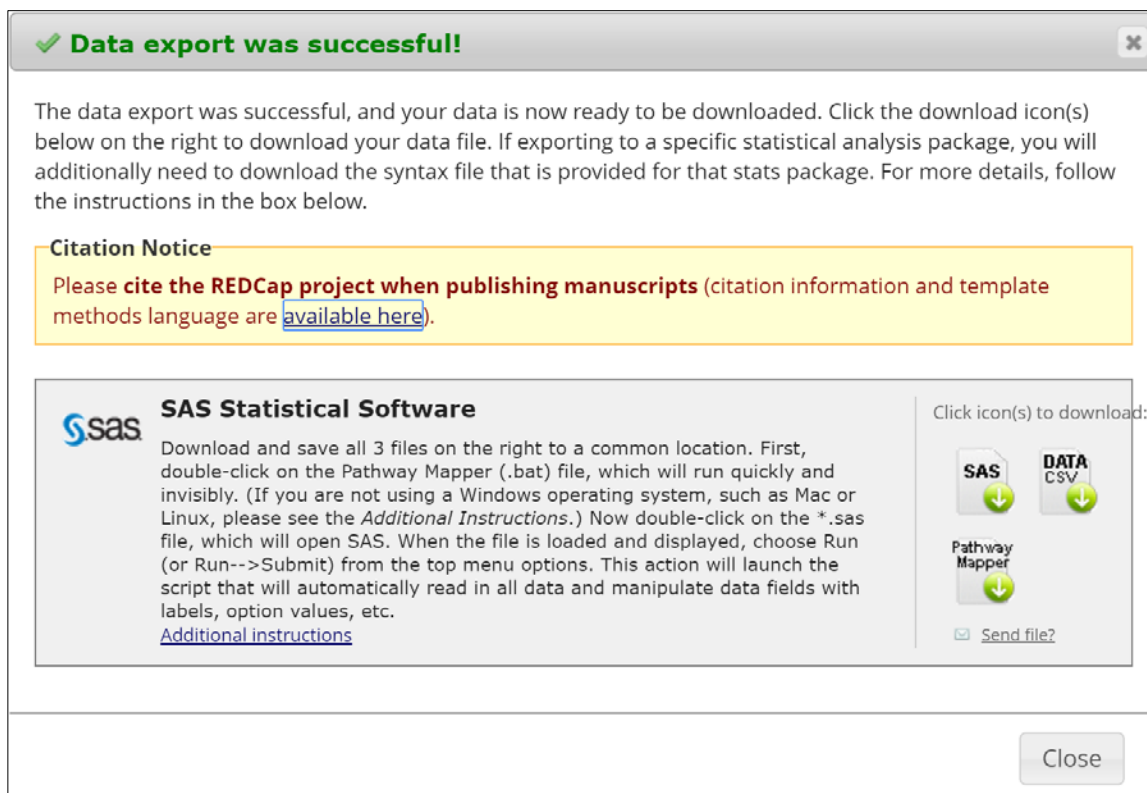


Figure 8. REDCap Dialogue Box for Exporting Data to SAS

```

data redcap;
  set redcap;
  label record_id='Record ID';
  label firstname='First Name';
  label lastname='Last Name';
  label dob='Date of Birth';
  label sex='Sex';
  label comments='Comments';
  label my_first_instrument_complete='Complete?';
run;

proc format;
  value sex_ 1='Male' 2='Female';
  value my_first_instrument_complete_ 0='Incomplete' 1='Unverified'
    2='Complete';
run;

data redcap;
  set redcap;

  format sex sex_.;
  format my_first_instrument_complete my_first_instrument_complete_.;

```

Figure 9. SAS Code generated when Exporting REDCap data



## SAS CODE API TO IMPORT DATA INTO REDCAP

While exporting data from REDCap is a more often seen need of programmers and analysts, occasionally investigators may ask for data to be loaded into REDCAP. For example, if a clinician is abstracting chart information into a REDCAP but doesn't want to do the additional steps of entering every single patient and every single demographic, and API import would be very helpful. The code below shows how to import a single record into REDCap. The use of macro language in PROC IML is demonstrated here so that once this simple task is conquered, more advanced programming techniques could be applied to load an entire data set. This can be done by looping through Macros which is explained in SAS Documentation or in other SAS User papers. When setting up the data for import, note in the code below the informat of the data. The date format is specified as YYYY-MM-DD and all data even numeric data is embedded in quotes. Pay careful attention in the code below to how the macro `&_inputs` is used.

```
PROC OPTIONS OPTION=RLANG;RUN;

%let inputs='This is another note';

PROC IML;
  _inputs="&inputs";

  SUBMIT _inputs/R;

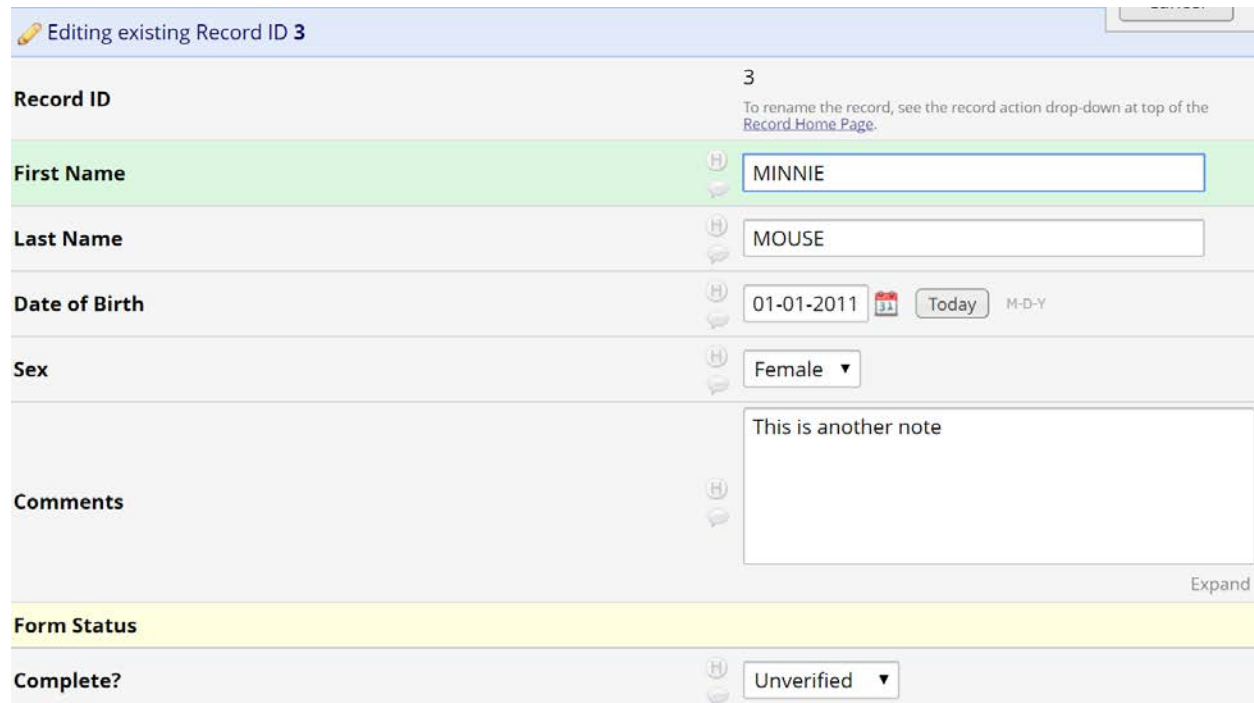
  library(RCurl)
  library(digest)
  library(jsonlite)

  api_token <- 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  api_url    <- 'https://redcap.myuniversity.edu/api'

  record<-c(
    record_id = '3', first='MINNIE',
    last='MOUSE', dob='2011-01-01',
    sex='2',      comments=&_inputs,
    my_first_instrument_complete='1'
  )
  data <- toJSON(list(as.list(record)), auto_unbox=TRUE)

  result <- postForm(
    api_url,          token=api_token,
    content='record', format='json',
    type='flat',     data=data
  )
ENDSUBMIT;
QUIT;
```

After running the API import code, open up REDCap and verify that the record(s) have been loaded correctly. Navigate to the left hand side of the REDCap Project and click on Add/Edit Records then navigate to one of the new records that have been uploaded. Figure 10 gives an example of what should be observed by running the above code.



Editing existing Record ID 3

**Record ID** 3  
To rename the record, see the record action drop-down at top of the [Record Home Page](#).

**First Name** MINNIE

**Last Name** MOUSE

**Date of Birth** 01-01-2011 Today M-D-Y

**Sex** Female

**Comments** This is another note  
Expand

**Form Status**

**Complete?** Unverified

**Figure 10. REDCap Data Entry Form for API Import Created Record**

## CONCLUSION

The Application Programming Interface (API) capabilities are currently focused on users of R, Python, Perl, Java, and other similar languages. The methodology demonstrated in this paper shows how a SAS user can leverage the API tools and playground of REDCAP with a few workarounds. The code shown here is just a beginning. It can be further enhanced by utilizing code found in REDCap's API playground or by building SAS macros and loops to migrate more than just a single record into REDCap.

## CONTACT INFORMATION

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

Leanne Goldstein  
[goldstein.leanne@gmail.com](mailto:goldstein.leanne@gmail.com)